

Douglas E. Comer

# SIECI

## komputerowe i intersieci

KOMPENDIUM WIEDZY  
KAŻDEGO ADMINISTRATORA!

WYDANIE V

Helion 



---

Douglas E. Comer

---

# SIECI

## komputerowe i intersieci

---

KOMPENDIUM WIEDZY  
KAŻDEGO ADMINISTRATORA!

---

WYDANIE V

Tytuł oryginału: Computer Networks and Internets, Fifth Edition

Tłumaczenie: Marek Pałczyński

Projekt okładki: Jan Paluch

ISBN: 978-83-246-3607-5

Authorized translation from the English language edition, entitled:  
Computer Networks and Internets, Fifth Edition, ISBN 0136061273,  
by Douglas E. Comer, published by Pearson Education, Inc,  
publishing as Prentice Hall,  
Copyright © 2009, 2004, 2001, 1999, 1997 by Pearson Education, Inc

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education Inc.

Polish language edition published by Helion S.A.

Copyright © 2012

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicielami.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/skit5>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

*Wszechobecnym pakietom*



# Spis treści

Przedmowa 19

## Część I Wprowadzenie do sieci komputerowych i aplikacji internetowych

27

### Rozdział 1. Wprowadzenie

29

- 1.1. Rozwój sieci komputerowych 29
- 1.2. Dlaczego komunikacja sieciowa wydaje się trudna? 30
- 1.3. Pięć kluczowych zagadnień sieciowych 30
- 1.4. Publiczne i prywatne obszary internetu 34
- 1.5. Sieci, współdziałanie i standardy 36
- 1.6. Stos protokołów i modele warstwowe 37
- 1.7. Przekazywanie danych między warstwami 39
- 1.8. Nagłówki i warstwy 40
- 1.9. Organizacja ISO i siedmiowarstwowy model odniesienia OSI 40
- 1.10. Kulisy standaryzacji 41
- 1.11. Pozostała część książki 42
- 1.12. Podsumowanie 43

### Rozdział 2. Kierunki rozwoju internetu

45

- 2.1. Wprowadzenie 45
- 2.2. Współdzielenie zasobów 45
- 2.3. Rozwój internetu 46
- 2.4. Od współdzielenia zasobów do komunikacji 47
- 2.5. Od tekstu do multimediów 49
- 2.6. Najnowsze trendy 50
- 2.7. Podsumowanie 51

### Rozdział 3. Aplikacje internetowe i programowanie sieciowe

55

- 3.1. Wprowadzenie 55
- 3.2. Dwa podstawowe pojęcia związane z internetem 56
- 3.3. Komunikacja połączeniowa 57
- 3.4. Model klient-serwer 58

3.5. Cechy aplikacji klienckich i serwerowych	59
3.6. Programy serwerowe oraz komputery pełniące rolę serwerów	59
3.7. Żądania, odpowiedzi i kierunek przepływu danych	60
3.8. Wiele aplikacji klienckich i serwerowych	60
3.9. Identyfikacja serwerów i demultiplesacja	61
3.10. Praca współbieżna serwerów	62
3.11. Pętla zależności między serwerami	63
3.12. Odwołania peer-to-peer	63
3.13. Programowanie sieciowe i interfejs gniazd	64
3.14. Gniazda, deskryptory i sieciowe operacje wejścia-wyjścia	64
3.15. Parametry i interfejs gniazd	65
3.16. Odwołania do gniazd w aplikacjach klienckich i serwerowych	66
3.17. Funkcje gniazda wykorzystywane po stronie klienta i serwera	66
3.18. Funkcja połączenia wykorzystywana jedynie po stronie klienta	68
3.19. Funkcje gniazda wykorzystywane jedynie po stronie serwera	69
3.20. Funkcje gniazda wykorzystywane w transmisji komunikatów	71
3.21. Inne funkcje gniazd	73
3.22. Gniazda, wątki i dziedziczenie	73
3.23. Podsumowanie	74

## Rozdział 4. Typowe aplikacje internetowe

79

4.1. Wprowadzenie	79
4.2. Protokoły warstwy aplikacji	79
4.3. Reprezentacja i transfer danych	80
4.4. Protokoły WWW	81
4.5. Reprezentacja dokumentów w standardzie HTML	81
4.6. Ujednolicony format adresowania zasobów i odsyłacze	83
4.7. Dostarczanie dokumentów za pomocą protokołu HTTP	84
4.8. Buforowanie stron w przeglądarkach	87
4.9. Budowa przeglądarki	88
4.10. Protokół transferu plików (FTP)	88
4.11. Komunikacja FTP	89
4.12. Poczta elektroniczna	92
4.13. Prosty protokół dostarczania poczty (SMTP)	93
4.14. Dostawcy usług internetowych, serwery pocztowe i dostęp do poczty elektronicznej	95
4.15. Protokoły dostępu do poczty (POP, IMAP)	96
4.16. Standardy zapisu wiadomości e-mail (RFC2822, MIME)	97
4.17. System nazw domenowych (DNS)	98
4.18. Nazwy domenowe rozpoczynające się od www	100
4.19. Hierarchia DNS i model powiązań serwerowych	101
4.20. Odwzorowanie nazw	101
4.21. Buforowanie danych w systemie DNS	103
4.22. Rodzaje wpisów DNS	104
4.23. Aliasy nazw i rekordy CNAME	105
4.24. Skróty w systemie DNS	106
4.25. Znaki narodowe w nazwach domenowych	106
4.26. Rozszerzalne formaty reprezentacji danych (XML)	107
4.27. Podsumowanie	108

<b>Część II Wymiana danych</b>	<b>111</b>
Rozdział 5. Podstawowe informacje na temat transmisji danych	113
5.1. Wprowadzenie 113	
5.2. Istota transmisji danych 114	
5.3. Założenia i zakres zagadnienia 114	
5.4. Teoretyczne elementy systemu komunikacyjnego 115	
5.5. Elementy modelu transmisji danych 116	
5.6. Podsumowanie 118	
Rozdział 6. Sygnały i źródła informacji	121
6.1. Wprowadzenie 121	
6.2. Źródła informacji 121	
6.3. Sygnały analogowe i cyfrowe 122	
6.4. Sygnały okresowe i nieokresowe 122	
6.5. Przebieg sinusoidalny i cechy sygnału 123	
6.6. Sygnał zespolony 124	
6.7. Znaczenie sygnałów zespolonych i sinusoidalnych 125	
6.8. Reprezentacja sygnału w dziedzinie czasu i częstotliwości 126	
6.9. Szerokość pasma sygnału analogowego 127	
6.10. Sygnały cyfrowe i ich poziomy 127	
6.11. Body i bity na sekundę 129	
6.12. Przekształcenie sygnału cyfrowego w sygnał analogowy 130	
6.13. Szerokość pasma sygnału cyfrowego 131	
6.14. Synchronizacja i uzgodnienia odnośnie sygnałów 131	
6.15. Kodowanie liniowe 132	
6.16. Wykorzystanie kodowania Manchester w sieciach komputerowych 134	
6.17. Przekształcenie sygnału analogowego w sygnał cyfrowy 135	
6.18. Twierdzenie Nyquista i częstotliwość próbkowania 136	
6.19. Twierdzenie Nyquista w transmisji telefonicznej 137	
6.20. Kodowanie i kompresja danych 137	
6.21. Podsumowanie 138	
Rozdział 7. Media transmisyjne	141
7.1. Wprowadzenie 141	
7.2. Transmisja przewodowa i bezprzewodowa 141	
7.3. Podział ze względu na rodzaj energii 142	
7.4. Zakłócenia elektromagnetyczne i szum 142	
7.5. Skrętka miedziana 143	
7.6. Ekranowanie — kabel współosiowy oraz skrętka ekranywana 145	
7.7. Kategorie skrętek 146	
7.8. Media przenoszące energię światlną oraz włókna światłowodowe 146	
7.9. Rodzaje włókien i transmisji światłowodowych 148	
7.10. Porównanie włókien światłowodowych i kabli miedzianych 149	
7.11. Technologie komunikacji w podczerwieni 150	

7.12. Laserowa komunikacja punkt-punkt	150
7.13. Komunikacja z wykorzystaniem fal elektromagnetycznych (radiowa)	151
7.14. Propagacja sygnału	152
7.15. Rodzaje satelitów	153
7.16. Geostacjonarne satelity komunikacyjne	153
7.17. Pokrycie obszaru Ziemi przez satelity geostacjonarne	155
7.18. Satelity niskoorbitowe i ich klastry	156
7.19. Wybór medium transmisyjnego	156
7.20. Pomiary parametrów medium transmisyjnego	157
7.21. Wpływ szumu na komunikację	157
7.22. Znaczenie pojemności kanału	158
7.23. Podsumowanie	159

## Rozdział 8. Niezawodność i kodowanie kanałowe 163

8.1. Wprowadzenie	163
8.2. Trzy główne przyczyny błędów transmisyjnych	163
8.3. Wpływ błędów transmisyjnych na dane	164
8.4. Dwie strategie obsługi błędów	165
8.5. Kody blokowe i splotowe	166
8.6. Przykład kodu blokowego — pojedyncza kontrola parzystości	167
8.7. Matematyka kodów blokowych i notacja ( $n,k$ )	168
8.8. Odległość Hamminga — miara siły kodu	168
8.9. Odległość Hamminga między elementami książki kodowej	169
8.10. Kompromis między detekcją błędów a narzutem transmisyjnym	170
8.11. Korekcja błędów — parzystość wierszy i kolumn	170
8.12. 16-bitowa suma kontrolna stosowana w internecie	171
8.13. Cykliczny kod nadmiarowy (CRC)	173
8.14. Sprzętowa implementacja algorytmu CRC	175
8.15. Mechanizmy automatycznego powtarzania żądań (ARQ)	175
8.16. Podsumowanie	176

## Rozdział 9. Tryby transmisji danych 179

9.1. Wprowadzenie	179
9.2. Podział trybów transmisji danych	179
9.3. Transmisja równoległa	180
9.4. Transmisja szeregowa	181
9.5. Kolejność wysyłania bitów i bajtów	182
9.6. Zależności czasowe w transmisji szeregowej	182
9.7. Transmisja asynchroniczna	183
9.8. Asynchroniczna transmisja znaków — RS-232	183
9.9. Transmisja synchroniczna	184
9.10. Bajty, bloki i ramki	185
9.11. Transmisja izochroniczna	186
9.12. Simpleks, półsimpleks i duplex	186
9.13. Urządzenia DCE i DTE	187
9.14. Podsumowanie	188

<b>Rozdział 10. Modulacja i modemy</b>	<b>191</b>
10.1. Wprowadzenie	191
10.2. Częstotliwość, fala nośna i propagacja	191
10.3. Modulacja analogowa	192
10.4. Modulacja amplitudy	192
10.5. Modulacja częstotliwości	193
10.6. Modulacja fazy	194
10.7. Modulacja amplitudy i twierdzenie Shannona	194
10.8. Modulacja, sygnał cyfrowy i kluczowanie	194
10.9. Kluczowanie fazy	195
10.10. Przesunięcie fazowe i diagram konstelacji	195
10.11. Kwadraturowa modulacja amplitudy	198
10.12. Modem — urządzenie do modulacji i demodulacji	198
10.13. Modemy optyczne i radiowe	200
10.14. Modemy telefoniczne	200
10.15. Modulacja QAM w telefonii	201
10.16. Modemy V.32 i V.32bis	201
10.17. Podsumowanie	202
<b>Rozdział 11. Multipleksacja i demultipleksacja</b>	<b>205</b>
11.1. Wprowadzenie	205
11.2. Multipleksacja	205
11.3. Podstawowe rodzaje multipleksacji	206
11.4. Multipleksacja z podziałem częstotliwości (FDM)	206
11.5. Zakres częstotliwości w kanale komunikacyjnym	208
11.6. Hierarchia FDM	209
11.7. Multipleksacja z podziałem długości fali	210
11.8. Multipleksacja z podziałem czasu	211
11.9. Synchroniczne zwielokrotnienie TDM	211
11.10. Ramkowanie w telefonicznych systemach TDM	212
11.11. Hierarchia TDM	213
11.12. Wada synchronicznego systemu TDM — puste szczeliny czasowe	214
11.13. Statystyczny algorytm TDM	215
11.14. Odwrotna multipleksacja	216
11.15. Multipleksacja kodowa	216
11.16. Podsumowanie	218
<b>Rozdział 12. Technologie łączystości dostępowych i rdzeniowych</b>	<b>221</b>
12.1. Wprowadzenie	221
12.2. Dostęp do internetu	221
12.3. Wąskopasmowe i szerokopasmowe technologie dostępowe	222
12.4. Łącze abonenckie i ISDN	223
12.5. Technologie cyfrowych linii abonenckich (DSL)	224
12.6. Charakterystyka łącza abonenckiego i mechanizmy adaptacyjne	225
12.7. Przepustowość łączystości ADSL	226
12.8. Instalacja ADSL i filtry	227

12.9. Modemy kablowe	228
12.10. Przepustowość modemów kablowych	228
12.11. Instalacja modemu kablowego	229
12.12. Sieć HFC	229
12.13. Światłowodowe technologie dostępowe	230
12.14. Terminologia związana z modemami	231
12.15. Technologie dostępu bezprzewodowego	231
12.16. Wysokowydajne połączenia rdzenia internetowego	231
12.17. Zakończenie obwodu, moduły CSU/DSU i NIU	233
12.18. Standardy łączysty cyfrowych	234
12.19. Standardy DS i ich przepustowości	235
12.20. Obwody o największej pojemności (standardy STS)	235
12.21. Standardy łączysty optycznych	235
12.22. Sufiks C	236
12.23. Synchroniczna sieć optyczna (SONET)	236
12.24. Podsumowanie	238
<b>Część III Przełączanie pakietów i technologie sieci komputerowych</b>	<b>241</b>
<b>Rozdział 13. Sieci lokalne — pakiety, ramki, topologie</b>	<b>243</b>
13.1. Wprowadzenie	243
13.2. Przełączanie obwodów	243
13.3. Przełączanie pakietów	245
13.4. Rozległe sieci pakietowe	246
13.5. Standardy formatów i identyfikatorów pakietów	247
13.6. Model i standardy IEEE 802	248
13.7. Sieci punkt-punkt i wielodostępne	250
13.8. Topologie sieci LAN	250
13.9. Identyfikacja pakietów, demultipleksacja i adresy MAC	252
13.10. Adresy w emisji pojedynczej, multiemisji i w rozgłoszeniach	253
13.11. Rozgłoszenia, multiemisja i efektywne dostarczanie danych do wielu jednostek	254
13.12. Ramki i proces ich formowania	255
13.13. Nadziewanie bajtami i bitami	256
13.14. Podsumowanie	257
<b>Rozdział 14. Podwarstwa MAC</b>	<b>261</b>
14.1. Wprowadzenie	261
14.2. Podział mechanizmów regulujących dostęp do medium	261
14.3. Statyczna i dynamiczna alokacja kanałów	262
14.4. Protokoły alokacji kanałów	263
14.5. Protokoły sterowania dostępem	264
14.6. Protokoły dostępu swobodnego	266
14.7. Podsumowanie	272

<b>Rozdział 15. Przewodowe technologie LAN (Ethernet i 802.3)</b>	<b>275</b>
15.1. Wprowadzenie	275
15.2. Ethernet	275
15.3. Format ramki ethernetowej	276
15.4. Pole typu i demultipleksacja	276
15.5. Ethernet w wersji IEEE (802.3)	277
15.6. Połączenia sieci LAN i karty sieciowe	278
15.7. Rozwój Ethernetu — gruby Ethernet	278
15.8. Cienki Ethernet	279
15.9. Skrętka i koncentratory ethernetowe	280
15.10. Fizyczna i logiczna topologia Ethernetu	281
15.11. Okablowanie budynkowe	281
15.12. Odmiany okablowania i przepustowości	281
15.13. Złącza kabli ethernetowych	283
15.14. Podsumowanie	284
<b>Rozdział 16. Technologie sieci bezprzewodowych</b>	<b>287</b>
16.1. Wprowadzenie	287
16.2. Podział sieci bezprzewodowych	287
16.3. Sieci osobiste (PAN)	288
16.4. Pasmo ISM w sieciach LAN i PAN	288
16.5. Technologie bezprzewodowych sieci lokalnych i Wi-Fi	289
16.6. Techniki rozpraszania widma	290
16.7. Inne standardy bezprzewodowych sieci LAN	291
16.8. Architektura bezprzewodowej sieci LAN	292
16.9. Nakładanie obszarów, stwarzyszenie się urządzeń i format ramki 802.11	293
16.10. Koordynacja działań punktów dostępowych	293
16.11. Rywalizacja o dostęp i obsługa bezkolizyjna	294
16.12. Technologie bezprzewodowych sieci MAN i standard WiMAX	296
16.13. Technologie i standardy sieci PAN	298
16.14. Inne technologie komunikacji na niedużych odległościach	300
16.15. Technologie bezprzewodowych sieci WAN	300
16.16. Klastry komórek i wielokrotne wykorzystywanie częstotliwości	302
16.17. Generacje technologii komórkowych	303
16.18. Technologia satelitarna VSAT	306
16.19. Satelity GPS	307
16.20. Radio programowe i przyszłość technologii bezprzewodowych	308
16.21. Podsumowanie	309
<b>Rozdział 17. Rozszerzenie sieci LAN — modemy optyczne, regeneratorы, mosty i przełączniki</b>	<b>313</b>
17.1. Wprowadzenie	313
17.2. Budowa sieci LAN i ograniczenia w jej zasięgu	313
17.3. Modemy optyczne	314
17.4. Regeneratorы	315
17.5. Mosty	315

17.6. Filtrowanie ramek	316
17.7. Dlaczego warto używać mostów?	317
17.8. Rozproszone drzewo rozpinające	318
17.9. Przełączanie i przełączniki warstwy 2.	319
17.10. Przełączniki sieci VLAN	321
17.11. Funkcje mostu w innych urządzeniach	322
17.12. Podsumowanie	322
<b>Rozdział 18. Technologie sieci WAN i routing dynamiczny</b>	<b>325</b>
18.1. Wprowadzenie	325
18.2. Sieci rozległe	325
18.3. Tradycyjna architektura sieci WAN	326
18.4. Budowanie sieci WAN	327
18.5. Zasada „zapisz i przekaż”	328
18.6. Adresacja w sieciach WAN	329
18.7. Wyznaczanie następnego skoku	330
18.8. Niezależność od źródła	332
18.9. Dynamiczne aktualizacje informacji o routingu w sieci WAN	332
18.10. Trasy domyślne	333
18.11. Wypełnianie tablicy przekazywania	334
18.12. Rozproszone mechanizmy wyznaczania tras	335
18.13. Wyznaczenie najkrótszej trasy w grafie	337
18.14. Problemy routingu	340
18.15. Podsumowanie	340
<b>Rozdział 19. Technologie sieciowe — przeszłość i teraźniejszość</b>	<b>345</b>
19.1. Wprowadzenie	345
19.2. Technologie łączy dostępowych	345
19.3. Technologie sieci LAN	347
19.4. Technologie sieci WAN	349
19.5. Podsumowanie	352
<b>Część IV Sieci TCP/IP</b>	<b>353</b>
<b>Rozdział 20. Internet — koncepcje, architektura i protokoły</b>	<b>355</b>
20.1. Wprowadzenie	355
20.2. Przyczyny powstania internetu	355
20.3. Idea jednolitych usług	356
20.4. Jednolite usługi w heterogenicznym świecie	356
20.5. Internet	357
20.6. Fizyczne łączenie sieci za pomocą routerów	357
20.7. Architektura internetu	358
20.8. Wdrażanie jednolitych usług	359
20.9. Wirtualna sieć	359
20.10. Protokoły internetowe	361

<b>20.11. Warstwy stosu TCP/IP</b>	<b>361</b>
<b>20.12. Stacje sieciowe, routery i warstwy protokołów</b>	<b>362</b>
<b>20.13. Podsumowanie</b>	<b>362</b>
<b>Rozdział 21. IP — adresowanie w internecie</b>	<b>365</b>
21.1. Wprowadzenie	365
21.2. Adresy wirtualnego internetu	365
21.3. Schemat adresowania IP	366
21.4. Hierarchia adresów IP	367
21.5. Klasy adresów IP	367
21.6. Notacja dziesiętna z kropkami	368
21.7. Podział przestrzeni adresowej	369
21.8. Organizacje zarządzające przydziałem adresów	370
21.9. Adresowanie bezklasowe i podsieci	370
21.10. Maski adresów	371
21.11. Notacja CIDR	373
21.12. Przykład notacji CIDR	374
21.13. Adresy stacji w notacji CIDR	375
21.14. Adresy IP o specjalnym znaczeniu	375
21.15. Zestawienie adresów IP o specjalnym znaczeniu	378
21.16. Adres rozgłoszeniowy w formacie Berkeley	378
21.17. Routery i zasady adresowania IP	379
21.18. Stacje o wielu interfejsach sieciowych	380
21.19. Podsumowanie	380
<b>Rozdział 22. Przekazywanie datagramów</b>	<b>383</b>
22.1. Wprowadzenie	383
22.2. Usługa transmisji bezpołączeniowej	383
22.3. Wirtualne pakiety	384
22.4. Datagram IP	384
22.5. Format nagłówka datagramu IP	385
22.6. Przekazywanie datagramu IP	387
22.7. Odczytywanie prefiksów sieci i przekazywanie datagramów	388
22.8. Dopasowanie o najdłuższym prefiksie	389
22.9. Adres docelowe i adresy następnego skoku	389
22.10. Brak gwarancji dostarczenia datagramu	390
22.11. Enkapsulacja IP	391
22.12. Transmisja datagramu w internecie	391
22.13. MTU i fragmentowanie datagramu	393
22.14. Odtwarzanie datagramu z fragmentów	394
22.15. Rejestrowanie fragmentów datagramu	395
22.16. Konsekwencje utraty pakietu	395
22.17. Fragmentowanie fragmentów	396
22.18. Podsumowanie	397

<b>Rozdział 23. Protokoły i technologie uzupełniające</b>	<b>401</b>
23.1. Wprowadzenie	401
23.2. Odwzorowanie adresów	401
23.3. Protokół odwzorowania adresu (ARP)	403
23.4. Format komunikatu ARP	403
23.5. Enkapsulacja ARP	405
23.6. Buforowanie ARP i przetwarzanie komunikatów	406
23.7. Teoretyczna granica stosowania adresów	408
23.8. Internetowy protokół komunikacji sterujących (ICMP)	408
23.9. Format komunikatu i enkapsulacja ICMP	410
23.10. Oprogramowanie, parametry i konfiguracja protokołu	411
23.11. Protokół dynamicznej konfiguracji stacji (DHCP)	411
23.12. Działanie protokołu DHCP i optymalizacja pracy	413
23.13. Format komunikatu DHCP	414
23.14. Pośrednictwo w dostępie do serwera DHCP	415
23.15. Translacja adresów sieciowych (NAT)	415
23.16. Działanie usługi NAT i adresy prywatne	416
23.17. Translacja NAT na poziomie warstwy transportowej (NAPT)	418
23.18. Operacja NAT a dostęp do serwerów	419
23.19. Oprogramowanie NAT i systemy przeznaczone do sieci domowych	420
23.20. Podsumowanie	420
<b>Rozdział 24. Przyszłość protokołu IP (IPv6)</b>	<b>425</b>
24.1. Wprowadzenie	425
24.2. Sukces protokołu IP	425
24.3. Potrzeba zmian	426
24.4. Model klepsydry i trudności we wprowadzaniu zmian	427
24.5. Nazwa i numer wersji	428
24.6. Funkcje IPv6	428
24.7. Format datagramu IPv6	429
24.8. Format podstawowego nagłówka protokołu IPv6	429
24.9. Jawni i niejawny rozmiar nagłówka	431
24.10. Fragmentacja, odtwarzanie datagramów i MTU trasy	431
24.11. Przeznaczenie wielokrotnych nagłówków	433
24.12. Adresacja IPv6	434
24.13. Zapis adresów IPv6 w formacie szesnastkowym z dwukropkami	435
24.14. Podsumowanie	436
<b>Rozdział 25. UDP — usługa transportu datagramów</b>	<b>439</b>
25.1. Wprowadzenie	439
25.2. Protokoły transportowe i komunikacja między jednostkami końcowymi	439
25.3. Protokół datagramów użytkownika	440
25.4. Zasada komunikacji bezpołączeniowej	441
25.5. Przetwarzanie komunikatów	441
25.6. Przebieg komunikacji UDP	442
25.7. Rodzaje interakcji i dostarczanie rozgłoszeniowe	443

25.8. <i>Identyfikacja punktów końcowych za pomocą numerów portów</i>	444
25.9. <i>Format datagramu UDP</i>	444
25.10. <i>Suma kontrolna UDP i pseudonagłówek</i>	445
25.11. <i>Enkapsulacja komunikatu UDP</i>	445
25.12. <i>Podsumowanie</i>	446
<b>Rozdział 26. TCP — usługa niezawodnego transportu danych</b>	<b>449</b>
26.1. <i>Wprowadzenie</i>	449
26.2. <i>Protokół sterowania transmisją</i>	449
26.3. <i>Usługi TCP świadczone na rzecz aplikacji</i>	450
26.4. <i>Usługi aplikacji końcowych i połączenia wirtualne</i>	451
26.5. <i>Techniki wykorzystywane w pracy protokołów transportowych</i>	452
26.6. <i>Techniki unikania przeciążeń</i>	456
26.7. <i>Sztuka projektowania protokołu</i>	458
26.8. <i>Obsługa utraconych pakietów w protokole TCP</i>	458
26.9. <i>Adaptacyjne retransmisje</i>	460
26.10. <i>Porównanie czasów retransmisji</i>	460
26.11. <i>Bufory, sterowanie przepływem i okna</i>	461
26.12. <i>Trójetapowe porozumienie</i>	462
26.13. <i>Kontrola przeciążenia</i>	464
26.14. <i>Format segmentu TCP</i>	465
26.15. <i>Podsumowanie</i>	466
<b>Rozdział 27. Routing internetowy i protokoły routingu</b>	<b>469</b>
27.1. <i>Wprowadzenie</i>	469
27.2. <i>Routing statyczny a routing dynamiczny</i>	469
27.3. <i>Routing statyczny w komputerze i trasa domyślna</i>	470
27.4. <i>Routing dynamiczny i routery</i>	471
27.5. <i>Routing w globalnym internecie</i>	472
27.6. <i>Idea systemu autonomicznego</i>	473
27.7. <i>Dwa rodzaje protokołów routingu internetowego</i>	473
27.8. <i>Trasy i transport danych</i>	476
27.9. <i>Protokół bram granicznych (BGP)</i>	476
27.10. <i>Protokół informowania o trasach (RIP)</i>	478
27.11. <i>Format pakietu RIP</i>	479
27.12. <i>Otwarty protokół wyznaczania najkrótszych tras (OSPF)</i>	479
27.13. <i>Przykład grafu OSPF</i>	481
27.14. <i>Obszary OSPF</i>	482
27.15. <i>Protokół systemów pośrednich (IS-IS)</i>	482
27.16. <i>Routing w multiemisji</i>	483
27.17. <i>Podsumowanie</i>	487

<b>Część V Inne aspekty funkcjonowania sieci komputerowych</b>	<b>489</b>
<b>Rozdział 28. Wydajność sieci (QoS i DiffServ)</b>	<b>491</b>
28.1. Wprowadzenie	491
28.2. Miary wydajności	491
28.3. Opóźnienie	492
28.4. Przepustowość, pojemność i efektywna szybkość dostarczania danych	494
28.5. Zrozumienie przepustowości i opóźnienia	495
28.6. Fluktuacja opóźnienia	496
28.7. Zależność między opóźnieniem a przepustowością	497
28.8. Pomiar opóźnienia, przepustowości i fluktuacji opóźnienia	499
28.9. Pomiar pasywny, małe pakiety i mechanizm NetFlow	500
28.10. Jakość usługi (QoS)	501
28.11. Ogólna i szczegółowa specyfikacja QoS	502
28.12. Implementacja mechanizmów QoS	505
28.13. Internetowe technologie QoS	506
28.14. Podsumowanie	508
<b>Rozdział 29. Multimedia i telefonia IP (VoIP)</b>	<b>513</b>
29.1. Wprowadzenie	513
29.2. Transmisja w czasie rzeczywistym	513
29.3. Opóźnione odtwarzanie i bufor fluktuacji opóźnienia	514
29.4. Protokół transportowy czasu rzeczywistego (RTP)	515
29.5. Enkapsulacja RTP	516
29.6. Telefonia IP	517
29.7. Sygnalizacja i standardy sygnalizacji VoIP	518
29.8. Elementy składowe systemu telefonii IP	519
29.9. Podsumowanie protokołów i podział na warstwy	523
29.10. Charakterystyka protokołu H.323	523
29.11. Warstwy systemu H.323	524
29.12. Charakterystyka protokołu SIP	524
29.13. Przebieg sesji SIP	525
29.14. Odwzorowanie numerów telefonicznych i routing	525
29.15. Podsumowanie	527
<b>Rozdział 30. Bezpieczeństwo sieci</b>	<b>531</b>
30.1. Wprowadzenie	531
30.2. Działalność przestępca i ataki sieciowe	531
30.3. Polityka bezpieczeństwa	534
30.4. Odpowiedzialność za dane i nadzór nad nimi	536
30.5. Technologie związane z bezpieczeństwem	536
30.6. Generowanie skrótów — weryfikacja spójności danych i uwierzytelnianie	537
30.7. Kontrola dostępu i hasła	538
30.8. Szyfrowanie — podstawowa technika zabezpieczeń	538
30.9. Szyfrowanie z użyciem klucza prywatnego	539

30.10. Szyfrowanie z użyciem klucza publicznego	539
30.11. Uwierzytelnianie z wykorzystaniem podpisów cyfrowych	540
30.12. Organa zarządzające kluczami i certyfikaty cyfrowe	541
30.13. Zapory sieciowe	543
30.14. Zapory sieciowe z filtrowaniem pakietów	544
30.15. Systemy wykrywania włamań	545
30.16. Skanowanie treści i szczegółowa inspekcja pakietów	546
30.17. Wirtualne sieci prywatne (VPN)	547
30.18. Wykorzystanie technologii VPN w pracy zdalnej	549
30.19. Szyfrowanie pakietów a tunelowanie	550
30.20. Rozwiązania z zakresu bezpieczeństwa sieci	552
30.21. Podsumowanie	553
<b>Rozdział 31. Zarządzanie siecią (SNMP)</b>	<b>557</b>
31.1. Wprowadzenie	557
31.2. Zarządzanie intranetem	557
31.3. Model FCAPS	558
31.4. Przykładowe elementy sieci	560
31.5. Narzędzia do zarządzania sieciami	561
31.6. Aplikacje do zarządzania sieciami	562
31.7. Prosty protokół zarządzania sieciami	563
31.8. Zasada „pobierz-zapisz” w protokole SNMP	564
31.9. Baza MIB i nazwy obiektów	565
31.10. Różnorodność zmiennych MIB	565
31.11. Zmienne tablicowe w bazie MIB	566
31.12. Podsumowanie	567
<b>Rozdział 32. Trendy w technologiach sieciowych i sposobach wykorzystywania sieci</b>	<b>571</b>
32.1. Wprowadzenie	571
32.2. Zapotrzebowanie na skalowalne usługi internetowe	571
32.3. Buforowanie treści (Akamai)	572
32.4. Rozkładanie obciążenia serwerów WWW	572
32.5. Wirtualizacja serwerów	573
32.6. Komunikacja P2P	573
32.7. Rozproszone centra danych i replikacja	574
32.8. Jednolita reprezentacja danych (XML)	574
32.9. Sieci społecznościowe	575
32.10. Mobilność i sieci bezprzewodowe	575
32.11. Cyfrowy przekaz wideo	575
32.12. Multiemisja	576
32.13. Dostęp szerokopasmowy i przełączanie	576
32.14. Przełączanie optyczne	577
32.15. Sieć w biznesie	577
32.16. Czujniki w domu i otoczeniu	577
32.17. Sieci ad hoc	578

32.18. Procesory wielordzeniowe i sieciowe	578
32.19. IPv6	578
32.20. Podsumowanie	579
<b>Dodatek A Uproszczony interfejs programistyczny</b>	<b>581</b>
<i>Wprowadzenie</i>	581
<i>Model komunikacji sieciowej</i>	582
<i>Model klient-serwer</i>	582
<i>Zasady komunikacji</i>	582
<i>Przykładowy interfejs programistyczny</i>	583
<i>Intuicyjna praca z interfejsem API</i>	584
<i>Opis interfejsu API</i>	584
<i>Kod aplikacji echo</i>	588
<i>Kod serwera aplikacji echo</i>	589
<i>Kod klienta aplikacji echo</i>	590
<i>Kod serwera czatu</i>	592
<i>Aplikacja WWW</i>	597
<i>Kod klienta WWW</i>	597
<i>Kod serwera WWW</i>	599
<i>Obsługa wielu połączeń z użyciem funkcji select</i>	603
<i>Podsumowanie</i>	604
<b>Skorowidz</b>	<b>607</b>

# Przedmowa

Wcześniejsze wydania książki *Sieci komputerowe i intersieci* zyskały wiele bardzo pochlebnych recenzji. Chciałbym szczególnie podziękować czytelnikom, którzy znaleźli czas, żeby napisać do mnie osobiście. Oprócz studentów, korzystających z tego opracowania w ramach zajęć, słowa uznania wyrazili inżynierowie, którzy docenili jego przejrzystość oraz przydatność w zdawaniu egzaminów certyfikacyjnych. Wiele pozytywnych opinii zyskały również przekłady na inne języki. Sukces na rynku nasyconym książkami na temat sieci jest szczególnie satysfakcjonujący. Wyjątkowość tego opracowania wynika z szerokiego zakresu tematycznego, logicznej organizacji, jasności przekazu, odniesienia do internetu oraz przydatności zarówno dla profesorów, jak i studentów.

W odpowiedzi na sugestie czytelników oraz ostatnie zmiany w przemyśle sieciowym kolejna edycja została całkowicie zreorganizowana, poprawiona i uaktualniona. Opisy starszych technologii zostały skrócone lub usunięte. Z kolei informacje o transmisji danych, która jest podstawą wszystkich przedmiotów sieciowych, zostały rozszerzone i przeniesione do drugiej części książki. Rozdziały dotyczące sieci komputerowych bazują na informacjach odnośnie transmisji danych i prezentują zarówno sieci przewodowe, jak i bezprzewodowe. Omówienie rozwiązań bezprzewodowych uwzględnia nowe standardy 802.11, a także technologie wykorzystywane w telefonii komórkowej, która już dzisiaj oferuje usługi transmisji danych i będzie zmierzała do zaadaptowania protokołów internetowych.

Ostatnie dyskusje na temat sposobu prowadzenia zajęć z sieci komputerowych doprowadziły do debaty o tym, czy poszczególne zagadnienia należy prezentować w kolejności od dolnej do górnej warstwy modelu sieci, czy od górnej do dolnej. W pierwszym rozwiązaniu student najpierw poznaje szczegóły samej transmisji, a potem dowiaduje się, w jaki sposób kolejne warstwy stosują rozszerzając funkcjonalność systemu. Drugie podejście zakłada rozpoczęcie kursu od przedstawienia zasad działania wysokopoziomowych aplikacji i stopniowe uszczegóławianie ich pracy. W tej książce wykorzystano obydwie techniki. W pierwszej części przedstawione zostały programy sieciowe oraz ogólne zasady komunikacji w internecie. Dzięki temu studenci mogą zapoznać się ze sposobami wykorzystania internetu przed przystąpieniem do szczegółowej analizy technologii zapewniających działanie poszczególnych mechanizmów. W dalszej części prezentacja zagadnień jest zgodna z założeniem, że do zrozumienia zasad funkcjonowania nowych rozwiązań konieczne jest zapoznanie się z technologiami, na których bazie te rozwiązania zostały zbudowane.

Książka jest przeznaczona dla studentów lub absolwentów uczelni, którzy nie znają podstaw działania sieci lub mają w tym zakresie niewielką wiedzę. Nie ma tutaj wyrafi-

nowanych równań matematycznych, nie są również potrzebne informacje na temat budowy i funkcjonowania systemów operacyjnych. Poszczególne idee zostały wyjaśnione w sposób przejrzysty. Omówieniom towarzyszą liczne przykłady i rysunki, które ilustrują zasady działania określonych technologii, a rezultaty analizy są przedstawiane bez dowodzenia ich w sposób matematyczny.

Książka jest odpowiedzią na podstawowe pytanie: „W jaki sposób działają sieci komputerowe i internet?”. Jest kompletnym przewodnikiem, obejmującym wszystkie aspekty funkcjonowania sieci, od niskopoziomowych mechanizmów transmisji danych, przez okablowanie, budowę sieci LAN i W&N, protokoły pracy internetowej, po aplikacje sieciowe. Opisuje wykorzystanie urządzeń w pracy protokołów sieciowych oraz używanie stosu protokołów w działaniu aplikacji, którymi posługują się użytkownicy sieci.

Książka została podzielona na pięć części. Pierwsza z nich dotyczy technik korzystania z internetu i aplikacji sieciowych. Zawiera opisy warstw protokołów, interakcji w modelu klient-serwer i budowy interfejsu programistycznego gniazd. Prezentuje również przykłady protokołów warstwy aplikacji, które są stosowane w internecie.

Druga część (rozdziały 5. – 12.) jest poświęcona transmisji danych i dostarcza informacji na temat urządzeń sieciowych oraz rozwiązań stanowiących podstawę ich działania, takich jak modulacja, multipleksacja i kodowanie kanałowe. Obejmuje prezentację trybów transmisji oraz wyjaśnienie związań z nimi pojęć, w tym **szerokości pasma i szybkości transmisji**. Ostatni rozdział tej części zawiera omówienie technologii dostępowych oraz łączów międzysieciowych stosowanych w internecie, a także zasady implementowania rozwiązań przedstawionych we wcześniejszych rozdziałach.

Trzecia część (rozdziały 13. – 19.) odnosi się do przełączania pakietów i związań z nim technologii. Omówiono tutaj przyczyny stosowania transmisji pakietowych. Przedstawiono opracowany przez organizację IEEE model protokołów warstwy 2., a także przewodowe i bezprzewodowe technologie komunikacji sieciowej. W tej części znajduje się również opis czterech podstawowych rodzajów sieci (LAN, MAN, PAN i WAN) oraz routingu w sieciach WAN. Tematem ostatniego rozdziału są technologie sieciowe stosowane w internecie.

Czwarta część (rozdziały 20. – 27.) koncentruje się na protokołach internetowych. Po przedstawieniu korzyści wynikających z pracy internetowej opisano architekturę tego typu rozwiązań, routery, adresowanie, mechanizmy odwzorowania adresów oraz stos protokołów TCP/IP. Szczegółowo są w niej wyjaśnione zasady działania protokołów IP, TCP, UDP, ICMP i ARP. Dzięki nim studenci mogą się przekonać, jak rozwiązania teoretyczne sprawdzają się w praktyce. W rozdziale 26. przedstawiono ważne zagadnienie niezawodności protokołów transportowych (na przykładzie TCP).

Ostatnia część książki (rozdziały 28. – 32.) obejmuje zagadnienia, które są niezależne od konkretnych warstw stosu protokołów, w tym wydajność sieci, bezpieczeństwo sieci, zarządzanie sieciami oraz uruchamianie systemów i usług multimedialnych. W każdym przypadku potrzebne są informacje z wcześniejszych rozdziałów. Umieszczenie tych tematów na końcu książki jest zgodne z zasadą szczegółowego wyjaśniania mechanizmu przed omówieniem sposobu jego wykorzystania. Nie oznacza więc, że zagadnienia z końcowych rozdziałów są mniej istotne.

Książka doskonale sprawdza się jako podręcznik do jednosemestralnego kursu sieci komputerowych na każdym etapie studiów. Obejmuje zagadnienia od okablowania po aplikacje. Zawiera wiele zadań praktycznych, które wykładowcy mogą zlecać studentom. Na przykład na Uniwersytecie Purdue studenci mają cotygodniowe zajęcia laboratoryjne, które obejmują wiele prezentowanych tutaj zagadnień, od pomiaru parametrów sieciowych, przez analizę pakietów, po programowanie sieciowe. Po zakończeniu kursu każdy z nich musi wiedzieć, w jaki sposób router wykorzystuje tablicę routingu do przekazywania datagramów IP, w jaki sposób datagramy są przesyłane przez sieć, z jakich pól składa się ramka ethernetowa, w jaki sposób mechanizm TCP rozróżnia połączenia i w jaki sposób realizowane są jednocześnie połączenia z portem 80 równolegle pracujących serwerów WWW. Studenci muszą także umieć obliczyć czas trwania pojedynczego bitu w sieci ethernet, wyjaśnić, dlaczego protokół TCP jest klasyfikowany jako mechanizm transportu między punktami końcowymi oraz dlaczego w technologii DSL możliwe jest przesyłanie danych tymi samymi przewodami, w których transmitowane są sygnały telefonii analogowej.

Celem kursu jest ogólne, nie szczegółowe, zaznajomienie się z problematyką komunikacji sieciowej. Chcąc właściwie przedstawić tę tematykę, nie można się skupić na wybranych zagadnieniach lub kilku technologiach. Kluczem do sukcesu jest w tym przypadku narzucenie odpowiedniego tempa prezentacji materiału. Aby przerobić większość opisanych w książce tematów, można przyspieszyć omawianie zagadnień związanych z niższymi warstwami stosu protokołów (przedstawionymi w części II) i poświęcić po cztery tygodnie na budowę sieci i działanie internetu. Zostaną wówczas około dwa tygodnie na wprowadzenie i tematy uzupełniające, takie jak zarządzanie siecią i bezpieczeństwo. Zagadnienia związane z programowaniem gniazd można uwzględnić na zajęciach programowania.

Wykładowcy powinni kłaść szczególny nacisk na idee i zasady rozwiązywania problemów. Technologie zmieniają się co kilka lat, ale zasady komunikacji pozostają stałe. Muszą też wytworzyć taką atmosferę, by studenci byli podekscytowani możliwością zgłębiania tajników komunikacji sieciowej.

Choć żaden z tematów nie jest szczególnie trudny do zrozumienia, studenci mogą się czuć przytłoczeni ilością wiadomości. Muszą poznać wiele nowych terminów, a mnogość skrótów i sieciowy żargon nie ułatwiają tego zadania. Często upływa wiele czasu, zanim nauczą się poprawnego używania poszczególnych określeń. Na Uniwersytecie Purdue prowadzane są cotygodniowe quizy, które ułatwiają studentom zapamiętywanie nowo wprowadzanych pojęć.

Programowanie i testy laboratoryjne mają kluczowe znaczenie w zrozumieniu zagadnień sieciowych. Z tego powodu jedną z najważniejszych części kursu są ćwiczenia praktyczne. Na Uniwersytecie Purdue zajęcia rozpoczynają się od zadania utworzenia programu klienckiego, który łączy się z serwerem WWW i pobiera z niego określone dane (na przykład informacje o temperaturze). Do wykonania tego zadania bardzo przydatne są informacje zamieszczone w Dodatku A. Dodatek ten opisuje bowiem uproszczoną wersję interfejsu API, który umożliwia tworzenie aplikacji bez uprzedniego zapoznania się z protokołami, adresami, gniazdami oraz (dość trudnymi) mechanizmami interfejsu programistycznego gniazd. Biblioteka API jest dostępna na stronie towarzyszącej książce. Oczywiście, w trakcie semestru studenci poznają techniki programowania gniazd, a na końcu

umieją utworzyć wielowątkowy serwer WWW (obsługa skryptów serwerowych jest zadaniem opcjonalnym, ale większość studentów je wykonuje). Poza tym studenci wykorzystują laboratoria do przechwytywania pakietów z rzeczywistych sieci komputerowych, piszą programy dekodujące nagłówki protokołów (na przykład nagłówki ethernetowe, TCP i UDP) oraz monitorują połączenia TCP. W przypadku braku stosowanego wyposażenia laboratorium można wykorzystać darmowe oprogramowanie, takie jak **Wireshark**.

Zapewnienie studentom dostępu do sieci użytkowej wzmagają entuzjazm i zachęca do eksperymentowania. Nasze doświadczenia są potwierdzeniem tego, że studenci, którzy mają dostęp do prawdziwej sieci, szybciej przyswajają sobie wiedzę z zakresu komunikacji sieciowej. Jeśli więc dana uczelnia nie dysponuje analizatorem pakietów, można go łatwo stworzyć przez zainstalowanie odpowiedniego oprogramowania w standardowym komputerze PC.

Twarzyszący książce serwis internetowy zawiera materiały, które ułatwiają pracę wykładowcy i pomagają czytelnikom w zrozumieniu przedstawianych tutaj zagadnień. Studenci, którzy nie mają dostępu do sieci, mogą na niej znaleźć również przykładowe, zarejestrowane wcześniej pakiety. Po napisaniu programu, który je odczyta, można je analizować w taki sam sposób, jakby zostały przechwycone w sieci. Dla wykładowców zamieszczono tam materiały do zajęć, rysunki i opracowania, które można wykorzystać na prezentacjach, oraz animacje pozwalające na czytelne przedstawienie wielu zagadnień. Na stronie dostępnych jest również wiele materiałów dodatkowych, które nie zostały zamieszczone w książce. Są to fotografie okablowania sieciowego i sprzętu, a także pliki z danymi, które można wykorzystać w projektach studenckich. Adres strony to:



<http://www.netbook.cs.purdue.edu>

Chciałbym podziękować wszystkim osobom, które przyczyniły się do powstania tej książki. Fred Baker i Dave Oran z firmy Cisco zasugerowali najważniejsze tematy. Lami Kaya zaproponowała strukturę książki, pomogła mi opracować treść związaną z transmisją danych, sprawdziła tekst i dostarczyła wielu wartościowych sugestii. Lami zarządzła również witrynę internetową związaną z książką. Chcę też podziękować mojej żonie i współpracowniczce Christine, której praca edytorska i liczne sugestie udoskonaliły wiele fragmentów opracowania.

Douglas E. Comer

marzec 2008

## O autorze

Dr Douglas Comer jest międzynarodowym autorytetem w dziedzinie protokołów TCP/IP, sieci komputerowych i internetu. Jako naukowiec ma swój wkład w powstanie internetu, ponieważ w latach 70. i 80. ubiegłego wieku był członkiem organizacji Internet Architecture Board — grupy odpowiedzialnej za kierowanie pracami nad rozwojem internetu. Był także przewodniczącym komitetu technicznego CSNET, członkiem komitetu wykonawczego CSNET oraz przewodniczącym Rady ds. Architektury Systemów Rozproszonych w agencji DARPA.

Douglas Comer był także konsultantem w wielu projektach sieciowych. Poza zajęciami na amerykańskich uczelniach, corocznie prowadzi wykłady przeznaczone dla nauczycieli akademickich i inżynierów sieciowych z całego świata. Zaprojektowany przez niego system operacyjny Xinu oraz implementacja protokołów TCP/IP są stosowane w wielu produktach komercyjnych.

Douglas Comer jest profesorem informatyki w Uniwersytecie Purdue. Będąc na urlopie, pełni funkcję wiceprezesa działu współpracy badawczej w Cisco Systems. Ostatnio prowadził kursy na temat sieci, internetu, architektury komputerów i systemów operacyjnych. Zaprojektował nowoczesne laboratorium, w którym studenci mają możliwość eksperymentowania z systemami operacyjnymi, sieciami komputerowymi i protokołami. Oprócz napisania wielu bestsellerowych książek, które zostały przetłumaczone na szesnaście języków, od dwudziestu lat pełni funkcję wydawcy magazynu „Software — Practice and Experience”. Jest również członkiem stowarzyszenia Association for Computing Machinery.

Więcej informacji na temat autora można znaleźć na stronie:

*[www.cs.purdue.edu/people/comer](http://www.cs.purdue.edu/people/comer)*



# **Entuzjastyczne komentarze na temat książki „Sieci komputerowe i intersieci”**

„To jest jedna z najlepszych książek, jakie kiedykolwiek czytałem. Dziękuję”.

*Gokhan Mutlu  
Uniwersytet Ege, Turcja*

„Nie mogłem przestać czytać. Jest po prostu świetna”.

*Lalit Y. Raju  
Regional Engineering College, Indie*

„Doskonała książka zarówno dla początkujących, jak i dla profesjonalistów — dobrze napisana, o szerokim zakresie tematycznym i dużej łatwości przyswajania zagadnień”.

*John Lin  
Bell Labs*

„Zakres tematyczny jest zadziwiająco szeroki”.

*George Varghese  
University of California w San Diego*

„To naprawdę najlepsza książka z tej dziedziny, jaką kiedykolwiek czytałem. Wielkie dzięki!”

*Chez Ciechanowicz  
Info. Security Group, University of London*

„Przedstawiony w dodatku 1 uproszczony serwer WWW jest doskonały — czytelnicy będą nim zachwyceni”.

*Dennis Brylow  
Marquette University*

„Wow, co za fantastyczna książka”.

*Jaffet A. Cordoba  
Autor opracowań technicznych*

„Książka jest świetna!”

*Peter Parry  
South Birmingham College, Wielka Brytania*

„Gdy przygotowywałem się do egzaminu CCNA, miałem problemy ze zrozumieniem modelu OSI i transmisji danych z wykorzystaniem mechanizmów TCP/IP. W tej książce znalazłem przejrzyste wyjaśnienie wspomnianych kwestii. Ona otworzyła mój umysł na fascynujący świat sieci i protokołów TCP/IP”.

*Solomon Tang  
PCCW, Hongkong*

„Nieocenione narzędzie, szczególnie dla programistów i informatyków, którzy szukają przejrzystych i kompleksowych informacji na temat sieci komputerowych”.

*Peter Chuks Obiefuna  
East Carolina University*

„Książka obejmuje wiele zagadnień, które autor przedstawia w sposób łatwy do przeczytania i zrozumienia. Jest to główny powód, dla którego ją kupiłem. Wydaje się doskonała dla studentów z grup zaawansowanych, ponieważ zawiera dużo materiału. Pozytywny odbiór zajęć ze strony studentów również dowodzi jej przydatności”.

*Jie Hu  
Saint Cloud State University*

„Mimo ogromnej liczby akronimów, które są charakterystyczne dla tej dziedziny nauki, książka nie odstrasza czytelnika. Comer jest doskonałym autorem, który rozwija i wyjaśnia terminologię. Opracowanie obejmuje cały zakres zagadnień związanych z sieciami komputerowymi, od przewodów po serwery WWW. Uważam, że jest znakomite”.

*Jennifer Seitzer  
University of Dayton*

# **CZĘŚĆ I**

## **Wprowadzenie do sieci komputerowych i aplikacji internetowych**

**Przegląd rozwiązań sieciowych  
oraz interfejsów programistycznych  
wykorzystywanych  
w komunikacji internetowej.**

### **Rozdziały:**

Rozdział 1. Wprowadzenie	29
Rozdział 2. Kierunki rozwoju internetu	45
Rozdział 3. Aplikacje internetowe i programowanie sieciowe	55
Rozdział 4. Typowe aplikacje internetowe	79

## **Zawartość rozdziału**

1.1.	Rozwój sieci komputerowych	29
1.2.	Dlaczego komunikacja sieciowa wydaje się trudna?	30
1.3.	Pięć kluczowych zagadnień sieciowych	30
1.4.	Publiczne i prywatne obszary internetu	34
1.5.	Sieci, współdziałanie i standardy	36
1.6.	Stos protokołów i modele warstwowe	37
1.7.	Przekazywanie danych między warstwami	39
1.8.	Nagłówki i warstwy	40
1.9.	Organizacja ISO i siedmiowarstwowy model odniesienia OSI	40
1.10.	Kulisy standaryzacji	41
1.11.	Pozostała część książki	42
1.12.	Podsumowanie	43

# 1

## Wprowadzenie

### 1.1. Rozwój sieci komputerowych

Sieci komputerowe rozwijają się bardzo gwałtownie. Od lat 70. ubiegłego stulecia komunikacja między komputerami przeszła drogę od mało istotnego tematu badawczego do jednego z głównych komponentów infrastruktury teleinformatycznej. Praca sieciowa jest elementem niemal każdej działalności biznesowej, od marketingu, przez produkcję, spełnianie, planowanie, rozliczenia, aż po księgowość. Z tego względu znaczna liczba przedsiębiorstw korzysta z wielu różnych sieci komputerowych. Sieci komputerowe znajdują również zastosowanie w szkolnictwie, na wszystkich jego poziomach — od podstawowego do wyższego. Dzięki nim uczniowie, studenci i nauczyciele mają stały dostęp do bieżących informacji. Z tego samego powodu sieci komputerowe są instalowane w biurach agencji rządowych oraz jednostkach wojskowych. Są one po prostu wszędzie.

Rozwój i wzrost zastosowań internetu należą do najbardziej interesujących i ekscytujących zagadnień związanych z komunikacją sieciową. Początki internetu datuje się na rok 1980, w którym powstał projekt badawczy obejmujący kilkadziesiąt serwisów sieciowych. Obecnie rozwiązanie to jest dojrzałym systemem komunikacyjnym, obejmującym swym zasięgiem wszystkie zamieszkałe regiony świata. Wielu użytkowników sieci korzysta z dostępu do internetu realizowanego na bazie technologii DSL, modemów kablowych oraz łączności bezprzewodowej.

Dostępność i użyteczność komunikacji sieciowej ma ogromny wpływ na światową gospodarkę. Możliwość przesyłania danych przez sieć znacznie ułatwia zdalną pracę i zmieniła sposób wymiany informacji w biznesie. Ponadto gwałtownie wzrosło zapotrzebowanie na nowe technologie, produkty i usługi internetowe. Większe znaczenie komunikacji sieciowej oznacza również nowe stanowiska pracy dla osób ze znajomością tych zagadnień. Firmy potrzebują bowiem pracowników, którzy będą umieli zaplanować, wdrożyć i utrzymać systemy sprzętowe i programowe składające się na sieci lokalne i szkieletowe. Zmiany są również widoczne w branży programistycznej. Tworzenie aplikacji nie jest już

ograniczone do pojedynczego komputera. Programiści muszą mieć wiedzę na temat programowania sieciowego, ponieważ coraz częściej tworzone przez nich programy muszą się komunikować z aplikacjami zainstalowanymi w innych systemach.

## 1.2. Dlaczego komunikacja sieciowa wydaje się trudna?

Zagadnienia związane z komunikacją sieciową wydają się skomplikowane, ponieważ należą do niezwykle bogatej i dynamicznej dziedziny techniki. Istnieje wiele różnych rozwiązań pracy sieciowej, a każda technologia różni się w pewien sposób od pozostałych. Firmy z branży internetowej bezustannie opracowują nowe produkty i usługi, często wykorzystując standardowe technologie w nowatorski sposób. Cała dziedzina może się więc wydawać niezwykle złożona, gdyż pozwala na łączenie ze sobą różnych wcześniejszych rozwiązań.

Komunikacja sieciowa sprawia szczególne trudności osobom, które dopiero rozpoczynają pracę z systemami internetowymi. Wynika to przede wszystkim z braku jednolitej teorii, wyjaśniającej wszystkie zależności między poszczególnymi elementami składowymi rozwiązania. Wiele organizacji opracowuje standardy sieciowej wymiany danych, ale część z tych opracowań jest niezgodna z pozostałymi standardami. Część organizacji i grup badawczych zajmuje się także przygotowywaniem modeli sieciowych, które mają oddawać ideę pracy sieciowej i wyjaśniać niuanse implementacyjne występujące między poszczególnymi systemami sprzętowymi i programowymi. Jednak z uwagi na różnorodność rozwiązań i nieustannie zmieniające się technologie, wspomniane modele są albo zbyt ogólne, by mogły precyzyjnie wyjaśniać różnice w budowie poszczególnych mechanizmów, albo zbyt złożone, by przedstawić zagadnienie w przystępny sposób.

Kolejnym wyzwaniem dla początkujących użytkowników sieci jest brak jednolitej terminologii. Zamiast stosować spójne nazewnictwo, poszczególne grupy osób posługują się odmiennymi określeniami. Badacze starają się zachować naukową precyzję terminologiczną. Przedstawiciele producentów często kojarzą produkt z ogólnym terminem technicznym lub wymyślają nowe nazwy, które pozwolą odróżnić dane urządzenie lub usługę od analogicznych wytworów konkurencji. Z kolei określenia stosowane przez techników bardzo często wynikają z nazw popularnych produktów sieciowych. Zamieszanie potęguje fakt stosowania terminów właściwych dla określonej technologii do opisu analogicznych cech innej technologii. W rezultacie bogaty zbiór terminologii oraz liczne akronimy (często o podobnym znaczeniu) sprawiają, że żargon administratorów sieci często niedokładnie oddaje rzeczywistość, zawiera skróty oraz nazwy własne produktów.

## 1.3. Pięć kluczowych zagadnień sieciowych

Aby dokładnie zrozumieć zawiłości pracy sieciowej, trzeba zapoznać się z jej podstawowymi zagadnieniami, obejmującymi pięć najważniejszych elementów rozwiązań sieciowych:

- aplikacje sieciowe i programowanie sieciowe,
- przekazywanie danych,

- przełączanie pakietów i technologie sieciowe,
- praca sieciowa z wykorzystaniem protokołów TCP/IP,
- dodatkowe koncepcje i technologie sieciowe.

### 1.3.1. Aplikacje sieciowe i programowanie sieciowe

Wszystkie usługi sieciowe wywoływane przez użytkowników są udostępniane przez odpowiednie oprogramowanie — aplikacja uruchomiona na jednym komputerze komunikuje się za pomocą sieci z aplikacją działającą w systemie drugiego komputera. Usługami sieciowymi są na przykład: dostarczanie poczty elektronicznej, transfer plików, przeglądanie stron WWW, prowadzenie rozmów telefonicznych, korzystanie z rozproszonych baz danych, a także prowadzenie telekonferencji (w tym wideokonferencji). Mimo że każda z aplikacji zapewnia dostęp do innej usługi za pomocą własnego interfejsu użytkownika, wszystkie aplikacje korzystają z tej samej wspólnej sieci komputerowej. Dostępność jednolitego mechanizmu transportu danych (niezależnego od rodzaju aplikacji) znacznie ułatwia pracę programistów, którzy muszą znać jedynie zasady działania interfejsu sieciowego oraz umieć posługiwać się pewnym zbiorem podstawowych funkcji sieciowych — we wszystkich programach wykorzystujących do komunikacji sieć komputerową używane są te same zbiory funkcji.

Jak będzie się można wkrótce przekonać, możliwe jest poznanie zasad działania aplikacji sieciowej lub nawet napisanie programu komunikującego się przez sieć bez gruntowej znajomości technologii sprzętowych i programistycznych stosowanych do przekazywania danych między jednostkami. Wydaje się więc, że perfekcyjne opanowanie interfejsu komunikacyjnego zwalnia twórcę aplikacji z obowiązku zgłębiania tajników funkcjonowania sieci. Programowanie sieciowe jest bardzo zbliżone do przygotowywania klasycznych aplikacji komputerowych. I choć istnieje możliwość opracowania programu bez uprzedniego poznania kompilatora, systemu operacyjnego oraz architektury komputera, zgromadzenie informacji na ich temat pozwala na przygotowanie produktu, który będzie bardziej niezawodny i wydajny. Podobnie zaznajomienie się z podstawami działania komponentów sieciowych daje gwarancję tworzenia lepszego oprogramowania sieciowego. Można to podsumować w następujący sposób:

*Programista rozumiejący technologie i mechanizmy komunikacji sieciowej może tworzyć bardziej niezawodne i wydajne aplikacje sieciowe.*

### 1.3.2. Transmisja danych

Termin **transmisja danych** (ang. *data communication*) odnosi się do niskopoziomowych mechanizmów i technologii przesyłania informacji w fizycznym medium transmisyjnym, takim jak przewód, fale radiowe lub światło. Przekazywanie danych jest przede wszystkim domeną inżynierii elektrycznej, która obejmuje problematykę projektowania i budowania systemów komunikacyjnych o dużym zasięgu. Z tego względu większość podstawowych

zależności w tej dziedzinie wynika z właściwości energetycznych systemów, które są analizowane przez fizyków. Na przykład możliwość transmitowania informacji we włóknie światłowodowym zależy od cech charakterystycznych światła oraz zjawiska odbijania promieni na granicy dwóch różnych ośrodków.

Ponieważ transmisja danych zależy od właściwości fizycznych toru, może się wydawać mało istotna w zrozumieniu zasad działania sieci. Fakt, że większość terminów i koncepcji odnosi się do zjawisk fizycznych, sprawia, że wydają się one istotne jedynie dla inżynierów projektujących niskopoziomowe komponenty transmisyjne. Na przykład techniki modulacji, które wykorzystują fizyczne formy energii (np. promieniowanie elektromagnetyczne) do przenoszenia informacji, wydają się nieistotne podczas projektowania i wykorzystywania protokołów komunikacyjnych. Przekonamy się jednak, że kilka kluczowych rozwiązań z dziedziny przekazywania danych ma istotny wpływ na budowę wielu warstw protokołów. W przypadku modulacji mamy bowiem do czynienia z pojęciem szerokości pasma, które bezpośrednio wiąże się z przepustowością sieciową.

Mechanizmy przekazywania danych obejmują również operację multipleksacji, która umożliwia przekazywanie informacji pochodzących z różnych źródeł w ramach wspólnego medium transmisyjnego i zapewnia także wydzielenie strumieni adresowanych do poszczególnych odbiorców. Przekonamy się, że multipleksacja nie występuje jedynie w modułach fizycznej transmisji danych — idea ta jest wykorzystywana w większości protokołów komunikacyjnych. Podobnie pojęcie szyfrowania, charakterystyczne dla przekazywania danych, jest podstawą większości rozwiązań gwarantujących bezpieczeństwo sieci komputerowych. Znaczenie tej warstwy można podsumować w następujący sposób:

*Mechanizmy przekazywania danych, choć odnoszą się do wielu niskopoziomowych szczegółów implementacyjnych, obejmują rozwiązania, na których bazują pozostałe komponenty systemów sieciowych.*

### 1.3.3. Przełączanie pakietów i technologie sieciowe

W latach 60. ubiegłego wieku przekazywanie danych zrewolucjonizowała nowa idea — przełączanie pakietów. Pierwsze sieci teletransmisyjne wywodziły się z systemów telegraficznych i telefonicznych, w których do utworzenia kanału komunikacyjnego konieczne było fizyczne połączenie dwóch urządzeń końcowych za pomocą pary przewodów. I choć z czasem mechaniczne łączenie przewodów zostało zastąpione przez przełączanie elektroniczne, zasadnicza koncepcja wymiany informacji pozostała niezmieniona — należało utworzyć obwód komunikacyjny i przesyłać za jego pomocą odpowiednie informacje. Technika przełączania pakietów istotnie zmieniła sieciową wymianę danych i obecnie stanowi podstawę działania internetu. Zamiast ustanawiania dedykowanego obwodu dane pochodzące od wielu użytkowników są przekazywane w ramach wspólnych połączeń sieciowych. Przełączanie pakietów bazuje na tych samych mechanizmach przekazywania danych, które stanowią podstawę funkcjonowania telefonii, ale wykorzystuje istniejące rozwiązania w nowy sposób. Rozwiązywanie to polega bowiem na dzieleniu zbioru danych na mniejsze fragmenty nazywane pakietami i dołączaniu do każdego pakietu identyfi-

katora odbiorcy. Gdy tak sformowany pakiet dotrze do urządzenia sieciowego, urządzenie to może wybrać odpowiednią trasę, gwarantującą, że dane zostaną ostatecznie dostarczone do wskazanego odbiorcy.

W teorii przełączanie pakietów wydaje się nieskomplikowane. W praktyce jednak stosowanych jest wiele różnych rozwiązań. Wybór odpowiedniego zależy od odpowiedzi, jakie zostaną udzielone na kilka kluczowych dla tego mechanizmu pytań. W jaki sposób oznaczane są jednostki docelowe oraz w jaki sposób nadawca może ustalić adres systemu docelowego? W jaki sposób urządzenia sieciowe mogą rozpoznać koniec pakietu i początek kolejnego? W jaki sposób gwarantowany jest równy dostęp do sieci większej liczby komputerów podłączonych do sieci? Czy istnieje możliwość zaadaptowania technik pakietowych do transmisji bezprzewodowej? Czy dane technologie sieciowe gwarantują odpowiednią wydajność, zasięg oraz czy ich stosowanie jest uzasadnione ekonomicznie? Na te pytania można udzielić różnych odpowiedzi w zależności od rodzaju wykorzystywanej technologii pakietowej. Większość osób zgłębiających problematykę transmisji pakietowej dochodzi do wspólnego wniosku:

*Różnorodność dostępnych technologii pakietowych wynika z tego, że każda z nich spełnia inne założenia odnośnie szybkości transmisji, zasięgu działania i opłacalności ekonomicznej. Poszczególne rozwiązania różnią się od siebie szczegółami implementacyjnymi, takimi jak rozmiar pakietu lub sposób identyfikacji odbiorcy.*

#### 1.3.4. Praca sieciowa z wykorzystaniem protokołów TCP/IP

W latach 70. wybuchła kolejna rewolucja w świecie sieci komputerowych. Powstał internet. Wielu badaczy zajmujących się technologiami pakietowymi pracowało nad utworzeniem jednego spójnego rozwiązania, które znalazłyby zastosowanie we wszystkich systemach sieciowych. W 1973 roku Vinton Cerf i Robert Kahn doszli do wniosku, że nie jest możliwe opracowanie takiego mechanizmu, który spełniłby wszystkie założenia, szczególnie jeśli uwzględnione w nich zostaną rozwiązania o niskiej wydajności i wyjątkowo niskiej cenie, przeznaczone do domów i biur. Zasugerowali wówczas, żeby odstąpić od poszukiwania jednej uniwersalnej technologii na rzecz łączenia różnych technik przełączania pakietów w jeden system. Zaproponowali opracowanie zbioru standardów opisujących połączenie różnych mechanizmów, co w konsekwencji doprowadziło do powstania **stosu protokołów internetowych TCP/IP** (skrótnie określanych mianem stosu TCP/IP). Idea, znana obecnie jako **sieć internetowa**, okazała się doskonałym rozwiązaniem i stanowi bardzo istotny element sieci komputerowych.

Główną przyczyną sukcesu standardów TCP/IP jest to, że działają one w sieciach heterogenicznych. Nie narzucają konkretnych ustawień mechanizmów przełączania pakietów (takich jak rozmiar pakietu lub metoda identyfikacji jednostki docelowej), lecz wirtualizują pakiety (uniezależniając je od konkretnej implementacji) oraz systemy identyfikacji odbiorców, a następnie odwzorowują wspomniane wirtualne pakiety na struktury charakterystyczne dla poszczególnych sieci, w których działają.

Zdolność stosu TCP/IP do obsługiwanego nowych sieci pakietowych jest główną przyczyną ciągłego rozwoju technologii pakietowych. Wraz ze wzrostem popularności internetu wzrasta wydajność komputerów, a aplikacje wymieniają coraz więcej danych, w tym również graficznych i audiowizualnych. Aby nadążyć za zwiększającym się wykorzystaniem sieci, inżynierowie opracowują coraz to nowsze technologie, które pozwalają na przesyłanie większej ilości danych i przetwarzanie większej liczby pakietów w jednostce czasu. Nowe rozwiązania są natychmiast wdrażane w internecie i współdziałają z dotychczasowymi mechanizmami. Jest to możliwe dzięki heterogeniczności sieci internetowych. Inżynierowie mogą więc eksperymentować z nowymi technikami bez zakłócania pracy istniejącej sieci. Podsumowując:

*Internet łączy wiele niezależnych sieci pakietowych, a powstała sieć internetowa jest znacznie efektywniejsza niż pojedyncze rozwiązanie sieciowe, ponieważ umożliwia wdrażanie nowych rozwiązań bez konieczności wymiany wcześniejszych komponentów.*

## 1.4. Publiczne i prywatne obszary internetu

Mimo że internet funkcjonuje jako spójny system komunikacyjny, składa się z wielu obszarów, które są zarządzane przez organizacje i osoby prywatne będące właścicielami określonej części sieci. Aby uściślić przynależność oraz przeznaczenie poszczególnych obszarów, wykorzystuje się określenia **sieć publiczna** oraz **sieć prywatna**.

### 1.4.1. Sieć publiczna

**Sieć publiczna** jest usługą dostępną dla subskrybentów. Mogą z niej korzystać osoby prywatne oraz przedsiębiorstwa wnoszące stosowne opłaty abonamentowe. Firma udostępniająca usługi komunikacyjne jest nazywana **dostawcą usług**, a w szczególnym przypadku **dostawcą usług internetowych** (ISP — ang. *Internet Service Provider*). Nazewnictwo pochodzi jeszcze z czasów, w których firmy telekomunikacyjne oferowały usługi telefonii analogowej. Niemniej prawdziwe jest stwierdzenie, że:

*Sieć publiczna pozostaje pod kontrolą dostawcy usług, który oferuje swoje usługi osobom prywatnym i firmom opłacającym abonament.*

Oczywiście, określenie **publiczna** odnosi się do ogólnej dostępności usługi, a nie do braku zabezpieczeń w przesyłaniu danych. W praktyce sieci publiczne często muszą spełniać bardzo rygorystyczne normy, które nakazują dostawcy usług internetowych ochronę informacji przed przypadkowym ujawnieniem.

*Określenie **publiczna** oznacza, że usługa jest ogólnie dostępna, a przekazywane przez sieć dane nie są ujawniane osobom postronnym.*

### 1.4.2. Sieć prywatna

**Sieć prywatna** pozostaje pod kontrolą określonej grupy użytkowników. Choć mogłoby się wydawać, że podział internetu na część publiczną i prywatną jest oczywisty, jednak wyznaczenie granicy tego podziału może się okazać dość trudne. Kontrolowanie sieci nie zawsze jest bowiem związane z posiadaniem praw własności do niej. Na przykład jeśli określona firma dzierżawi łącze od dostawcy usług internetowych w celu przekazywania wewnętrznych informacji, łącze to należy do sieci prywatnej przedsiębiorstwa.

*Sieć jest prywatna, jeśli korzysta z niej tylko jedna organizacja. W sieci prywatnej mogą być zatem stosowane łącza dzierżawione od dostawcy usług internetowych.*

Producenci urządzeń sieciowych dzielą sieci prywatne na cztery kategorie:

- sieci odbiorców prywatnych,
- sieci małych (domowych) biur (SOHO — ang. *Small Office/Home Office*),
- sieci małych i średnich przedsiębiorstw (SMB — ang. *Small-To-Medium Business*),
- sieci korporacyjne.

Ponieważ przynależność do poszczególnych kategorii wynika przede wszystkim z parametrów marketingowych, nazwy kategorii nie są ściśle zdefiniowane. Można jednak wyróżnić pewne cechy charakterystyczne każdego z typów rozwiązań (mimo braku jednoznacznej definicji). W kolejnych akapitach zostały więc zamieszczone opisy zasięgu i przeznaczenia, a nie parametry techniczne poszczególnych rozwiązań.

**Sieć odbiorcy prywatnego.** Sieci należące do osób prywatnych są zazwyczaj najtańszymi formami sieci LAN. Powstają na przykład wówczas, gdy użytkownik kupi podstawową wersję przełącznika LAN i wykorzysta go do połączenia komputera z drukarką. Instalacje tego typu bazują też często na routeraх bezprzewodowych.

**Sieci małych biur (SOHO).** Sieci SOHO mają nieznacznie większy rozmiar od sieci odbiorców prywatnych i składają się z co najmniej dwóch komputerów, jednej drukarki lub większej ich liczby, routera internetowego, a także innych urządzeń, jak na przykład kasę fiskalną. Większość systemów SOHO obejmuje układy podtrzymywania zasilania oraz inne komponenty gwarantujące nieprzerwaną pracę sieci.

**Sieci małych i średnich przedsiębiorstw (SMB).** Sieć SMB łączy zazwyczaj wiele komputerów rozmieszczonych w wielu biurach jednego budynku, a także jednostki w działach produkcyjnych (na przykład w dziale spedycji). Sieci SMB składają się zazwyczaj z wielu przełączników warstwy 2. połączonych za pomocą routera, wykorzystując szerokodostępowe łącze internetowe i mogą obejmować również punkty dostępu bezprzewodowego.

**Sieci korporacyjne.** Sieci korporacyjne wyznaczają infrastrukturę IT w dużych przedsiębiorstwach. Zazwyczaj obejmują kilka lokalizacji geograficznych z wieloma budynkami w każdej z nich. Bazują na przełącznikach warstwy 2. i routeraх, a także wielu łączach internetowych o wysokiej przepustowości. Często wykorzystuje się w nich zarówno przewodowe, jak i bezprzewodowe technologie komunikacyjne.

Ogólnie rzecz ujmując:

*Sieć prywatna może obejmować sieć domową, sieć małego lub średniego biura bądź sieć korporacyjną.*

## 1.5. Sieci, współdziałanie i standardy

Komunikacja zawsze wymaga współdziałania co najmniej dwóch jednostek — nadajnika i odbiornika informacji. Choć w praktyce zazwyczaj konieczne jest również zastosowanie komponentów pośrednich (tj. urządzeń przekazujących pakiety). Powodzenie komunikacji zależy od tego, czy urządzenia sieciowe uzgodnią wspólny sposób reprezentowania i przekazywania informacji. Wymaga to ustalenia wielu szczegółowych parametrów transmisyjnych. Na przykład dwie jednostki wymieniające dane w sieci przewodowej muszą stosować jednakowe poziomy napięć, identyczne mechanizmy reprezentowania danych za pomocą sygnałów elektrycznych, a także takie same formaty komunikatów.

Zdolność dwóch systemów do komunikowania się jest określana jako **zdolność do współdziałania**. Jeśli wymiana informacji między takimi jednostkami przebiega bez jakichkolwiek nieporozumień, można uznać, że poprawnie **współdziały**. Procedury uzgadniania szczegółów transmisji oraz jednakowych zasad przetwarzania danych regulują odpowiednie specyfikacje.

*Wymiana danych wymaga uprzedniego uzgodnienia przez urządzenia sieciowe parametrów transmisji — od poziomów napięć reprezentujących bity danych, po format i znaczenie przekazywanych komunikatów. Gwarancją poprawnego współdziałania systemów sieciowych jest respektowanie reguł, które opisują wszystkie aspekty komunikacji.*

Posługując się dalej terminologią dyplomatyczną, można stwierdzić, że termin **protokół komunikacyjny** (**protokół sieciowy** lub po prostu **protokół**) odnosi się do specyfikacji komunikacji sieciowej. Dany protokół może na przykład opisywać niskopoziomowe techniki transmisji bezprzewodowej albo mechanizmy wymiany komunikatów na poziomie aplikacji. Protokół opisuje procedurę postępowania w czasie przekazywania informacji. Szczególnie istotną rolę odgrywa w przypadkach występowania błędów lub zaistnienia nietypowej sytuacji. W protokole są zazwyczaj opisane właściwe zasady postępowania na wypadek niestandardowych zdarzeń (na przykład braku spodziewanej odpowiedzi z drugiej jednostki). Podsumowując:

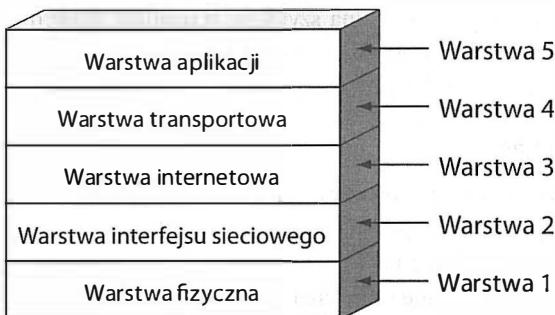
*Protokół komunikacyjny opisuje szczegółowo procedurę postępowania w danym aspekcie komunikacji między komputerami. Uwzględnia działania, które muszą zostać podjęte w przypadku wystąpienia błędów transmisyjnych lub nieprzewidzianych zdarzeń. Dany protokół może definiować niskopoziomowe parametry łącza (na przykład poziomy napięć i rodzaj sygnału) lub mechanizmy warstw wyższych (takie jak format komunikatów wymienianych przez aplikacje).*

## 1.6. Stos protokołów i modele warstwowe

Opracowując protokoły, trzeba zachować szczególną dbałość o to, by wynikowy system komunikacyjny był w pełni funkcjonalny i jednocześnie wydajny. W celu uniknięcia wiełokrotnego definiowania tych samych funkcji każdy protokół odnosi się do tego samego aspektu komunikacji, który nie jest opisany w innym protokole. Skąd zatem możemy mieć pewność, że poszczególne rozwiązania będą ze sobą poprawnie współpracować? Gwarantuje to całosciowy plan projektu. Zamiast tworzyć niezależne od siebie mechanizmy, protokoły są opracowywane grupowo i stanowią kompletnie zbioru rozwiązań, nazywane **stosami** protokołów lub **rodzinami** protokołów. Każdy protokół stosu odpowiada za realizację jednego zadania w czasie komunikacji. Działając wspólnie, nadzorują wszystkie aspekty wymiany danych, włącznie z obsługą błędów sprzętowych i innych zdarzeń. Ponadto są one zaprojektowane w taki sposób, aby ich praca w stosie była bardzo efektywna.

Podstawowym sposobem spójnego reprezentowania zbioru protokołów jest stosowanie **modelu warstwowego**. Model warstwowy ułatwia podział wszystkich zadań komunikacyjnych na współpracujące ze sobą bloki funkcjonalne. Każdy blok jest nazywany **warstwą**. Określenie to wynika z faktu ułożenia protokołów w liniowy stos. Powiązanie protokołów z warstwami ułatwia projektantom i wdrożeniom zarządzanie systemami, gdyż pozwala na skoncentrowanie się w danym czasie na wybranym aspekcie komunikacji.

Idea modelu warstwowego została zilustrowana na rysunku 1.1, na którym przedstawiono model warstwowy protokołów internetowych. Sposób ułożenia poszczególnych warstw jest przyczyną wprowadzenia określenia **stos**, które jest często stosowane do opisu oprogramowania zarządzającego protokołami w komputerze — na przykład w zdaniu „Czy stos TCP/IP działa w tym komputerze?”.



Rysunek 1.1. Model warstwowy odpowiadający protokołom internetowym (TCP/IP)

W wyjaśnieniu przyczyn takiego podziału na warstwy pomogą nam kolejne rozdziałki książek, w których opisane zostały protokoły poszczególnych warstw. Na razie konieczne jest jedynie zapoznanie się z przeznaczeniem każdej warstwy oraz zapamiętanie protokołów wykorzystywanych w komunikacji między komputerami. Funkcje przedstawionych bloków zostały opisane w kolejnych punktach tego podrozdziału. Kolejny podrozdział zawiera omówienie mechanizmów przekazywania danych przez warstwy.

## Warstwa 1 — fizyczna

Protokoły warstwy **fizycznej** definiują szczegółowe parametry medium transmisyjnego oraz związanych z nim komponentów sprzętowych. Na tym poziomie opisywane są wszystkie właściwości elektryczne, częstotliwości radiowe oraz zasady generowania sygnału.

## Warstwa 2 — interfejsu sieciowego<sup>1</sup>

Protokoły funkcjonujące w warstwie **interfejsu sieciowego** wyznaczają zasady wymiany informacji między protokołami wyższych warstw (implementowanych zazwyczaj programowo) a mechanizmami warstwy niższej (realizowanej sprzętowo). Zawarte są w nich specyfikacje adresów sieciowych i maksymalnych rozmiarów pakietów obsługiwanych przez sieć, protokołów wykorzystywanych w dostępie do medium transmisyjnego oraz adresów sprzętowych typowych dla warstwy 2.

## Warstwa 3 — internetowa

Protokoły warstwy **internetowej** stanowią podstawę funkcjonowania internetu. Mechanizmy implementowane w warstwie 3. odpowiadają za komunikację między dwoma komputerami z wykorzystaniem internetu (tj. wielu połączonych ze sobą sieci). Wyznaczają strukturę adresowania, format pakietów, metody podziału dużych pakietów internetowych na mniejsze bloki danych przeznaczone do transmisji oraz zasady zgłaszania błędów komunikacyjnych.

## Warstwa 4 — transportowa

Warstwa **transportowa** obsługuje komunikację między aplikacją zainstalowaną na jednym komputerze a programem działającym w drugiej jednostce. Mechanizmy w niej zaimplementowane wyznaczają maksymalną szybkość transmisji danych (przy której odbiornik poprawnie przetwarza nadchodzące informacje), nie dopuszczają do przeciążenia sieci, nadzorują dostarczanie danych w poprawnej kolejności.

## Warstwa 5 — aplikacji

Protokoły działające w najwyższej warstwie stosu wyznaczają zasady interakcji dwóch komunikujących się aplikacji. Specyfikacje warstwy aplikacji opisują format i znaczenie wymienianych komunikatów, a także procedury postępowania w czasie połączenia. Na poziomie warstwy 5. definiowane są mechanizmy przekazywania poczty, transferu plików, przeglądania stron internetowych, korzystania z usług telefonicznych i wideokonferencyjnych.

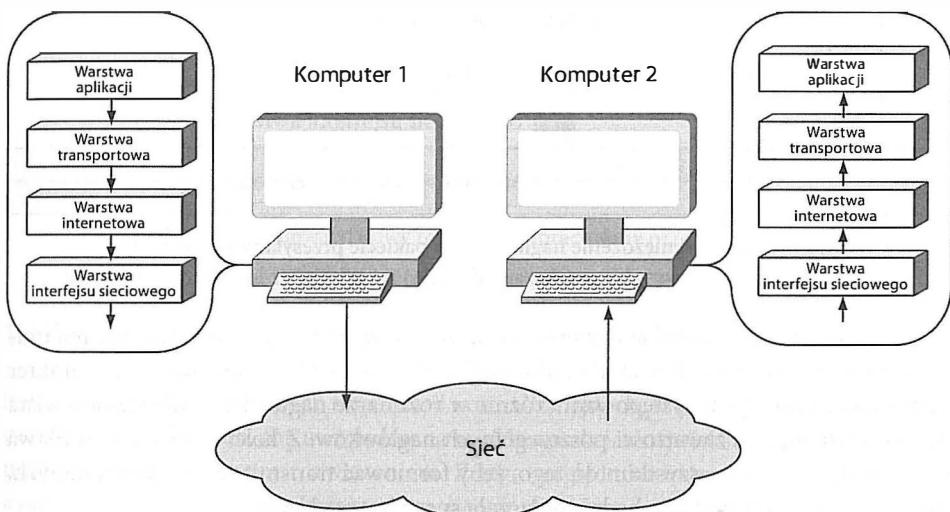
---

<sup>1</sup> W niektórych publikacjach zamiast warstwy **interfejsu sieciowego** używany jest termin **warstwy łącza danych**. Określenie to wprowadza jednak pewną niejednoznaczność w przypadku posługiwania się równocześnie innym modelem warstwowym, w którym również występuje **warstwa łącza danych** (model ten został opisany w dalszej części książki).

## 1.7. Przekazywanie danych między warstwami

Podział na warstwy nie jest jedynie abstrakcyjnym procesem, który ma pomóc w zrozumieniu działania protokołów. Przeciwnie, implementacja protokołów odwzorowuje model warstwowy, zapewniając przekazywanie danych wynikowych z jednej warstwy na wejście kolejnej. Co więcej, w celu zwiększenia wydajności przetwarzania, zamiast dostarczania kopii pakietów, oprogramowanie obsługujące protokoły sąsiednich warstw przekazuje wskaźniki na pakiety. Dzięki temu operacja przekazania danych jest realizowana bardzo szybko.

Aby przeanalizować zasadę działania protokołów, rozważmy przykład dwóch komputerów przyłączonych do jednej sieci. Na rysunku 1.2 zostały przedstawione stosy protokołów funkcjonujące w systemach obydwu jednostek. Gdy aplikacja wysyła dane, umieszcza je w pakucie. Pakiet ten następnie przechodzi przez wszystkie warstwy stosu komputera nadawczego, po czym zostaje wyemitowany w warstwie fizycznej<sup>2</sup>. Po dostarczeniu danych do jednostki docelowej pakiet jest przekazywany przez poszczególne warstwy w górę stosu. Jeśli aplikacja odbiorcza wygeneruje odpowiedź, proces się powtarza. Zmienia się jedynie kierunek przesyłania informacji. Pakiet odpowiedzi przechodzi przez wszystkie warstwy drugiej jednostki i po dostarczeniu do pierwszego komputera jest przekazywany w górę stosu protokołów urządzenia odbiorczego.



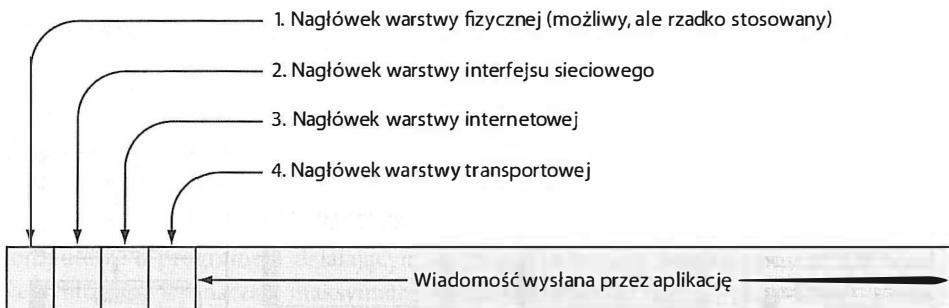
Rysunek 1.2. Przekazywanie danych przez warstwy stoso protokołów w czasie sieciowej wymiany informacji

<sup>2</sup> Rysunek przedstawia pracę pojedynczej sieci. Działanie warstw protokołów podczas przekazywania danych za pomocą urządzeń pośrednich (takich jak routery) zostanie omówione w części poświęconej architekturze internetu.

## 1.8. Nagłówki i warstwy

Każda warstwa obsługi protokołów wykonuje pewne operacje, które mają na celu zagwarantowanie odbioru danych w odpowiedniej kolejności. Aby wykonać wspomniane operacje, oprogramowanie działające w każdej z jednostek musi wymieniać dodatkowe informacje. W tym celu każda warstwa jednostki wysyłającej dane dodaje do pakietu pewne nadmiarowe informacje. Analogiczna warstwa systemu odbiorczego usuwa te informacje z pakietu i uwzględnia je w swoim działaniu.

Dodatkowy blok danych dodany przez określony protokół jest nazywany **nagłówkiem**. W zrozumieniu zasady posługiwania się nagłówkami pomocny będzie przykład pakietu przesyłanego w sieci między dwoma komputerami przedstawionym na rysunku 1.2. Nagłówki są dołączane do pakietów danych przez oprogramowanie warstw systemu nadawczego w chwili, gdy pakiety są przez nie przetwarzane. Oznacza to, że stosowny nagłówek dodaje warstwę transportową, następnie warstwa internetowa itd. Zatem przechwycony w sieci pakiet będzie zawierał nagłówki w kolejności przedstawionej na rysunku 1.3.



Rysunek 1.3. Zagnieźdżenie nagłówków w pakiecie przesyłanym przez sieć.

Początek pakietu (pierwszy bit przekazany do sieci) znajduje się z lewej strony rysunku

Choć na rysunku wszystkie nagłówki mają jednakowy rozmiar, w praktyce nie ma uniwersalnego rozmiaru nagłówka. Ponadto nagłówek warstwy transportowej ma charakter opcjonalny. Przyczyna występowania różnic w rozmiarze nagłówka stanie się oczywista po przeanalizowaniu zawartości poszczególnych nagłówków. Z kolei wiedząc, że warstwa fizyczna służy przede wszystkim do tego, żeby formować transmitowany sygnał danych, nie należy się spodziewać, że będzie dołączała specjalny nagłówek.

## 1.9. Organizacja ISO i siedmiowarstwowy model odniesienia OSI

W czasie gdy opracowywany był stos protokołów internetowych, dwie największe organizacje standaryzacyjne połączyły swoje siły, aby przygotować alternatywny model odniesienia. Jednocześnie utworzyły one zbiór protokołów odpowiedzialny za komunikację między sieciami. Wspomniane organizacje to:

- Międzynarodowa Organizacja Normalizacyjna (ISO — ang. *International Organization for Standardization*),
- Międzynarodowa Unia Telekomunikacyjna, Sektor Normalizacji Telekomunikacji (ITU-T — ang. *International Telecommunications Union — Telecommunication Standardization Sector*).<sup>3</sup>

Model warstwowy ISO jest znany jako **siedmiowarstwowy model odniesienia łączenia systemów otwartych** (OSI — ang. *Open System Interconnection*). Nazwa modelu jest często mylona z uwagi na podobieństwo akronimów OSI i ISO. Szukając informacji na jego temat, można znaleźć właściwe odnośniki zarówno po wpisaniu hasła **siedmiowarstwowy model OSI**, jak i po wprowadzeniu słów **siedmiowarstwowy model ISO**. Wszystkie warstwy modelu zostały przedstawione na rysunku 1.4.



Rysunek 1.4. Siedmiowarstwowy model OSI opracowany przez organizację ISO

## 1.10. Kulisy standaryzacji

Podobnie jak większość innych organizacji normalizacyjnych, ISO i ITU opracowują nowe standardy w taki sposób, aby uwzględniona w nich została jak największa liczba opinii stron zainteresowanych nowym rozwiązaniem. Z tego względu niektóre standardy sprawiają wrażenie kompromisów o charakterze politycznym, a nie naukowym czy inżynierskim. Siedmiowarstwowy model OSI również budzi wiele kontrowersji i rzeczywiście jest efektem politycznego kompromisu. Zarówno sam model, jak i protokoły stosu OSI zostały opracowane jako konkurencja w stosunku do protokołów internetowych.

Organizacje ISO i ITU są ogromnymi organizacjami normalizacyjnymi, które definiują rozwiązania w dziedzinie telefonii i innych ogólnoświatowych systemów komunikacji. Protokoły internetowe zostały z kolei opracowane przez niewielką grupę badaczy. Organizacje

---

<sup>3</sup> W czasie tworzenia standardów organizacja ITU miała nazwę Międzynarodowy Komitet Doradczy do spraw Telefonii i Telegrafii (CCITT — ang. *Consultative Committee for International Telephone and Telegraph*).

normalizacyjne mogły więc zyskać pewność, że narzucony przez nie zestaw protokołów będzie powszechnie stosowany, a rozwiązania proponowane przez grupę badaczy zostaną zmarginalizowane. W pewnym momencie nawet rząd Stanów Zjednoczonych został przekonany, że stos TCP/IP powinien zostać zastąpiony przez protokoły OSI.

Ostatecznie jednak okazało się, że z technicznego punktu widzenia technologia TCP/IP przewyższała rozwiązania OSI i w ciągu kilku lat prace nad opracowaniem i wdrożeniem niezależnych protokołów zostały porzucone. Organizacje standaryzacyjne przygotowały więc siedmiowarstwowy model odniesienia, w którym nie występuje warstwa internetowa. Przez wiele lat zwolennicy tego rozwiązania starali się rozszerzyć definicje modelu tak, aby objął on stos TCP/IP. Dowodzili, że warstwę trzecią można traktować jako warstwę internetową, a kilka wspomagających protokołów można umieścić w warstwie piątej i szóstej. Najśmieszniejsze jest to, że wielu inżynierów nadal określa własne aplikacje jako rozwiązania **warstwy siódmej**, wiedząc, że warstwy piąta i szósta pozostają niewykorzystane i są niepotrzebne.

## 1.11. Pozostała część książki

Niniejsza książka została podzielona na pięć części. Kolejne rozdziały pierwszej części prezentują aplikacje sieciowe i zasady programowania sieciowego. Osoby dysponujące komputerem mogą dzięki niej budować i wykorzystać programy internetowe, zapoznając się jednocześnie z dalszymi opracowaniami. Pozostałe cztery części są omówieniem zasad działania poszczególnych technologii sieciowych. W drugiej części znajduje się opis procedur wymiany informacji i mechanizmów transmisji danych. Zostało w niej wyjaśnione to, w jaki sposób energia elektryczna i elektromagnetyczna może zostać wykorzystana do przenoszenia informacji za pośrednictwem przewodów oraz w sposób bezprzewodowy.

Trzecia część dotyczy przede wszystkim technik przełączania pakietów. Wyjaśnia, dla którego sieci komputerowe bazują na transmisji pakietów. Zawiera opis formatów pakietów, procedur kodowania transmisyjnego oraz mechanizmów przekazywania pojedynczych pakietów przez sieć do jednostki docelowej. W tej części przedstawiono również podział sieci komputerowych na różne kategorie, w tym na sieci lokalne (LAN — ang. *Local Area Network*) oraz sieci rozległe (WAN — ang. *Wide Area Network*). Każda kategoria została szczegółowo scharakteryzowana, a omówieniu towarzyszą liczne przykłady technologii.

Czwarta część książki odnosi się do komunikacji międzymiędzysieciowej i związanego z nią stosu protokołów TCP/IP. Zawiera opis struktury internetu oraz samych protokołów TCP/IP. Omówione w niej zostały takie zagadnienia, jak schemat adresowania IP, odwzorowanie adresów internetowych na adresy sprzętowe, routing internetowy oraz protokoły routingu. W części tej wyjaśniono również wiele pojęć, które stanowią podstawę funkcjonowania internetu, w tym enkapsulację, fragmentację, kontrolę przeciążenia i przepływu pakietów, obwody wirtualne, translację adresów, przyłączanie do sieci, protokół IPv6, a także wiele innych protokołów pomocniczych.

W piątej części zostały zamieszczone omówienia wielu innych zagadnień, które są związane z siecią jako całością, ale nie można ich powiązać z żadnym konkretnym aspektem jej działania. Jej rozdziały dotyczą problemów wydajności sieci, wprowadzania nowych technologii, zabezpieczania sieci oraz zarządzania siecią.

## 1.12. Podsumowanie

Duża liczba technologii, produktów i sposobów łączenia sieci sprawia, że komunikacja sieciowa jest niezwykle złożonym zagadnieniem. Zrozumienie go zależy od zapoznania się z pięcioma kluczowymi pojęciami sieciowymi: aplikacjami sieciowymi i programowaniem sieciowym, wymianą danych, przełączaniem pakietów i technologiami sieciowymi, komunikacją między sieciami za pomocą protokołów TCP/IP, a także z kilkoma rozwiązaniami, które odnoszą się do większej liczby warstw modelu (na przykład bezpieczeństwem danych i zarządzaniem siecią).

Ponieważ w transmisji danych uczestniczy wiele urządzeń, konieczne jest uzgodnienie odpowiednich parametrów komunikacji, w tym cech elektrycznych sygnału (na przykład poziomu napięcia) oraz formatu i znaczenia wiadomości. Poprawne współdziałanie wymaga, aby każda jednostka działała zgodnie z ustaloną zbiorem protokołów, definiujących wszystkie elementy wymiany danych. Z kolei w celu zapewniania odpowiedniego współdziałania samych protokołów cały stos protokołów jest opracowywany w tym samym czasie. Działania projektowe koncentrują się wokół **modelu warstwowego**, który ułatwia zadanie dzięki temu, że pozwala inżynierom na zajęcie się jednym konkretnym aspektem komunikacji bez obaw o wpływ ich prac na inne mechanizmy systemu. Stosowane w internecie protokoły TCP/IP są przygotowane zgodnie z pięciowarstwowym modelem odniesienia, mimo że firmy telekomunikacyjne i Międzynarodowa Organizacja Normalizacyjna zaproponowały stosowanie modelu siedmiowarstwowego.

## ZADANIA

- 1.1. Podaj przyczyny szybkiego rozwoju internetu w ostatnich latach.
- 1.2. Wymień gałęzie przemysłu zależne od sieci komputerowych.
- 1.3. Na podstawie treści rozdziału określ, czy możliwe jest tworzenie aplikacji internetowych bez znajomości architektury internetu oraz stosowanych technologii. Uzasadnij swoją odpowiedź.
- 1.4. Jaki aspekt funkcjonowania sieci opisuje termin transmisja danych?
- 1.5. Na czym polega przełączanie pakietów i dlaczego jest ono tak istotne w internecie?
- 1.6. Przedstaw krótko historię internetu, uwzględniając początki jego rozwoju.
- 1.7. Na czym polega współdziałanie technologii i dlaczego jest ono szczególnie istotne w internecie?
- 1.8. Czym jest protokół komunikacyjny? Jakie dwa aspekty komunikacji opisuje protokół?
- 1.9. Czym jest stos protokołów? Jaka jest korzyść ze stosowania stosów protokołów?
- 1.10. Opisz model warstwowy TCP/IP. Wyjaśnij, w jaki sposób powstał.
- 1.11. Wymień warstwy modelu TCP/IP i krótko je scharakteryzuj.
- 1.12. Wyjaśnij przyczynę dodawania i usuwania nagłówków podczas przekazywania danych przez kolejne warstwy modelu.
- 1.13. Wymień najważniejsze organizacje normalizacyjne, które opracowują standardy wymiany danych i budowy sieci komputerowych.
- 1.14. Opisz krótko przeznaczenie warstw modelu OSI.

## *Zawartość rozdziału*

- 2.1. Wprowadzenie 45
- 2.2. Współdzielenie zasobów 45
- 2.3. Rozwój internetu 46
- 2.4. Od współdzielenia zasobów do komunikacji 47
- 2.5. Od tekstu do multimedialnych 49
- 2.6. Najnowsze trendy 50
- 2.7. Podsumowanie 51

# 2

## *Kierunki rozwoju internetu*

### **2.1. Wprowadzenie**

Celem niniejszego rozdziału jest przedstawienie trendów w rozwoju technologii sieciowych i internetu. Omówienie rozpoczyna się od prezentacji historii internetu z uwzględnieniem założeń leżących u podstaw tego rozwiązania oraz przejścia od scentralizowanego zarządzania zasobami do w pełni rozproszonego systemu sieciowego.

W dalszych rozdziałach książki rozważania te zostaną uzupełnione o analizę specjalistycznych aplikacji internetowych. Poza opisem samych mechanizmów komunikacji przedstawiono w nich również interfejs programistyczny, który umożliwia programom internetowym komunikowanie się.

### **2.2. Współdzielenie zasobów**

Pierwsze wdrożenia sieci komputerowych były realizowane w czasie, gdy komputery miały bardzo duże rozmiary, były bardzo drogie, a główne zastosowanie technologii sieciowej sprowadzało się do **współdzielenia zasobów**. Typowe rozwiązanie polegało na przyłączeniu do centralnego komputera wielu stacji roboczych wyposażonych jedynie w klawiaturę i monitor. W późniejszym okresie sieci umożliwiały użytkownikom współdzielenie urządzeń peryferyjnych, takich jak drukarki. Ogólnie rzecz biorąc:

*Wczesne sieci komputerowe miały na celu umożliwienie wspólnego korzystania z kosztownych scentralizowanych zasobów firmowych.*

W latach 60. **Agencja Zaawansowanych Projektów Badawczych** (ARPA<sup>4</sup> — ang. *Advanced Research Projects Agency*), stanowiąca jednostkę Departamentu Obrony Stanów Zjednoczonych, rozpoczęła intensywne prace nad współdzieleniem zasobów informatycznych. Naukowcy potrzebowali bowiem wysoko wydajnych komputerów, które były w owym czasie niezwykle kosztowne. Budżet ARPA nie pozwalał na zakup wielu tego typu jednostek. Pomyśleano więc o sieciowej wymianie danych, która wyeliminowałaby konieczność kupowania oddzielnych jednostek na potrzeby każdego projektu. Agencja planowała połączenie wszystkich komputerów siecią i zainstalowanie na nich oprogramowania, które pozwalałoby naukowcom na realizację określonych zadań w ramach systemu, który najlepiej by się do tego nadawał.

Zaproszono więc do współpracy kilku najznamienitszych naukowców i postawiono przed nimi zadanie opracowania zasad komunikacji sieciowej. Jednocześnie zatrudniono też inżynierów, których zadanie polegało na przekształceniu projektu w działający system o nazwie ARPANET. Projekt okazał się rewolucyjny. Zespół badawczy zdecydował się na zastosowanie techniki **przełączania pakietów**, stanowiącej podstawę dzisiejszego internetu. Prace zespołu były w późniejszym czasie kontynuowane jako projekt Internet.<sup>5</sup> Tak więc internet rozpoczął swoje istnienie w latach 80. jako wynik prac naukowców, a w latach 90. stał się wielkim sukcesem komercyjnym.

## 2.3. Rozwój internetu

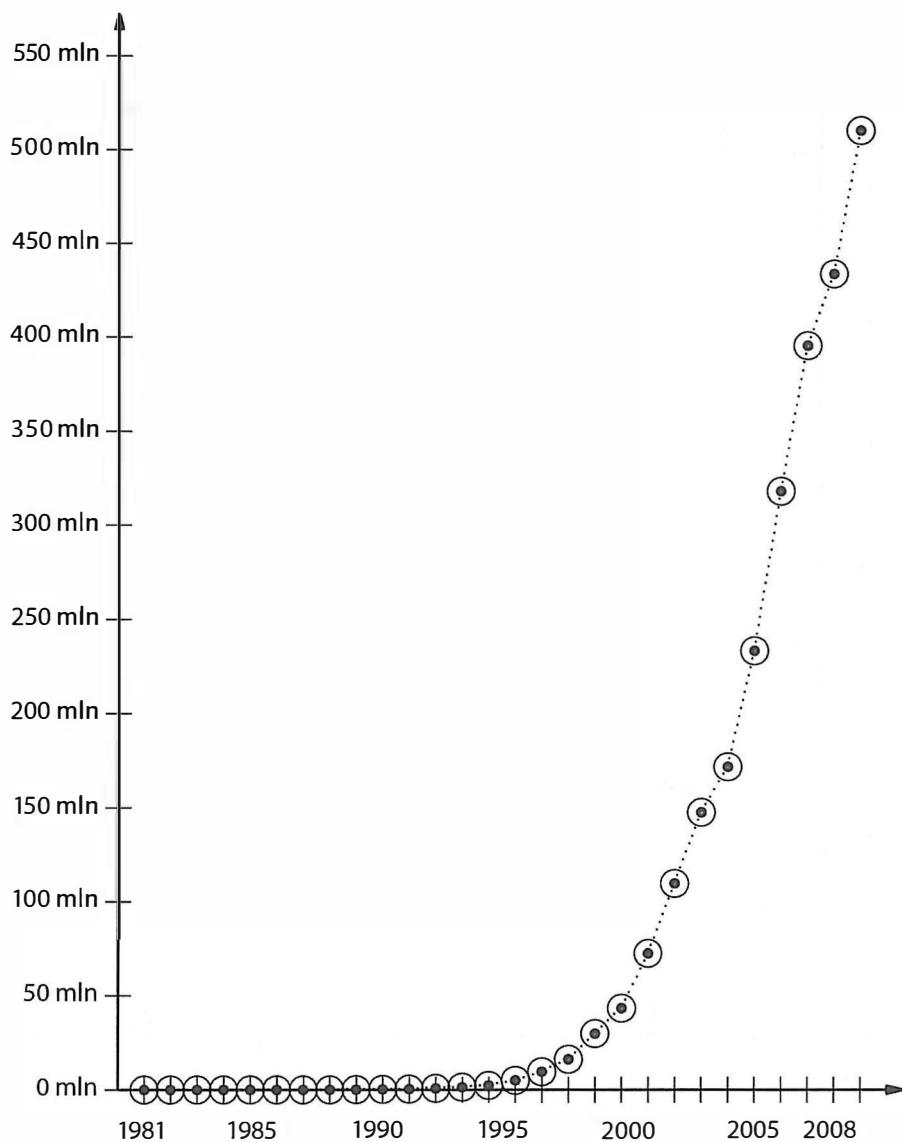
W czasie krótszym niż 30 lat internet przekształcił się z prototypowego rozwiązania obejmującego kilka lokalizacji w globalny system komunikacyjny dostępny we wszystkich krajach świata. Szybkość rozwoju okazała się imponująca. Na rysunku 2.1 przedstawiono liczbę komputerów przyłączonych do sieci internet w funkcji czasu (od roku 1981 do 2008).

Na osi Y wykresu 2.1 przedstawiono w skali liniowej wartości od zera do pięciuset pięćdziesięciu milionów. Skala liniowa nie pozwala jednak na precyzyjne określenie liczby jednostek w poszczególnych okresach, ponieważ eliminuje niewielkie wartości charakterystyczne dla początków internetu. Można więc odnieść wrażenie, że rozwój sieci był zahamowany aż do 1994 roku, a największy wzrost odnotowujemy w ostatnich latach. W rzeczywistości w roku 1998 co sekundę do sieci przyłączany był jeden nowy komputer. Trend ten się nasilał, dzięki czemu w roku 2007 w każdej sekundzie internet powiększał się o kolejne dwie stacje. Aby zaobserwować początkowy wzrost liczby stacji, warto przyjrzeć się rysunkowi 2.2, który przedstawia wykres w skali logarytmicznej.

Z analizy rysunku 2.2 wynika, że przez ponad 20 lat obserwujemy wykładniczy wzrost liczby komputerów przyłączanych do internetu. Oznacza to, że liczba stacji podwaja się co około dziewięć do czternastu miesięcy. Co ciekawe, przyrost jednostek osłabił się nieznacznie w latach 90., gdy znaczna większość obywateli państw wysoko rozwiniętych uzyskała dostęp do sieci.

<sup>4</sup> W niektórych okresach istnienia agencja wykorzystywała w nazwie słowo **Defense** (obronny), używając wówczas akronimu **DARPA**.

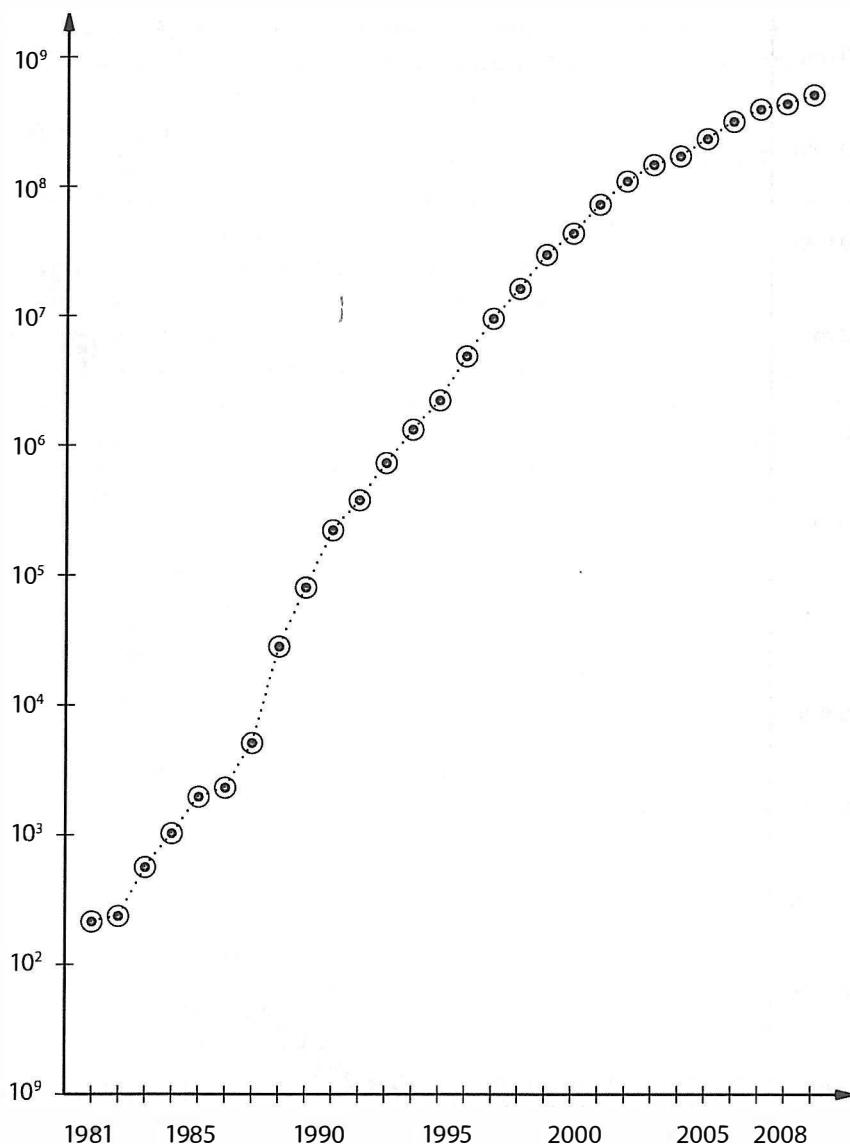
<sup>5</sup> Przełączanie pakietów zostało opisane szczegółowo w rozdziale 13.



Rysunek 2.1. Rozwój internetu — liczba komputerów w funkcji czasu

## 2.4. Od współdzielienia zasobów do komunikacji

Wraz z rozwojem internetu można było zauważać zmiany dwojakiego rodzaju. Po pierwsze, istotnie wzrosła szybkość transmisji danych — łącza szkieletowe mogą obecnie przenosić 100 000 razy więcej bitów w sekundzie niż w początkowej fazie działania sieci. Druga zmiana wydaje się oczywista — internet przestał być domeną działań naukowców i inżynierów, nie ogranicza się również do oferowania dostępu do wspólnych zasobów aplikacjom badawczym.



Rysunek 2.2. Rozwój internetu — skala logarytmiczna

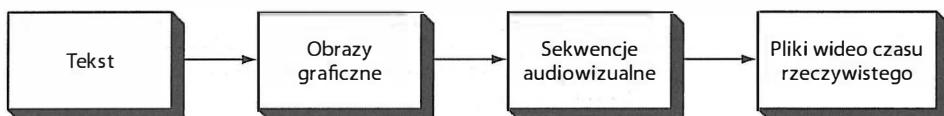
Odejście od modelu współdzielonych zasobów i włączanie nowych rodzajów aplikacji jest wynikiem dwóch zmian technologicznych. Z jednej strony, większe szybkości transmisyjne umożliwiły szybkie przekazywanie dużych zbiorów danych. Z drugiej strony, komputery osobiste stały się urządzeniami bardzo wydajnymi i niedrogimi, oferującymi swoim użytkownikom bardzo dużą moc obliczeniową i efektywne systemy graficzne. Ograniczyło to zapotrzebowanie na współdzielone zasoby infrastruktury informatycznej.

A zatem:

*Dostępność wysoko wydajnych technik obliczeniowych i komunikacyjnych spowodowała, że zastosowanie internetu zmieniło się z medium oferującego dostęp do współdzielonych zasobów w system komunikacyjny ogólnego przeznaczenia.*

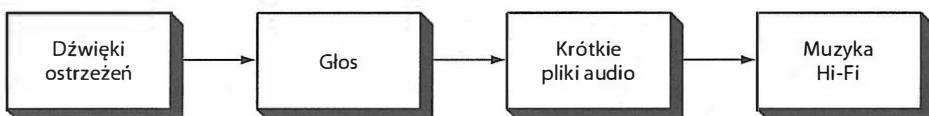
## 2.5. Od tekstu do multimedialów

Jedna z najbardziej zauważalnych zmian dotyczy rodzaju danych przesyłanych w internecie. Na rysunku 2.3 zilustrowano kolejne etapy wspomnianych zmian.



Rysunek 2.3. Zmiana rodzaju danych przesyłanych przez użytkowników w internecie

Zgodnie z rysunkiem w początkowej fazie istnienia internet bazował na danych tekstowych. Doskonałym przykładem jest w tym przypadku poczta elektroniczna, zapewniająca przesyłanie wiadomości, które były prezentowane za pomocą wstępnie ustalonej czcionki. W latach 90. komputery zostały wyposażone w kolorowe monitory, zdolne do wyświetlania grafiki. Powstały również aplikacje umożliwiające przesyłanie obrazów. Pod koniec lat 90. przesyłanie wideoklipów, a także sekwencji wizyjnych czasu rzeczywistego było już technicznie wykonalne. Analogiczne przeobrażenia zachodziły w transmisji dźwięku. Ich ilustracją jest rysunek 2.4.



Rysunek 2.4. Zmiany w sposobie przekazywania dźwięku przez internet

Połączenie tekstu, grafiki, dźwięku i sekwencji wizyjnych opisywane jest jednym terminem — **multimedia**. Dokumenty multimedialne stanowią znaczną część treści przesyłanych obecnie w internecie. W ostatnich latach poprawiła się również jakość tego typu materiałów. Większa szerokość pasma zapewnia bowiem możliwość przekazywania filmów wideo o wysokiej rozdzielczości oraz muzyki Hi-Fi. Podsumowując:

*Wykorzystanie internetu przeszło drogę od przesyłania statycznych dokumentów tekstowych do strumieniowania danych multimedialnych o wysokiej jakości.*

## 2.6. Najnowsze trendy

Mimo tak znaczącego postępu nadal opracowywane są nowe technologie i aplikacje internetowe. Większość zmian dotyczy tradycyjnych systemów telekomunikacyjnych. Doskonałymi przykładami są w tym przypadku telefonia i telewizja kablowa, które przechodząc z transmisji analogowej do cyfrowej, zaadaptowały rozwiązania internetowe. Wzrasta również wykorzystanie sieci w urządzeniach mobilnych. Lista najistotniejszych zmian została przedstawiona w tabeli 2.1.

Tabela 2.1. Przykłady zmian w komunikacji sieciowej i w internecie

Dziedzina	Zmiany
Telefonia	Przejście od systemu analogowego do technologii IP (VoIP — Voice over IP).
Telewizja kablowa	Przejście od transmisji analogowej do wykorzystania protokołu IP.
Telefonia komórkowa	Zastąpienie sieci analogowych cyfrowymi systemami 3G.
Dostęp do internetu	Zwiększenie udziału łączys bezprzewodowych (Wi-Fi).
Dostęp do danych	Zastąpienie scentralizowanych systemów rozproszonymi usługami P2P.

Jednym z najciekawszych obserwowanych zjawisk jest to, że zmieniają się aplikacje internetowe, mimo iż wykorzystywana przez nie technologia pozostaje taka sama. Potwierdzeniem tego faktu jest wykaz nowo powstałych aplikacji sieciowych, zamieszczony w tabeli 2.2.

Tabela 2.2. Przykłady aplikacji internetowych

Aplikacja	Zastosowanie
Wysokiej jakości systemy telekonferencyjne	Komunikacja biznesowa
Systemy nawigacyjne	Wojsko, spedycja, odbiorcy indywidualni
Sieci sensorowe	Ochrona środowiska, systemy bezpieczeństwa
Usługi społecznościowe	Odbiorcy indywidualni, organizacje wolontariuszy

Dostępność wysokiej jakości systemów telekonferencyjnych, takich jak *TelePresence* firmy Cisco, ma niezwykle duże znaczenie dla firm, gdyż umożliwia organizowanie spotkań bez potrzeby ponoszenia wydatków związanych z podrózami. W wielu przedsiębiorstwach ograniczenie kosztów podróży istotnie zmniejsza całkowite koszty funkcjonowania.

Usługi społecznościowe, takie jak Facebook, Second Life i YouTube, są nie mniejszą fascynującą, ponieważ doprowadziły do powstania nowych więzi społecznych — istnieją grupy

osób, które znają się jedynie dzięki internetowi. Socjologowie sądzą, że rozwiązania tego typu pozwolą większym grupom osób na poznanie innych użytkowników sieci o podobnych zainteresowaniach, a tym samym zainicjują powstanie małych grup społecznych.

## 2.7. Podsumowanie

Celem rozpoczęcia prac Agencji Zaawansowanych Projektów Badawczych było opracowanie systemu, który umożliwiałby badaczom ARPA współdzielenie zasobów obliczeniowych. W późniejszym okresie działalności agencja skupiła się na komunikacji sieciowej i zainicjowała działanie internetu — sieci rozwijającej się przez dekady w postępie wykładniczym.

Wraz z pojawieniem się wysoko wydajnych komputerów osobistych oraz sieci komputerowych o dużej przepustowości zmienił się rodzaj zainteresowania internetem. Współdzielenie zasobów informatycznych ustąpiło miejsca różnym formom komunikacji. Zamiast tekstu rozpoczęto przesyłanie grafiki, krótkich filmów wideo oraz strumieni audiowizualnych czasu rzeczywistego. Analogiczne zmiany dotyczyły przekazu dźwięku. Dzięki temu możliwe stało się przekazywanie w internecie dokumentów multimedialnych.

Technologie internetowe w różny sposób oddziałują na społeczeństwo. Ostatnie zmiany są zauważalne przede wszystkim w telefonii, telewizji kablowej i sieciach komórkowych. We wszystkich wymienionych gałęziach telekomunikacji wykorzystywane są cyfrowe technologie internetowe. Wzrosło również znaczenie bezprzewodowego dostępu do internetu oraz dostępności sieci w urządzeniach przenośnych.

Choć podstawowe rozwiązania internetowe pozostają w zasadzie niezmienne od lat, na rynku aplikacji wciąż pojawiają się nowe produkty, które powiększają zbiór zastosowań systemu. Przedsiębiorstwa korzystają z systemów telekonferencyjnych, aby wyeliminować koszty podróży służbowych. Sieci sensorowe, mapy i systemy nawigacji umożliwiają ochronę środowiska i podnoszenie poziomu bezpieczeństwa obywateli. Ułatwiają również podróżowanie. Z kolei usługi społecznościowe odpowiadają za powstawanie nowych grup społecznych i organizacji.

## ZADANIA

- 2.1. Dlaczego w latach 60. tak istotne było współdzielenie zasobów informatycznych?
- 2.2. Wykres zamieszczony na rysunku 2.1 sugeruje, że rozwój internetu nastąpił dopiero po roku 1995. Jaka jest tego przyczyna?
- 2.3. Założmy, że każdego roku do sieci przyłączanych jest sto milionów nowych komputerów. Ile czasu upływa między przyłączeniami dwóch kolejnych jednostek, przy założeniu, że operacje te są równomiernie rozłożone w czasie?
- 2.4. Posługując się wykresem 2.2, oszacuj, ile nowych komputerów zostanie przyłączonych do internetu w roku 2020.
- 2.5. Jakie zmiany w wykorzystaniu internetu nastąpiły wraz z udostępnieniem usług WWW?
- 2.6. Wymień kolejne etapy zmian w przekazywaniu grafiki przez internet.

- 2.7. Opisz zmiany w transmisji dźwięku w interenie.
- 2.8. Jaki wpływ na telewizję kablową ma technologia internetowa?
- 2.9. Jaka technologia internetowa jest wykorzystywana w telefonii?
- 2.10. Dlaczego istotne jest zwiększenie udziału systemów bezprzewodowego dostępu do internetu?
- 2.11. Wymień nowe aplikacje internetowe oraz grupy ich odbiorców.
- 2.12. Wymień aplikacje internetowe, z których korzystasz na co dzień, a które nie były dostępne dla Twoich rodziców, gdy byli w tym samym wieku.

{}



# Zawartość rozdziału

3.1. Wprowadzenie	55
3.2. Dwa podstawowe pojęcia związane z internetem	56
3.3. Komunikacja połączeniowa	57
3.4. Model klient-serwer	58
3.5. Cechy aplikacji klienckich i serwerowych	59
3.6. Programy serwerowe oraz komputery pełniące rolę serwerów	59
3.7. Żądania, odpowiedzi i kierunek przepływu danych	60
3.8. Wiele aplikacji klienckich i serwerowych	60
3.9. Identyfikacja serwerów i demultiplexacja	61
3.10. Praca współbieżna serwerów	62
3.11. Pętla zależności między serwerami	63
3.12. Odwołania peer-to-peer	63
3.13. Programowanie sieciowe i interfejs gniazd	64
3.14. Gniazda, deskryptory i sieciowe operacje wejścia-wyjścia	64
3.15. Parametry i interfejs gniazd	65
3.16. Odwołania do gniazd w aplikacjach klienckich i serwerowych	66
3.17. Funkcje gniazda wykorzystywane po stronie klienta i serwera	66
3.18. Funkcja połączenia wykorzystywana jedynie po stronie klienta	68
3.19. Funkcje gniazda wykorzystywane jedynie po stronie serwera	69
3.20. Funkcje gniazda wykorzystywane w transmisji komunikatów	71
3.21. Inne funkcje gniazd	73
3.22. Gniazda, wątki i dziedziczenie	73
3.23. Podsumowanie	74

# 3

## *Aplikacje internetowe i programowanie sieciowe*

### **3.1. Wprowadzenie**

Internet oferuje swoim użytkownikom wiele różnych usług, w tym przeglądanie stron WWW, przesyłanie poczty elektronicznej, wymianę komunikatów tekstowych oraz ustanawianie wideokonferencji. Żadna z tych usług nie jest jednak elementem infrastruktury komunikacyjnej stanowiącej podstawę funkcjonowania systemu. Internet zapewnia mechanizmy ogólnego przeznaczenia wykorzystywane przez wszystkie usługi sieciowe, które działają dzięki programom uruchomionym w komputerach przyłączonych do internetu. Dzięki temu można opracowywać nowe usługi bez konieczności modyfikowania zasad działania sieci.

W tym rozdziale zostały opisane dwa rozwiązania kluczowe dla działania aplikacji internetowych. Pierwsze to tok postępowania programu w czasie komunikowania się z inną aplikacją za pośrednictwem internetu. Drugi mechanizm to interfejs programistyczny gniazda, wykorzystywany w większości aplikacji sieciowych.

Z treści rozdziału wynika, że nie trzeba znać szczegółowo zasad transmisji danych lub protokołów sieciowych, aby można było tworzyć użyteczne aplikacje. Każdy, kto opanuje kilka podstawowych technik programistycznych, może projektować programy komunikujące się ze sobą za pośrednictwem internetu. Omówienie tego zagadnienia znajduje się również w kolejnym rozdziale, w którym przedstawiono szczegóły działania niektórych aplikacji sieciowych, takich jak poczta elektroniczna.

Choć rozpoczęcie prac nie nastręcza trudności i jest możliwe utworzenie aplikacji internetowej bez zrozumienia zasad funkcjonowania sieci, zapoznanie się z protokołami i technologiami sieciowymi pozwala na tworzenie bardziej wydajnego i niezawodnego kodu,

który będzie się właściwie skalował w zależności od zastosowań. Niezbędne do tego informacje, w tym opis protokołów i mechanizmów przekazywania danych, zostały zamieszczone w dalszej części tego rozdziału.

### 3.2. Dwa podstawowe pojęcia związane z internetem

W komunikacji internetowej wyróżnia się dwa podstawowe formaty danych: **strumień** i **komunikat**. Różnice między nimi zostały przedstawione w tabeli 3.1.

Tabela 3.1. Dwa formaty danych wykorzystywane w aplikacjach internetowych

Strumień	Komunikat
Charakter połączeniowy	Charakter bezpołączeniowy
Komunikacja jeden-do-jednego	Komunikacja jeden-do-wielu
Sekwencja pojedynczych bajtów	Sekwencja pojedynczych komunikatów
Dowolna długość przekazu	Ograniczenie każdego komunikatu do 64 bajtów
Wykorzystanie w większości aplikacji	Wykorzystanie w aplikacjach multimedialnych
Zastosowanie protokołu TCP	Zastosowanie protokołu UDP

#### 3.2.1. Transport strumieni

Termin **strumień** odnosi się do mechanizmu, w którym jedna aplikacja dostarcza do innej aplikacji zbiór bajtów o określonej kolejności. W praktyce przesyłane są dwa strumienie danych między komunikującymi się jednostkami — po jednym w każdym kierunku. Na przykład, gdy przeglądarka wykorzystuje usługę strumieniowania do komunikacji z serwery WWW, wysyła żądanego dostarczenia strony do serwera, a serwer odpowiada, przekazując do komputera odpowiednią stronę. Sieć umożliwia wprowadzenie danych z dowolnej aplikacji i zapewnia dostarczenie ich do drugiego programu.

Mechanizm strumieniowania przenosi sekwencje bajtowe bez nadawania im jakiegokolwiek znaczenia i bez wyznaczania granic zakresów. Aplikacja nadawcza może w danej chwili wygenerować pojedynczy bajt lub blok bajtów. Sieć z kolei dostosowuje we własnym zakresie liczbę bajtów przekazywanych w określonej chwili. Oznacza to, że zdarza się połączenie mniejszych bloków bajtów w jeden większy zbiór, a także podział dużych bloków na mniejsze fragmenty. Ogólne założenie jest następujące:

*Choć wszystkie bajty są dostarczane we właściwej kolejności, mechanizm strumieniowania nie gwarantuje, że porcje danych rejestrowane przez aplikację odbiorczą odpowiadają zbitkom generowanym przez program nadawczy.*

### 3.2.2. Transport komunikatów

Alternatywny mechanizm komunikacji internetowej bazuje na transporcie **komunikatów** odbieranych przez sieć i dostarczanych do odległej stacji. Każdy komunikat dostarczony do odbiorcy odpowiada dokładnie komunikatowi wygenerowanemu po stronie nadawcy — sieć nigdy nie przekazuje częściowych komunikatów ani nie łączy kilku bloków w większy zbiór. Zatem jeśli nadawca formuje komunikat o  $n$  bajtach, odbiorca otrzyma dokładnie  $n$  bajtów w nadchodzącym komunikacie.

Dostarczanie komunikatów jest realizowane na zasadzie emisji pojedynczej (ang. *unicast*), multiemisji (ang. *multicast*) lub w sposób rozgłoszeniowy (ang. *broadcast*). Oznacza to, że informacja wygenerowana w jednym komputerze jest przekazywana tylko do jednego komputera docelowego, do grupy komputerów odbiorczych lub do wszystkich jednostek w danej sieci. Ponadto aplikacje działające w wielu jednostkach mogą przesyłać komunikaty do jednej wybranej aplikacji. Zatem transport komunikatów pozwala na komunikację typu jeden-do-jednego, jeden-do-wielu oraz wiele-do-jednego.

Usługa dostarczania komunikatów nie gwarantuje poprawnej kolejności odbioru tychże komunikatów, a nawet dostarczenia ich do jednostki zdalnej. Dopuszczalne są więc:

- utrata komunikatów (tj. komunikaty nie są dostarczane),
- duplikowanie komunikatów (dostarczana jest więcej niż jedna kopia komunikatu),
- dostarczanie w przypadkowej kolejności.

Programista korzystający z mechanizmu przekazywania komunikatów musi zapewnić, że aplikacja będzie działała poprawnie nawet wówczas, gdy pakiety zostaną utracone lub będą odbierane w niewłaściwej kolejności<sup>6</sup>. Ponieważ większość aplikacji wymaga gwarancji dostarczenia danych, programiści zazwyczaj korzystają z usług strumieniowych. Wyjątkami są na przykład operacje transmisji wideo, w których niezbędna jest multiemisja, a programy obsługujące przekaz są przystosowane do reagowania na niewłaściwą kolejność pakietów lub ich utratę. Z tego względu dalsze rozważania dotyczą mechanizmu strumieniowania.

## 3.3. Komunikacja połączeniowa

Usługa strumieniowania danych internetowych ma charakter **połączeniowy** (ang. *connection-oriented*), co oznacza, że jej działanie jest zbliżone do realizacji połączenia telefonicznego. Zanim dwie aplikacje wymienią między sobą informacje, muszą ustawić **połączenie**. Z chwilą nawiązania połączenia dane mogą być wymieniane w obydwu kierunkach. Po zakończeniu przekazywania informacji połączenie musi zostać rozłączone. Poszczególne operacje zostały przedstawione w algorytmie 3.1.

---

<sup>6</sup> Przyczyny powstawania wspomnianych błędów zostały opisane w kolejnych rozdziałach.

### Algorytm 3.1. Komunikacja za pomocą mechanizmu połączeniowego

Cel:

Komunikacja za pomocą mechanizmu połączeniowego

Realizacja:

Dwie aplikacje żądają ustanowienia połączenia.

Aplikacje wykorzystują połączenie do wymiany danych.

Aplikacje żądają zakończenia połączenia.

}

## 3.4. Model klient-serwer

Analizując pierwszy etap algorytmu 3.1, można sobie zadać pytanie, w jaki sposób dwie działające w różnych systemach aplikacje koordynują swoje działania tak, aby w tym samym czasie zażądać ustanowienia połączenia. Odpowiedzią jest zastosowanie pewnej formy interakcji, którą opisuje model *klient-serwer*. Jedna z aplikacji, pełniąca rolę *serwera*, jest uruchamiana jako pierwsza i otrzymuje zadanie oczekiwania na kontakt. Drugi program, określany mianem *klienta*, rozpoczyna swoje działanie jako drugi i inicjuje połączenie. Kolejne etapy postępowania są widoczne w tabeli 3.2.

Tabela 3.2. Cechy modelu klient-serwer

Aplikacja serwerowa	Aplikacja kliencka
Uruchamiana jako pierwsza	Uruchamiana jako druga
Nie wymaga wstępnej informacji na temat klienta ustanawiającego połączenie	Musi dysponować informacją o serwerze, z którym ustanowi połączenie
Oczekuje pasywnie przez dowolnie długi czas na kontakt ze strony klienta	Inicjuje połączenie za każdym razem, gdy jest to niezbędne
Komunikuje się z klientem przez wysyłanie i odbieranie danych	Komunikuje się z serwerem przez wysyłanie i odbieranie danych
Pozostaje uruchomiona po obsłużeniu żądania klienckiego, oczekując na kolejne	Może zakończyć pracę po wymianie informacji z serwerem

W kolejnych punktach rozdziału opisane zostały sposoby wykorzystania modelu klient-serwer w działaniu różnych usług. Na tym etapie trzeba jednak zapamiętać, że:

*Zapewniając podstawowe mechanizmy komunikacyjne, internet nie inicjuje połączeń ani nie akceptuje połączeń pochodzących ze zdalnych komputerów. Wszelkie usługi są realizowane przez programy funkcjonujące jako aplikacje klienckie i serwerowe.*

### 3.5. Cechy aplikacji klienckich i serwerowych

Mimo że istnieją pewne nieznaczne odstępstwa, większość implementacji modelu klient-serwer działa w jednakowy sposób. Oto kilka ogólnych cech programu klienckiego:

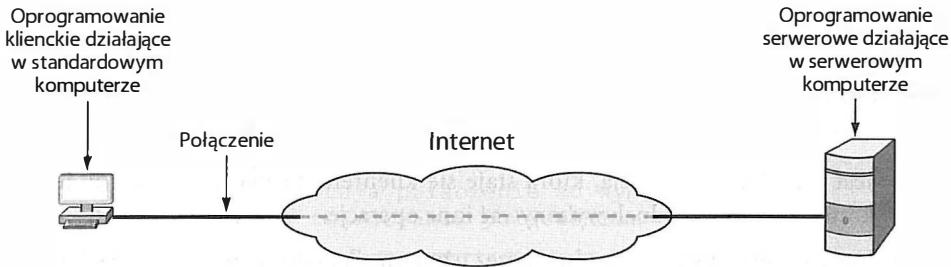
- Jest niezależną aplikacją, która staje się klientem na pewien czas, gdy jest to konieczne. Może jednak wykonywać inne operacje.
- Jest wywoływany bezpośrednio przez użytkownika i działa tylko w ramach jednej sesji.
- Działa lokalnie — w systemie operacyjnym komputera danego użytkownika.
- Inicjuje komunikację z serwerem.
- W razie potrzeby może korzystać z wielu usług. Niemniej zazwyczaj odwołuje się do jednego serwera w danym czasie.
- Nie wymaga szczególnie wydajnych komponentów sprzętowych.

Do cech oprogramowania serwerowego należy natomiast zaliczyć:

- Jest specjalnie przygotowanym programem udostępniającym jedną usługę, która może obsługiwać jednocześnie wiele żądań klienckich.
- Jest uruchamiane automatycznie wraz ze startem systemu, a jego działanie obejmuje wiele sesji komunikacyjnych.
- Działa w ramach wydajnego środowiska sprzętowego.
- Oczekuje pasywnie na kontakt ze strony jednostek klienckich.
- Akceptuje połączenia z poszczególnych jednostek klienckich, zapewniając im dostęp do pojedynczej usługi.
- Wymaga dostępności bardzo wydajnych komponentów sprzętowych oraz skomplikowanego systemu operacyjnego.

### 3.6. Programy serwerowe oraz komputery pełniące rolę serwerów

Termin *serwer* często okazuje się niejednoznaczny. Teoretycznie odnosi się on do programu, który oczekuje na połączenie, a nie komputera, w którym to oprogramowanie działa. Nienajlej w przypadku, gdy dany komputer jest przeznaczony do utrzymywania jednego programu serwerowego lub większej liczby takich programów, sam bywa określany mianem *serwera*. Zamieszanie niekiedy potęgują dostawcy sprzętu komputerowego, którzy klasyfikują jako serwery jednostki o wydajnych procesorach, pojemnych pamięciach i niestandardowych systemach operacyjnych. Podział ten jest widoczny na rysunku 3.1.



Rysunek 3.1. Połączenie klient-serwer

### 3.7. Żądania, odpowiedzi i kierunek przepływu danych

Określenia **klient** i **serwer** zależą od tego, która strona inicjuje połączenie. Po ustanowieniu połączenia możliwa jest dwukierunkowa komunikacja (tj. dane przypływają od klienta do serwera oraz od serwera do klienta). Zazwyczaj klient wysyła żądania do serwera, a serwer reaguje, odsyłając odpowiedź. W niektórych przypadkach jednostka kliencka generuje całą serię żądań, spodziewając się serii odpowiedzi ze strony serwera (na przykład klient bazy danych może pozwolić użytkownikowi na pobranie więcej niż jednej informacji w danej chwili). Ideę tę można podsumować następująco:

*W modelu klient-serwer informacje mogą być przekazywane w dowolnym kierunku lub w obydwu kierunkach jednocześnie. Choć większość usług działa w ten sposób, że oczekuje od klienta wygenerowania żądania (lub kilku żądań), a od serwera odesłania odpowiedzi, możliwe są również inne rodzaje interakcji.*

### 3.8. Wiele aplikacji klienckich i serwerowych

Klient i serwer są programami komputerowymi. Komputer z kolei umożliwia uruchamianie wielu aplikacji w tym samym czasie. W rezultacie komputer może wykonywać:

- pojedynczy program kliencki;
- pojedynczy program serwerowy;
- wiele kopii programu klienckiego komunikującego się z danym serwerem;
- wiele programów klienckich przeznaczonych do komunikacji z różnymi serwerami;
- wiele programów serwerowych odpowiadających właściwym aplikacjom klienckim.

Umożliwienie komputerom uruchamiania wielu aplikacji klienckich okazuje się bardzo użyteczne, ponieważ pozwala na jednoczesne odwoływanie się do większej liczby usług. Na przykład użytkownik systemu może otworzyć trzy okna i pracować równolegle z trzema programami — odbierać i przeglądać pocztę elektroniczną, korzystać z czatu i pobierać strony internetowe. Każda aplikacja pełni funkcję klienta, który kontaktuje się z określonym

serwerem. Oczywiście, możliwe jest również posługiwanie się dwoma kopiami pojedynczej aplikacji, z których każda odwołuje się do oddzielnego serwera (na przykład dwie kopie przeglądarki internetowej).

Niezwykle użyteczna jest również możliwość uruchamiania większej liczby programów serwerowych wykorzystujących wspólne zasoby sprzętowe. Pojedynczy komputer wymaga mniejszego nakładu pracy związanej z administrowaniem systemem. Najważniejsze jest jednak to, że zapotrzebowanie na usługi serwerowe jest często sporadyczne, przez co pojedynczy program serwerowy pozostaje nieobciążony przez większą część czasu. Serwer oczekujący na żądania nie zużywa zasobów procesora, a jeśli zapotrzebowanie na usługi nie jest zbyt duże, konsolidacja oprogramowania serwerowego pozwala na znaczne obniżenie kosztów sprzętu bez pogorszenia wydajności rozwiązania. Podsumowując:

*Pojedynczy wydajny komputer może udostępniać w tym samym czasie wiele usług.  
Każda z tych usług jest jednak obsługiwana przez oddzielną program serwerowy.*

### 3.9. Identyfikacja serwerów i demultipleksacja

W jaki sposób klient wskazuje serwer? Protokoły internetowe bazują na dwóch formach identyfikacji, posługując się:

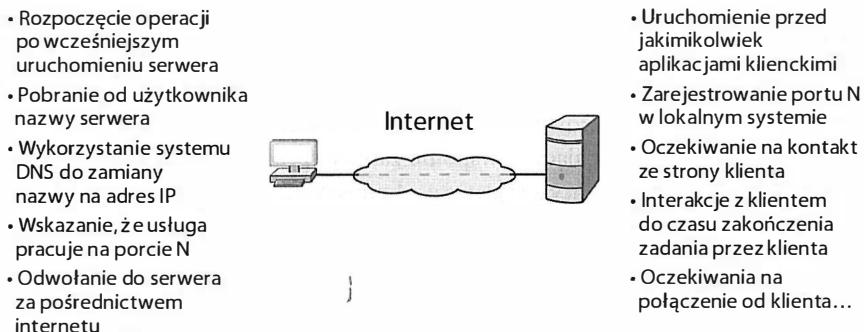
- identyfikatorem komputera, w którym pracuje aplikacja serwerowa;
- identyfikatorem usługi działającej w systemie komputera.

**Wskazanie komputera.** Każdy komputer przyłączony do internetu otrzymuje unikalny 32-bitowy identyfikator, nazywany adresem IP (adresem protokołu internetowego)<sup>7</sup>. Podczas odwoływania się do serwera klient musi określić adres IP tegoż serwera. Aby ułatwić ludziom rozróżnianie jednostek serwerowych, każdemu komputerowi przypisuje się również jednoznaczną nazwę, a opisany w rozdziale 4. system nazw domenowych przekształca wspomnianą nazwę na odpowiadający mu adres IP. Dzięki temu użytkownik może wprowadzić nazwę taką jak [www.cisco.com](http://www.cisco.com) zamiast adresu IP tego serwisu.

**Wskazanie usługi.** Każdej funkcjonującej w internecie usłudze przypisana jest niepowtarzalna 16-bitowa wartość liczbową, nazywana **numerem portu protokołu** (albo krócej **numerem portu**). Na przykład poczcie elektronicznej przypisano numer 25, a usłudze WWW port 80. Gdy serwer rozpoczyna swoją pracę, rejestruje w lokalnym systemie numer portu oferowanej przez siebie usługi. Klient, odwołując się do zdalnego serwera, uwzględnia numer portu usługi w treści przesyłanego żądania. W chwili odebrania żądania po stronie serwera, oprogramowanie serwerowe na podstawie numeru portu wyznacza aplikację, która będzie odpowiedzialna za przetworzenie żądania.

Ilustracją opisanego mechanizmu jest rysunek 3.2, na którym wymieniono wszystkie etapy komunikacji między klientem i serwerem.

<sup>7</sup> Szczegółowe informacje na temat adresów IP zamieszczone w rozdziale 21.



Rysunek 3.2. Zadania realizowane w czasie komunikacji między klientem a serwerem

### 3.10. Praca współbieżna serwerów

Z analizy rysunku 3.2 można wywnioskować, że określony serwer obsługuje jednego klienta w danym czasie. **Szeregowo** przetwarzanie żądań rzeczywiście jest stosowane w pewnych przypadkach. Jednak większość serwerów realizuje żądania **współbieżnie** — serwer uruchamia więcej niż jeden **wątek**<sup>8</sup> w celu obsługi większej liczby klientów w tym samym czasie.

Aby uświadomić sobie, jak ważna jest współbieżna obsługa żądań, rozważmy, co by się stało, gdyby klient pobierał z serwera film. Gdyby serwer obsługiwał jedno żądanie w danym czasie, wszystkie pozostałe programy klienckie musiałyby oczekiwać na zakończenie wcześniejszej transmisji. Serwery o działaniu współbieżnym nie wymagają od klienta oczekiwania. Dlatego gdyby program zdalny dostarczył żądanie przesłania krótkiego pliku (na przykład utworu muzycznego), drugi transfer rozpoczęłyby się bezzwłocznie i mógłby się zakończyć przed zakończeniem przesyłania filmu.

Szczegółowe rozwiązanie współbieżnego przetwarzania żądań zależy od budowy systemu operacyjnego. Niemniej zasadnicza idea pozostaje taka sama — kod serwera jest dzielony na dwie części: program główny oraz podprogram obsługi żądania. Wątek główny nasłuchiwa nadchodzących połączeń klienckich i powołuje wątki przeznaczone do obsługi tych połączeń. Każdy wątek obsługi wymienia dane z pojedynczym klientem i realizuje odpowiedni kod przetwarzania żądania. Po zakończeniu zadania wątek ten kończy swoją pracę. W tym samym czasie wątek główny podtrzymuje działanie serwera — po utworzeniu wątku obsługi żądania przechodzi do trybu oczekiwania na kolejne odwołania klienckie.

Jeśli więc w danej chwili realizowanych jest  $N$  żądań, w systemie działa  $N+1$  wątków. Wątek główny oczekuje na kolejne żądania, a  $N$  wątków wykonuje zadania zlecone przez poszczególne programy klienckie.

*Serwery współbieżne wykorzystują wątki do obsługi wielu żądań klienckich w tym samym czasie.*

<sup>8</sup> W niektórych systemach operacyjnych wątki tego typu są realizowane jako **wątki wykonawcze** lub **procesy**.

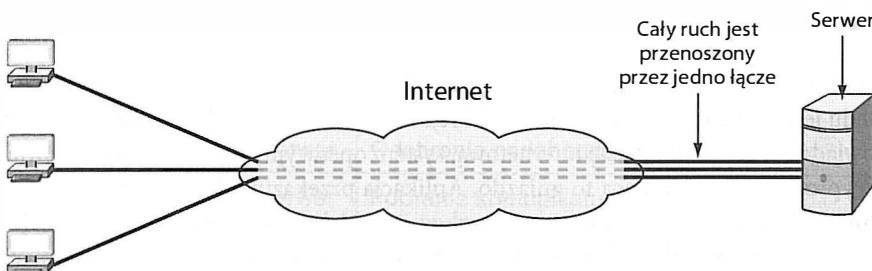
### 3.11. Pętla zależności między serwerami

Zgodnie z definicją każdy program odwołujący się do innej aplikacji pełni funkcję klienta, a każdy program odbierający żądania z innej aplikacji pracuje jako serwer. W praktyce jednak podział ten często się zaciera, ponieważ serwer jednej usługi może działać jak klient innej. Na przykład podczas wypełniania formularza na stronie internetowej serwer WWW może stać się klientem systemu bazodanowego. Ponadto może również pełnić funkcję klienta usługi zabezpieczenia systemu (na przykład w celu sprawdzenia, czy użytkownik ma prawo dostępu do serwisu).

Oczywiście, programiści muszą zachować odpowiednią staranność, projektując aplikację, aby nie dopuścić do powstania pętli zależności. Przeanalizujmy na przykład, co by się stało, gdyby usługa  $X_1$  pełniła rolę klienta usługi  $X_2$ , która z kolei byłaby klientem usługi  $X_3$ , a ta odwoływałaby się do usługi  $X_1$ . Łącuch żądań byłby generowany nieprzerwanie aż do momentu wyczerpania zasobów każdego z serwerów. Ryzyko wystąpienia takiej sytuacji jest szczególnie duże, gdy poszczególne usługi są opracowywane niezależnie. Nie ma wówczas osoby, która nadzorowałaby pracę nad kodem wszystkich serwerów.

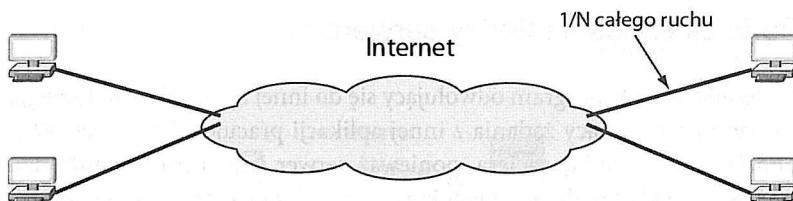
### 3.12. Odwołania peer-to-peer

Jeśli usługa jest udostępniana przez pojedynczy serwer, połączenie między tym serwerem a internetem może się okazać niewystarczające do efektywnego realizowania zadań. Rozwiązanie tego typu zostało przedstawione na rysunku 3.3.



Rysunek 3.3. Zator sieciowy w rozwiązaniu bazującym na pojedynczym serwerze

Nasuwa się więc pytanie, czy usługi internetowe mogą być świadczone bez ryzyka zatoru? Jeden ze sposobów uniknięcia „wąskich gardel” stanowi podstawę działania aplikacji współdzielenia plików. Model wspomnianego rozwiązania nazywany jest architekturą *peer-to-peer (p2p)*. Zakłada się w nim brak jednego centralnego serwera. Dane są natomiast rozmieszczone równomiernie na  $N$  serwerach. Każde żądanie klienckie jest natomiast dostarczane do właściwego serwera. Ponieważ dany serwer udostępnia jedynie  $1/N$  zbioru danych, natężenie ruchu między serwerem a internetem wynosi  $1/N$  wartości właściwej dla architektury uwzględniającej pojedynczy serwer. Dzięki temu oprogramowanie serwowe może działać na tych samych komputerach, w których pracują programy klienckie. Ilustracją opisanej koncepcji jest rysunek 3.4.



Rysunek 3.4. Interakcje w systemie peer-to-peer

### 3.13. Programowanie sieciowe i interfejs gniazd

Interfejs służący aplikacji do komunikacji jest nazywany **interfejsem programistycznym aplikacji** (API — ang. *Application Programming Interface*)<sup>9</sup>. Choć konkretna implementacja mechanizmów API zależy od rodzaju systemu operacyjnego, programiści operują pewnym „de facto standardem” interfejsu API, który odpowiada za komunikację w internecie. Jest on znany jako interfejs **API gniazd**, często również określany po prostu jako **interfejs gniazd**. Rozwiążanie to jest dostępne w wielu systemach operacyjnych, w tym w systemie Windows firmy Microsoft, a także w wielu odmianach systemu UNIX, włącznie z Linuksem. Istotne jest to, że:

*Interfejs API gniazd jest de facto standardem w komunikacji internetowej.*

### 3.14. Gniazda, deskryptory i sieciowe operacje wejścia-wyjścia

Interfejs API gniazd został opracowany jako element systemu operacyjnego UNIX. Z tego względu jest powiązany z komponentami wejścia-wyjścia. Gdy aplikacja tworzy **gniazdo** odpowiadające za komunikację internetową, system operacyjny zwraca liczbę całkowitą — **deskryptor** — identyfikującą to gniazdo. Aplikacja przekazuje deskryptor gniazda jako parametr funkcji za każdym razem, gdy wykonuje jakiekolwiek operacje na gnieździe (na przykład podczas wysyłania danych do sieci lub odbierania nadchodzących informacji).

W wielu systemach operacyjnych deskryptory gniazd są powiązane z innymi deskryptorami modułów wejścia-wyjścia. W rezultacie aplikacja może posługiwać się funkcjami `read` i `write` w celu wykonania operacji wejścia-wyjścia w odniesieniu do gniazda lub do pliku. Podsumowując:

*Gdy aplikacja tworzy gniazdo, system operacyjny zwraca jego deskryptor (liczbę całkowitą o niewielkiej wartości). Deskryptor ten identyfikuje gniazdo.*

---

<sup>9</sup> Uproszczony interfejs API (z siedmioma funkcjami) oraz przykładowy kod wykorzystania API do tworzenia aplikacji internetowej (w tym działającego serwera WWW) zostały zamieszczone w dodatku A.

### 3.15. Parametry i interfejs gniazd

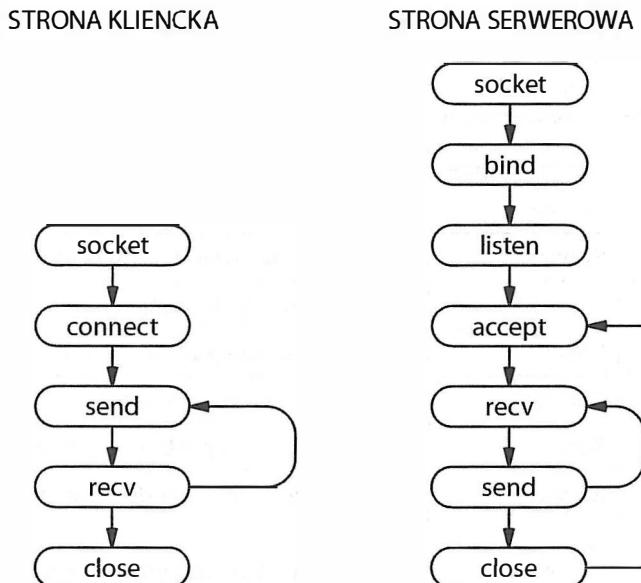
Programowanie gniazd różni się od implementacji tradycyjnych operacji wejścia-wyjścia, ponieważ aplikacja musi określić wiele dodatkowych parametrów, na przykład adres komputera zdalnego, numer portu protokołu, oraz to, czy dany program pełni funkcję klienta, czy serwera (tj. czy inicjuje połączenie, czy nie). Aby uniknąć posługiwania się pojedynczymi funkcjami z dużą liczbą parametrów, projektanci interfejsu API gniazd zdecydowali o zdefiniowaniu wielu funkcji. Zaletą takiego podejścia jest to, że większość funkcji wymaga przekazania trzech lub mniej parametrów. Wadą jest z kolei konieczność wywołania kilku funkcji przed skorzystaniem z gniazda. Najważniejsze funkcje gniazd zestawiono w tabeli 3.3.

Tabela 3.3. Zestawienie najważniejszych funkcji interfejsu API gniazd

Nazwa	Zastosowanie	Przeznaczenie
accept	Serwer	Akceptacja nadchodzącego połączenia
bind	Serwer	Określenie adresu IP i portu protokołu
close	Klient i serwer	Zakończenie komunikacji
connect	Klient	Ustanowienie połączenia ze zdalną aplikacją
getpeername	Serwer	Pozyskanie adresu IP klienta
getsockopt	Serwer	Pobranie informacji o bieżącym gnieździe
listen	Serwer	Przygotowanie gniazda do wykorzystania przez serwer
recv	Klient i serwer	Odebranie nadchodzących danych lub komunikatu
recvmsg	Klient i serwer	Odebranie nadchodzącego komunikatu
recvfrom	Klient i serwer	Pobranie komunikatu i adresu nadawcy
send (write)	Klient i serwer	Wysłanie danych lub komunikatu
sendmsg	Klient i serwer	Wysłanie komunikatu
sendto	Klient i serwer	Wysłanie komunikatu (wariant sendmsg)
setsockopt	Klient i serwer	Zmiana parametrów gniazda
shutdown	Klient i serwer	Zakończenie połączenia
socket	Klient i serwer	Utworzenie gniazda w celu realizacji powyższych funkcji

### 3.16. Odwołania do gniazd w aplikacjach klienckich i serwerowych

Na rysunku 3.5 przedstawiono sekwencję odwołań realizowanych przez typowe aplikacje kliencką i serwerową wykorzystujące połączenie strumieniowe. Jako pierwszy dane wysyła klient. Serwer z kolei oczekuje na dostarczenie informacji. W praktyce niektóre systemy działają w taki sposób, że serwer jako pierwszy wysyła dane (tj. funkcje send i recv są wywoływanie w odwrotnej kolejności).



Rysunek 3.5. Sekwencja wywołań funkcji gniazd w aplikacjach klienckiej i serwerowej, wykorzystujących transmisję strumieniową

### 3.17. Funkcje gniazda wykorzystywane po stronie klienta i serwera

#### 3.17.1. Funkcja socket

Funkcja `socket` tworzy gniazdo i zwraca liczbową wartość deskryptora:

```
deskryptor = socket(rodzina_protokolow, typ, protokol)
```

Parametr `rodzina_protokolow` określa grupę protokołów przypisaną gniazdu. Charakterystyczny dla internetu stos TCP/IP jest oznaczany za pomocą identyfikatora `PF_INET`.

Parametr `typ` definiuje rodzaj komunikacji. Strumieniowe przesyłanie danych jest opisywane za pomocą wartości `SOCK_STREAM`. Natomiast bezpołączeniowy transfer komunikatów wskazuje wartości `SOCK_DGRAM`.

Ostatni parametr (`protokol`) odpowiada konkretному protokołowi wykorzystywanemu w pracy gniazda. Dzięki temu, że poza typem komunikacji istnieje możliwość określenia również samego protokołu, dwa lub więcej protokołów jednej rodziny może realizować tę samą usługę. Wartości parametru `protokol` zależą od wskazanej rodziny protokołów.

### 3.17.2. Funkcja send

Zarówno klient, jak i serwer wykorzystują funkcję `send` do wysyłania danych. Zazwyczaj za jej pomocą klient generuje żądania, a serwer odsyła odpowiedzi. Funkcja `send` pobiera cztery parametry:

```
send(socket, data, length, flags)
```

Parametr `socket` odpowiada deskryptoriowi utworzonego wcześniej gniazda. Druga wartość (`data`) wyznacza adres w pamięci, pod którym przechowywane są dane przeznaczone do przesłania. Parametr `length` jest wartością całkowitą określającą liczbę bajtów danych. Natomiast wartość `flags` składa się z bitów włączających pewne dodatkowe opcje gniazda<sup>10</sup>.

### 3.17.3. Funkcja recv

Klient i serwer wykorzystują funkcję `recv` do pobierania danych przesłanych przez drugą stronę. Składnia wywołania funkcji jest następująca:

```
recv(socket, buffer, length, flags)
```

Parametr `socket` jest deskryptorem gniazda, z którego dane zostaną pobrane. Wartość `buffer` wyznacza adres w pamięci, pod którym powinna zostać zapisana odebrana informacja. Parametr `length` określa rozmiar tego bufora. Ostatnia wartość (`flags`) umożliwia modyfikację standardowego działania funkcji (na przykład pobranie nadchodzącej wiadomości, ale bez usunięcia oryginalnego komunikatu z gniazda). Funkcja `recv` blokuje działanie programu aż do odebrania danych, a następnie umieszcza je w buforze. Liczba zapisanych w buforze bajtów nie przekracza wartości `length`. Wynikiem wywołania funkcji jest dokładna liczba pobranych bajtów.

---

<sup>10</sup> Wiele ze wspomnianych opcji jest przeznaczonych do uruchamiania kodu i nie jest wykorzystywanych w klasycznych interakcjach klient-serwer.

### 3.17.4. Odczyt i zapis danych gniazd

W niektórych systemach operacyjnych (na przykład w systemie Linux) zamiast funkcji `recv` i `send` można stosować standardowe funkcje `read` i `write`. Funkcja `read` pobiera trzy parametry, odpowiadające dokładnie parametrom wywołania `recv`. Również funkcja `write` wymaga określenia dokładnie takich samych wartości jak funkcja `send`.

Najważniejszą zaletą stosowania wywołań `read` i `write` jest uogólnienie kodu — można napisać program w taki sposób, aby przekazywał dane do komponentu o określonym deskryptorze i odbierał je z określonego komponentu bez uszczegóławiania, czy jest to plik, czy gniazdo. Dzięki temu przed zaimplementowaniem komunikacji sieciowej programista może testować rozwiązywanie na lokalnych plikach dyskowych. Wadą stosowania funkcji `read` i `write` jest jednak to, że przed przeniesieniem kodu do innego systemu często konieczna jest zmiana wywołań.

### 3.17.5. Funkcja close

Funkcja `close` nakazuje systemowi operacyjnemu usunięcie gniazda<sup>11</sup>. Oto jej składnia:

```
close(socket)
```

Parametr `socket` odpowiada oczywiście gniazdu, które powinno zostać usunięte. Jeśli w chwili realizacji instrukcji połączenie jest nawiązane, funkcja `close` doprowadzi do jego zakończenia (tj. poinformuje drugą stronę o zakończeniu połączenia). Zamknięcie gniazda skutkuje natychmiastowym przerwaniem wymiany danych — deskryptor zostaje zwolniony, co uniemożliwia aplikacji wysyłania i odbierania informacji.

## 3.18. Funkcja połączenia wykorzystywana jedynie po stronie klienta

Aby ustanowić połączenie z serwerem, program kliencki wywołuje funkcję `connect`. Składnię instrukcji pokazano poniżej.

```
connect(socket, saddress, saddresslen)
```

Parametr `socket` odpowiada deskryptorowi gniazda wykorzystywanego w połączeniu. Wartość `saddress` jest strukturą typu `sockaddr`, która przechowuje adres serwera oraz numer portu protokołu<sup>12</sup>. Parametr `saddresslen` wyznacza długość adresu serwera liczoną w bajtach.

<sup>11</sup> W interfejsie Windows Socket firmy Microsoft funkcję `close` zastępuje funkcja `closesocket`.

<sup>12</sup> Połączenie adresu IP i numeru portu jest często nazywane adresem punktu końcowego (ang. *endpoint address*).

W przypadku gniazd przeznaczonych do transmisji strumieniowej funkcja `connect` inicjuje połączenie z określonym serwerem na poziomie warstwy transportowej. Serwer musi oczekwać na takie połączenie (zobacz opisaną poniżej funkcję `accept`).

## 3.19. Funkcje gniazd wykorzystywane jedynie po stronie serwera

### 3.19.1. Funkcja bind

Po utworzeniu gniazda nie przechowuje jakichkolwiek informacji na temat lokalnego lub zdalnego adresu ani numeru portu protokołu. W celu dostarczenia informacji o numerze portu, na którym prowadzony będzie nasłuch, program serwerowy musi wywołać funkcję `bind`. Oto jej składnia:

```
bind(socket, localaddr, addrlen)
```

Parametr `socket` odpowiada deskryptorowi wykorzystywanego gniazda. Wartość `localaddr` jest strukturą wyznaczającą adres przypisywany do danego gniazda. Z kolei parametr `addrlen` to liczba całkowita odzwierciedlająca długość adresu liczoną w bajtach.

Ponieważ gniazdo może być wykorzystywane przez różne protokoły, format adresu jest zależny od konkretnego zastosowania. Interfejs API definiuje więc ogólną strukturę adresu oraz opisuje sposób wykorzystania ogólnego zapisu w ramach określonej rodziny protokołów. Ogólny format adresu wyznacza struktura `sockaddr` składająca się z trzech pól (choć istnieją również inne warianty tej struktury):

```
struct sockaddr {
    u_char sa_len;          /* całkowita długość adresu */
    u_char sa_family;        /* rodzina adresu */
    char sa_data[14];        /* właściwy adres */
};
```

Pole `sa_len` składa się z jednego oktetu wyznaczającego rozmiar adresu. Pole `sa_family` określa rodzinę, do której należy dany adres (stała odpowiadająca adresom internetowym to `AF_INET`). Sam adres jest przechowywany w polu `sa_data`.

Każda rodzina protokołów zawiera dokładną definicję adresów zapisywanych w polu `sa_data` struktury `sockaddr`. Na przykład protokoły internetowe bazują na strukturze `sockaddr_in`:

```
struct sockaddr_in {
    u_char sin_len;          /* całkowita długość adresu */
    u_char sin_family;        /* rodzina adresu */
    u_short sin_port;         /* numer portu protokołu */
    struct in_addr sin_addr;  /* adres IP komputera */
    char sin_zero[8];         /* niewykorzystywane (wypełnione zerami) */
};
```

Dwa pierwsze pola struktury `sockaddr_in` odpowiadają dokładnie dwóm pierwszym polom ogólnej struktury `sockaddr`. Z kolei trzy kolejne wartości wyznaczają format adresu internetowego. Warto tutaj zwrócić uwagę na dwa elementy. Po pierwsze, każdy

adres zawiera informacje zarówno o komputerze, jak i o porcie protokołu wykorzystywanym w tym komputerze. Pole `sin_addr` przechowuje adres IP jednostki. Natomiast pole `sin_port` zawiera numer portu protokołu. Po drugie, mimo że do zapisania samego adresu potrzebnych jest sześć bajtów, w ogólnej strukturze adresu zarezerwowanych jest na ten cel czternaście bajtów. Przez to ostatnie pole w strukturze `sockaddr_in` jest wypełnione ośmioma bajtami o zerowej wartości (dzięki temu rozmiary struktur są jednakowe).

Zgodnie z wcześniejszym stwierdzeniem funkcja `bind` służy do wyznaczenia numeru portu, na którym serwer oczekuje na nadchodzące żądania. Jednak poza samym numerem portu protokołu struktura `sockaddr_in` zawiera również informacje o adresie. Choć serwer może wykorzystać ją do określenia adresu, rozwiązanie to bywa kłopotliwe, szczególnie gdy jednostka korzysta z większej liczby interfejsów sieciowych (dysponuje wówczas wieloma adresami sieciowymi). Aby umożliwić programom serwerowym pracę w systemach o wielu interfejsach sieciowych, interfejs API gniazd uwzględnia specjalną stałą `INADDR_ANY`, która pozwala na wskazanie numeru portu z uwzględnieniem wszystkich adresów danego komputera. Podsumowując:

*Mimo że struktura `sockaddr_in` zawiera pole adresu, interfejs API gniazd udostępnia stałą, która umożliwia przypisanie określonego numeru portu do wszystkich adresów komputera.*

### 3.19.2. Funkcja listen

Po wykonaniu instrukcji `bind` do określenia portu protokołu serwer wywołuje funkcję `listen` w celu przełączenia gniazda w tryb pasywny, co z kolei uruchamia oczekiwanie na połączenia ze strony klientów. Funkcja `listen` pobiera dwa parametry:

```
listen(socket, queuesize)
```

Parametr `socket` odpowiada deskryptorowi gniazda, a wartość `queuesize` wyznacza rozmiar kolejki żądań gniazda. System operacyjny wyznacza dla każdego gniazda oddzielną kolejkę żądań. Początkowo jest ona pusta. Jednak wraz z odbieraniem żądań klienckich każde takie żądanie jest zapisywane w kolejce. Gdy serwer zainicjuje operację dostarczenia odebranego żądania, system pobierze kolejne żądanie z kolejki. Wartość rozmiaru kolejki jest niezwykle istotna, ponieważ zapełnienie kolejki powoduje odrzucanie wszystkich kolejnych nadchodzących żądań.

### 3.19.3. Funkcja accept

Serwer wywołuje funkcję `accept` w celu ustanowienia połączenia z klientem. Jeśli w kolejce zostało zarejestrowane jakiekolwiek żądanie, wywołanie funkcji `accept` kończy się bezzwłocznie. Jeżeli jednak żadne żądanie nie zostało wcześniej zbuforowane, system blokuje wykonywanie programu serwerowego do czasu dostarczenia żądania ze strony klienta.

Po zaakceptowaniu połączenia serwer wykorzystuje je do wymiany danych z klientem. Z chwilą zakończenia komunikacji serwer zamknie połączenie.

Funkcja `accept` ma następującą składnię:

```
newsock = accept(socket, caddress, caddresslen)
```

Parametr `socket` odpowiada deskryptoriowi gniazda, które zostało utworzone po stronie serwera i skojarzone z określonym portem protokołu. Wartość `caddress` jest adresem o strukturze `sockaddr`, a parametr `caddresslen` to wskaźnik na wartość całkowitą. Funkcja `accept` wypełnia obszar `caddress` adresem klienta nawiązującego połączenia i zapisuje w obszarze `caddresslen` rozmiar tego adresu. Następnie powołuje nowe gniazdo (`newsock`) do obsługi połączenia i zwraca deskryptor tego gniazda do funkcji wywołującej. Nowe gniazdo jest wykorzystywane przez serwer do komunikacji z klientem, a po zakończeniu wymiany danych zostaje usunięte. Pierwotne gniazdo serwerowe działa zgodnie z wcześniejszymi założeniami — po zakończeniu komunikacji z klientem serwer wykorzystuje pierwotne gniazdo do ustanawiania kolejnych połączeń z jednostkami klienckimi. Oryginalne gniazdo służy więc jedynie do rejestrowania nachodzących żądań, a zasadnicza wymiana danych odbywa się z zastosowaniem gniazda utworzonego dynamicznie przez funkcję `accept`.

## 3.20. Funkcje gniazd wykorzystywane w transmisji komunikatów

Funkcje gniazd przeznaczone do wysyłania i odbierania komunikatów są znacznie bardziej skomplikowane niż instrukcje operujące strumieniami, ponieważ wykorzystują wiele dodatkowych opcji. Na przykład nadawca może określić, czy chce zachować adres odbiorcy w gnieździe i ograniczyć kolejne operacje jedynie do wysyłania danych, czy też będzie definiował adres odbiorcy podczas każdorazowego generowania komunikatu. Co więcej, pewne funkcje umożliwiają nadawcy zdefiniowanie adresu i komunikatu w jednej strukturze, która następnie zostanie przekazana jako parametr wywołania tej funkcji, a inne instrukcje wymagają określenia adresu i komunikatu jako niezależnych parametrów.

### 3.20.1. Funkcje `sendto` i `sendmsg`

Funkcje `sendto` i `sendmsg` umożliwiają oprogramowaniu klienckiemu i serwerowemu wysyłanie komunikatów za pomocą niepołączonych gniazd. Obie wymagają od programu wywołującego określenia jednostki docelowej. Funkcja `sendto` korzysta z odrębnych parametrów komunikatu i adresu docelowego:

```
sendto(socket, data, length, flags, destaddress, addresslen)
```

Cztery pierwsze parametry odpowiadają dokładnie parametrom funkcji `send`. Ostatnie dwa wyznaczają natomiast adres jednostki docelowej oraz długość pola adresu. Wartość `destaddress` jest strukturą `sockaddr` (a w szczególności `sockaddr_in`).

Funkcja `sendmsg` wykonuje to samo zadanie co `sendto`, ale dzięki specjalnej strukturze wykorzystuje skróconą listę parametrów. Mniejsza liczba parametrów sprawia, że programy bazujące na funkcjach `sendmsg` są łatwiejsze do analizowania:

```
sendmsg(socket, msgstruct, flags)
```

Parametr `msgstruct` jest strukturą, która zawiera informacje na temat adresu docelowego, długości adresu, przesyłanego komunikatu oraz długości tego komunikatu:

```
struct msgstruct {           /* struktura wykorzystywana przez funkcję sendmsg */
    struct sockaddr *m_saddr;  /* wskaźnik na adres docelowy */
    struct datavec *m_dvec;   /* wskaźnik na komunikat (wektor) */
    int m_dvlength;          /* liczba elementów w wektorze */
    struct access *m_rights; /* wskaźnik na listę uprawnień */
    int m_alength;           /* liczba elementów na liście */
};
```

Znaczenie poszczególnych pól struktury nie jest istotne — warto jedynie zapamiętać, że istnieje możliwość połączenia wielu wartości w jedną strukturę. Większość aplikacji wykorzystuje tylko trzy pierwsze pola, które określają docelowy adres, listę elementów danych składających się na komunikat oraz liczbę elementów listy.

### 3.20.2. Funkcje `recvfrom` i `recvmsg`

Niepołączone gniazdo może być wykorzystywane do odbierania komunikatów od nieustalonej grupy klientów. W takim przypadku system dostarcza informacje o adresie nadawcy wraz z każdym nadchodzącym komunikatem (odbiorca wykorzystuje następnie dany adres do wygenerowania odpowiedzi). Funkcja `recvfrom` wymaga przekazania parametru, który określa adres nadawcy kolejnego komunikatu:

```
recvfrom(socket, buffer, length, flags, sndraddr, saddrlen)
```

Cztery pierwsze parametry odpowiadają dokładnie wartościom funkcji `recv`. Dwa dodatkowe (`sndraddr` i `saddrlen`) są natomiast wykorzystywane do zapisania adresu internetowego nadawcy. Wartość `sndraddr` jest wskaźnikiem na strukturę `sockaddr`, w której system zapisuje adres nadawcy. Z kolei `saddrlen` jest wskaźnikiem na liczbę całkowitą, reprezentującą długość adresu. Funkcja `recvfrom` przechowuje adres nadawcy w dokładnie taki sam sposób, w jaki jest on przetwarzany w funkcji `sendto`, co upraszcza generowanie odpowiedzi.

Funkcja `recvmsg`, będąca uzupełnieniem instrukcji `sendmsg`, działa analogicznie do funkcji `recvfrom`, ale wymaga określenia mniejszej liczby parametrów. Jej składnia to:

```
recvmsg(socket, msgstruct, flags)
```

Parametr `msgstruct` przekazuje adres struktury zawierającej adres internetowy źródła wiadomości oraz adres nadawcy. Wartość `msgstruct` rejestrowana przez funkcję `recvmsg` ma dokładnie taki sam format, jaki jest wymagany przez funkcję `sendmsg`. Dzięki temu wysyłanie odpowiedzi staje się znacznie łatwiejsze.

### 3.21. Inne funkcje gniazd

Interfejs API gniazd zawiera wiele funkcji wspomagających komunikację. Na przykład po zaakceptowaniu przez serwer nadchodzącego połączenia, oprogramowanie serwerowe może wykonać instrukcję `getpeername` w celu pobrania adresu jednostki, która zainicjowała połączenie. Klient lub serwer mogą także wywołać funkcję `gethostname`, aby pozyskać informacje o komputerze, na którym program został uruchomiony.

Operowanie opcjonalnymi ustawieniami gniazd należy do zadań dwóch specjalnych funkcji. Funkcja `setsockopt` zapisuje wartości w strukturze opcji. Natomiast funkcja `getsockopt` służy do pobierania tych wartości. Za pomocą opcji można obsługiwać niestandardowe przypadki użycia gniazd (na przykład zwiększać rozmiar wewnętrznego bufora).

Za tłumaczenie adresów internetowych na nazwy jednostek odpowiadają dwie instrukcje. Funkcja `gethostbyname` zwraca adres internetowy komputera o określonej nazwie. Oprogramowanie klienckie często wykonuje instrukcję `gethostbyname`, aby przekształcić nazwę wprowadzoną przez użytkownika w adres IP wskazywanej jednostki. Funkcja `gethostbyaddr` dokonuje odwrotnego odwzorowania — zwraca nazwę komputera na podstawie jego adresu IP. Programy klienckie i serwerowe mogą wykorzystywać instrukcję `gethostbyaddr` do przekształcania adresów w nazwy zrozumiałe dla użytkowników.

### 3.22. Gniazda, wątki i dziedziczenie

Interfejs API gniazd doskonale sprawdza się w wielowątkowych aplikacjach serwerowych. Choć szczegóły implementacyjne zależą od konkretnego systemu operacyjnego, ogólne zasady stosowania interfejsu są zgodne z następującą regułą:

*Każdy nowo tworzony wątek dziedziczy z wątku bazowego kopie wszystkich otwartych gniazd.*

Implementacja gniazd zawiera mechanizm *zliczania referencji*, nadzorujący pracę każdego gniazda. Podczas tworzenia gniazda system ustawia licznik referencji na 1. Gniazdo istnieje, dopóki wartość licznika jest dodatnia. Wraz z utworzeniem nowego wątku wątek ten dziedziczy wskaźniki do każdego otwartego przez program gniazda, a system zwiększa licznik referencji każdego gniazda o 1. Gdy wątek wywołuje funkcję `close`, system zmniejsza licznik danego gniazda. Z chwilą wyzerowania licznika referencji gniazdo jest usuwane.

W przypadku jednoczesnej pracy wielu wątków serwerowych wątek główny jest właściwym gniazdem wykorzystywanego do odbierania nadchodzących połączeń. Wątek główny odpowiada również za tworzenie wątków potomnych, których zadanie polega na obsłudze połączeń. W momencie powołania nowego wątku obydwa wątki mają dostęp do gniazda pierwotnego oraz nowo utworzonego, a licznik referencji każdego z gniazd wynosi 2. Wątek potomny wywołuje funkcję `close` nowego gniazda, a wątek usługi wywołuje funkcję `close` w odniesieniu do gniazda podstawowego. W obydwu przypadkach licznik referencji jest zmniejszany do wartości 1. Gdy komunikacja z klientem zostanie zakończona, wątek usługi wywołuje funkcję `close` w odniesieniu do nowego gniazda, zmniejszając jego licznik do zera, co z kolei powoduje usunięcie tego gniazda. Czas życia gniazd można zatem opisać w następujący sposób:

*Gniazdo wykorzystywane do odbierania połączeń istnieje tak długo, jak długo działa główny wątek serwera. Gniazda przeznaczone do realizacji konkretnych połączeń są utrzymywane jedynie na czas działania wątku obsługującego tych połączeń.*

### 3.23. Podsumowanie

Wszystkie usługi internetowe są realizowane przez aplikacje, które wykorzystują strumienie lub komunikaty do wymiany danych z klientami. Strumienie gwarantują dostarczenie sekwencji bajtowych w niezmienionej kolejności, ale nie pozwalają na określenie wielkości każdej porcji danych. Przekazywanie komunikatów umożliwia zachowanie rozmiaru poszczególnych bloków danych, ale jest obarczone ryzykiem utraty, powielenia lub zakłócenia kolejności dostarczanych informacji.

Podstawowy model wymiany danych w sieci jest określany jak model klient-serwer. Program oczekujący na połączenia pełni w nim funkcję serwera. Z kolei aplikacja inicjująca połączenie jest nazywana klientem.

Każdy komputer otrzymuje niepowtarzalny adres, a każda usługa (na przykład usługa poczty elektronicznej lub stron internetowych) dysponuje identyfikatorem określonym jako numer portu protokołu. Numer portu jest definiowany przez serwer w chwili uruchomienia programu serwerowego. Podczas odwoływania się do usługi aplikacja kliencka wyznacza zarówno adres komputera zdalnego, jak również numer portu protokołu, na którym serwer prowadzi nasłuch.

Pojedynczy klient może się komunikować z większą liczbą serwerów. Serwery te mogą być uruchomione na różnych komputerach. Dany serwer może ponadto pełnić rolę klienta w połączeniu z inną usługą. Projektując tego typu systemy, trzeba jednak zachować szczególną ostrożność, aby nie doszło do zapętlenia odwołań między jednostkami.

Zasady współdziałania programów z oprogramowaniem protokołów komunikacyjnych reguluje interfejs programistyczny gniazd. Choć implementacje tego mechanizmu są różne w poszczególnych systemach operacyjnych, interfejs API gniazd jest *de facto* standardem. Programy sieciowe tworzą gniazda, a następnie wywołują różne funkcje umożliwiające wykorzystanie tych gniazd. Serwer komunikacji strumieniowej posługuje

się funkcjami `socket`, `bind`, `listen`, `accept`, `recv`, `send` i `close`. Aplikacja kliencka wywołuje natomiast funkcje `socket`, `connect`, `send`, `recv` oraz `close`.

Ponieważ wiele usług obsługuje jednocześnie połączenia, gniazda zostały zaprojektowane w taki sposób, aby umożliwiał działywanie współbieżnych aplikacji. Nowo utworzony wątek ma bowiem dostęp do wszystkich gniazd wątku nadziednego.

## ZADANIA

- 3.1. Jakie są dwa podstawowe rodzaje komunikacji internetowej?
- 3.2. Podaj sześć cech komunikacji strumieniowej.
- 3.3. Podaj sześć cech komunikacji z wykorzystaniem komunikatów.
- 3.4. Jeśli nadawca wykorzystuje komunikację strumieniową i zawsze generuje bloki danych złożone z 1024 bajtów, jaki może być rozmiar bloków danych dostarczanych przez sieć do odbiorcy?
- 3.5. Jeśli nadawca chce przekazywać kopie każdego bloku danych do trzech odbiorców, który rodzaj komunikacji powinien zastosować?
- 3.6. Wymień trzy wady dostarczania komunikatów przez internet.
- 3.7. Przedstaw ogólny algorytm działania systemów połączeniowych.
- 3.8. Któż spośród dwóch aplikacji internetowych jest serwerem?
- 3.9. Porównaj aplikacje klienckie i serwerowe i wskaź różnicę między nimi.
- 3.10. Jaka jest różnica między serwerem a komputerem serwerowym?
- 3.11. Czy dane mogą być przesyłane z jednostki klienckiej do serwerowej? Wyjaśnij zagadnienie.
- 3.12. Wymień możliwe kombinacje serwerów i klientów, które mogą działać w danym komputerze.
- 3.13. Czy wszystkie komputery mogą udostępniać wiele usług w sposób efektywny? Uzasadnij odpowiedź.
- 3.14. Za pomocą jakich (dwóch) identyfikatorów opisywany jest dany serwer?
- 3.15. Wymień wszystkie kroki, które klient musi wykonać, aby nawiązać połączenie z serwerem po tym, jak użytkownik określi nazwę domenową serwera.
- 3.16. Jakkie funkcje systemu operacyjnego wykorzystuje serwer do obsługi jednoczesnych żądań z wielu jednostek klienckich?
- 3.17. Jakkie problemy wydajnościowe uzasadniają stosowanie komunikacji P2P?
- 3.18. Wymień dwa systemy operacyjne udostępniające interfejs API gniazd.
- 3.19. W jaki sposób aplikacja odwołuje się do gniazda po jego utworzeniu?
- 3.20. Wymień podstawowe funkcje interfejsu API gniazd.
- 3.21. Przedstaw typową sekwencję odwołań do gniazda realizowaną po stronie serwera i klienta.
- 3.22. Jakim funkcjom gniazd odpowiadają funkcje `read` i `write`?
- 3.23. Czy klient posługuje się funkcją `bind`? Uzasadnij odpowiedź.
- 3.24. Dlaczego stosowana jest stała `INADDR_ANY`?
- 3.25. Czy funkcja `sendto` jest wykorzystywana w komunikacji strumieniowej, czy podczas przesyłania komunikatów?

- 3.26. Założmy, że gniazdo jest otwarte i został utworzony nowy wątek. Czy nowo utworzony wątek będzie mógł skorzystać z gniazda?
- 3.27. Przeanalizuj kod serwera WWW zamieszczony w Dodatku A, a następnie opracuj analogiczny kod serwerowy z wykorzystaniem interfejsu API gniazd.
- 3.28. Zaimplementuj interfejs przedstawiony w Dodatku A za pomocą funkcji gniazd.

{}



# Zawartość rozdziału

- }  
4.1. Wprowadzenie 79  
4.2. Protokoły warstwy aplikacji 79  
4.3. Reprezentacja i transfer danych 80  
4.4. Protokoły WWW 81  
4.5. Reprezentacja dokumentów w standardzie HTML 81  
4.6. Ujednolicony format adresowania zasobów i odsyłacze 83  
4.7. Dostarczanie dokumentów za pomocą protokołu HTTP 84  
4.8. Buforowanie stron w przeglądarkach 87  
4.9. Budowa przeglądarki 88  
4.10. Protokół transferu plików (FTP) 88  
4.11. Komunikacja FTP 89  
4.12. Poczta elektroniczna 92  
4.13. Prosty protokół dostarczania poczty (SMTP) 93  
4.14. Dostawcy usług internetowych, serwery pocztowe i dostęp do poczty elektronicznej 95  
4.15. Protokoły dostępu do poczty (POP, IMAP) 96  
4.16. Standardy zapisu wiadomości e-mail (RFC2822, MIME) 97  
4.17. System nazw domenowych (DNS) 98  
4.18. Nazwy domenowe rozpoczynające się od www 100  
4.19. Hierarchia DNS i model powiązań serwerowych 101  
4.20. Odwzorowanie nazw 101  
4.21. Buforowanie danych w systemie DNS 103  
4.22. Rodzaje wpisów DNS 104  
4.23. Aliasy nazw i rekordy CNAME 105  
4.24. Skróty w systemie DNS 106  
4.25. Znaki narodowe w nazwach domenowych 106  
4.26. Rozszerzalne formaty reprezentacji danych (XML) 107  
4.27. Podsumowanie 108

# 4

## *Typowe aplikacje internetowe*

### 4.1. Wprowadzenie

Wcześniejsze rozdziały stanowiły wprowadzenie do zagadnień związanych z internetem i programowaniem sieciowym. Celem tego rozdziału jest przedstawienie usług internetowych jako aplikacji programowych oraz szczegółowe zaprezentowanie modelu klient-serwer, który jest wykorzystywany przez wspomniane aplikacje do wymiany danych. Omówienie obejmuje również interfejs API gniazd.

W rozdziale tym kontynuowany jest przegląd aplikacji internetowych. Wyjaśniona została koncepcja protokołu transmisyjnego, a także sposób implementowania protokołów transmisyjnych w aplikacjach sieciowych. W końcowej części znajduje się omówienie standardowych aplikacji internetowych oraz wykorzystywanych przez nie protokołów transportowych.

### 4.2. Protokoły warstwy aplikacji

Za każdym razem, gdy programista tworzy dwie aplikacje komunikujące się przez sieć, musi zastanowić się nad następującymi elementami rozwiązania:

- składnia i semantyka wymienianych komunikatów;
- czy interakcję inicjuje klient, czy serwer;
- działania podejmowane w przypadku wystąpienia błędów;
- sposób poinformowania obydwu stron o obowiązku zakończenia połączenia.

Definiując szczegóły komunikacji, programista tworzy **protokół warstwy aplikacji**. Wyróżnia się dwa rodzaje protokołów warstwy aplikacji w zależności od ich przeznaczenia:

- **Komunikację prywatną.** Programista opracowuje dwie aplikacje, które komunikują się za pośrednictwem internetu z założeniem, że powstały system jest przeznaczony do użytku prywatnego. W większości przypadków interakcje między programami nie są szczególnie skomplikowane, dzięki czemu twórcy oprogramowania mogą pisać kod bez formalnej specyfikacji protokołu.
- **Standardowe usługi.** Usługi internetowe są opracowywane w nadziei, że tworzeniem aplikacji udostępniających daną usługę użytkownikom będzie się zajmowało wielu niezależnych programistów. W takich przypadkach protokół warstwy aplikacji musi być udokumentowany i niezależny od implementacji. Dodatkowo specyfikacja musi być dostatecznie przejrzysta i jednoznaczna, aby powstające programy klienckie i serwerowe mogły ze sobą poprawnie współdziałać.

Rozmiar samego dokumentu opisującego protokół zależy od złożoności usługi. Specyfikacje prostych usług zajmują jedną stronę tekstu. Na przykład wśród wielu standardowych usług internetowych znajduje się usługa DAYTIME, która umożliwia klientom pobranie daty i czasu zgodnie z ustawieniami serwera. Protokół komunikacyjny jest w niej wyjątkowo łatwy w implementacji — klient ustanawia połączenie z serwerem, serwer wysyła wartość daty i czasu w formacie ASCII, a następnie kończy połączenie. Odpowiedź serwera może mieć treść:

```
Sat Sep 9 20:18:37 2008
```

Klient pobiera dane aż do napotkania **znacznika końca pliku**. Podsumowując:

*Aby umożliwić aplikacjom współdziałanie ze standardowymi aplikacjami, konieczne jest opracowanie standardu protokołu warstwy aplikacji, który jest niezależny od konkretnej implementacji mechanizmu komunikacyjnego.*

### 4.3. Reprezentacja i transfer danych

Protokoły warstwy aplikacji definiują dwa aspekty interakcji programów — reprezentację i sposób transferu danych. W wyjaśnieniu tego zagadnienia pomocna jest tabela 4.1.

Tabela 4.1. Dwa kluczowe aspekty działania protokołu warstwy aplikacji

Aspekt	Opis
Reprezentacja danych	Składnia wymienianych danych, szczególna postać informacji wykorzystywana do transmisji, reprezentacja wartości liczbowych oraz treści plików.
Transfer danych	Interakcje między klientem i serwerem, składnia i znaczenie wiadomości, dopuszczalne i niedopuszczalne formy wymiany danych, obsługa błędów, zakończenie interakcji.

W przypadku nieskomplikowanych usług obydwa aspekty funkcjonowania mogą być opisane za pomocą pojedynczego standardu protokołu. Bardziej złożone serwisy

często bazują na oddzielnych standardach. Na przykład wspomniany wcześniej protokół DAYTIME został opisany w pojedynczej specyfikacji, która zawiera informacje o tym, że data i czas są reprezentowane za pomocą łańcucha ASCII oraz że transfer danych ogranicza się do przesłania odpowiedzi ze strony serwera i zamknięcia połączenia. W kolejnym punkcie został natomiast opisany mechanizm dostarczania stron WWW, który bazuje na oddzielnym protokole opisu strony oraz protokole transferu treści. Twórcy protokołów wyraźnie rozgraniczają te dwa elementy komunikacji:

*Zgodnie z przyjętą konwencją słowo transfer występujące w nagłówku specyfikacji protokołu warstwy aplikacji oznacza, że protokół odnosi się do części związanej z przekazywaniem danych.*

## 4.4. Protokoły WWW

Usługa WWW wydaje się najpowszechniej wykorzystywanaą obecnie usługą internetową. Jednak z uwagi na dużą złożoność mechanizmu zasady jego funkcjonowania zostały opisane w kilku standardach protokołów, odnoszących się do różnych aspektów pracy. Najważniejsze standardy wymieniono w tabeli 4.2.

Tabela 4.2. Najważniejsze standardy usługi WWW

Standard	Przeznaczenie
Hipertekstowy język znaczników (HTML — <i>HyperText Markup Language</i> )	Standard reprezentacji danych, wykorzystywany do opisu treści oraz szaty graficznej strony internetowej.
Ujednolicony format adresowania zasobów (URL — <i>Uniform Resource Locator</i> )	Standard reprezentacji danych, określający format i znaczenie identyfikatorów stron internetowych.
Protokół transferu dokumentów hipertekstowych (HTTP — <i>HyperText Transfer Protocol</i> )	Protokół transmisyjny, który określa zasady interakcji między przeglądarką i serwerem WWW w czasie przesyłania danych.

## 4.5. Reprezentacja dokumentów w standardzie HTML

Hipertekstowy język znaczników (HTML) jest standardem reprezentacji danych, który określa składnię stron internetowych. Oto kilka najważniejszych jego cech:

- Bazuje na tekściej reprezentacji treści.
- Opisuje strony zawierające elementy multimedialne.
- Wykorzystuje deklaracyjną, a nie proceduralną formę zapisu.
- Zamiast formatowania uwzględnia opis znacznikowy.

- Umożliwia osadzanie odsyłaczy do dowolnych obiektów.
- Pozwala na włączanie do dokumentu metadanych.

Mimo że dokument HTML jest plikiem tekstowym, programista może zawrzeć w nim dowolnie skomplikowaną treść obejmującą grafikę, sekwencje audio i wideo oraz sam tekst. W zasadzie twórcy rozwiązania powinni wykorzystać w nazwie termin *hipermedia* zamiast *hipertekst*, ponieważ dowolny obiekt strony HTML (na przykład rysunek) może stanowić odsyłacz do innej strony (nazywany niekiedy *hiperłączem*).

Dokument HTML jest klasyfikowany jako **deklaratywny**, ponieważ język opisu umożliwia autorom stron określenie rodzaju wyświetlanej treści oraz sposobu jej prezentacji. Stwierdzenie, że jest **językiem znacznikowym**, wynika z tego, że zawiera ogólne wskazówki na temat wyświetlania elementów strony, a nie uwzględnia szczegółowych instrukcji formatowania. Na przykład projektant witryny może zdefiniować stopień ważności nagłówka, ale nie musi określić rodzaju czcionki, jej rozmiaru oraz odstępów między literami<sup>13</sup>. Zadanie wyboru odpowiedniego sposobu prezentacji należy do przeglądarki. Dzięki stosowaniu zapisu znacznikowego przeglądarka może dostosować daną stronę do parametrów wyświetlacza urządzenia, w którym jest uruchomiona. Dzięki temu dokument może być poprawnie prezentowany zarówno na ekranach o wysokiej rozdzielcości, jak i na wyświetlaczach o niskiej rozdzielcości (takich, jakie są montowane w urządzeniach przenośnych — iPhone lub PDA).

Podsumowując:

*Hipertekstowy język znaczników jest standardem opisu stron internetowych. Aby strona została wyświetlona w wybranym urządzeniu, przeglądarka musi odczytać z dokumentu HTML ogólne wskazówki na temat prezentacji treści i samodzielnie ustalić szczegóły formatowania.*

Opis dokumentu HTML bazuje na **znacznikach** (ang. *tags*). Znacznik jest pojedynczym słowem otoczonym znakami mniejszości i większości. Jego zadanie polega na dostarczeniu informacji na temat struktury dokumentu, a także wskazówek odnośnie formatowania strony. Tylko znaczniki regulują sposób prezentacji treści — ewentualne wstawienie nadmiarowych znaków odstępu (dodatkowych wierszy i spacji) nie wpływa w jakikolwiek sposób na wygląd strony w przeglądarce.

Dokument HTML rozpoczyna się znacznikiem `<html>` i kończy znacznikiem `</html>`. Para `<head>` i `</head>` wyznacza obszar nagłówka strony, natomiast para `<body>` i `</body>` wskazuje treść. Umieszczone w sekcji nagłówka znaczniki `<title>` i `</title>` otaczają tekst będący tytułem strony. Ogólna struktura dokumentu HTML została przedstawiona na rysunku 4.1.

Do osadzania obrazów w treści strony HTML służy znacznik `<img>`. Oto przykład jego zastosowania:

```

```

---

<sup>13</sup> Istnieją rozszerzenia kodu HTML, które zapewniają możliwość zdefiniowania rodzaju czcionki, rozmiaru oraz innych elementów formatowania.

```

<html>
  <head>
    <title>
      Tekst stanowiący tytuł dokumentu
    </title>
  </head>
  <body>
    Zasadnicza treść dokumentu
  </body>
</html>

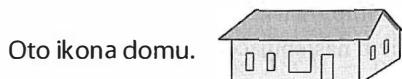
```

Rysunek 4.1. Ogólna struktura dokumentu HTML

Taki zapis oznacza, że plik *ikona\_domu.gif* zawiera obraz, który przeglądarka powinna wstawić w danym miejscu dokumentu. Ewentualne dodatkowe atrybuty znacznika *<img>* pozwalają na określenie sposobu otaczania obrazu tekstem strony. Na rysunku 4.2 przedstawiono wynik interpretacji poniższego kodu HTML, który zapewnia wyśrodkowanie rysunku względem tekstu:

Oto ikona domu. 

Przeglądarka umieści rysunek w taki sposób, aby był on wyśrodkowany w pionie względem tekstu.



Rysunek 4.2. Przykład rozmieszczenia elementów w dokumencie HTML

## 4.6. Ujednolicony format adresowania zasobów i odsyłacze

Aby wskazać stronę internetową w sieci, wykorzystywane są ciągi **ujednoliconego formatu adresowania zasobów** (adresy URL). Ogólna postać adresu URL to:

protokół://nazwa\_komputera:port/nazwa\_dokumentu?parametry

Człon *protokół* odpowiada nazwie protokołu, który jest wykorzystywany do pobrania dokumentu. Fragment *nazwa\_komputera* wskazuje nazwę domenową jednostki udostępniającej dokument, natomiast *:port* jest opcjonalną definicją portu protokołu, na którym serwer nasłuchuje żądań klienckich. Ciąg *nazwa\_dokumentu* ma charakter opcjonalny i oczywiście odnosi się do nazwy pobieranej strony. Ostatni element — *%parametry* — umożliwia ewentualne przekazanie parametrów strony.

Na przykład adres URL

`http://helion.pl/ksiazki/  
↳administracja-sieci-tcp-ip-dla-kazdego-brian-komar,tcpidk.htm`

wyznacza protokół *http*, komputer o nazwie *helion.pl* oraz plik *ksiazki/administracja-sieci-tcp-ip-dla-kazdego-brian-komar,tcpidk.htm*.

Zazwyczaj użytkownicy przeglądarek pomijają niektóre elementy, skracając zapis do postaci:

helion.pl

W powyższym adresie pominięto specyfikator protokołu (domyślnie *http*), numer portu (domyślnie 80), nazwę dokumentu (domyślnie *index.html*) oraz parametry (domyślnie nie są przekazywane żadne parametry).

Adres URL zawiera wszystkie informacje, które są potrzebne przeglądarce do pobrania strony. Na podstawie umiejscowienia znaków dwukropka, ukośnika i znaku zapytania przeglądarka dzieli adres URL na cztery elementy: specyfikator protokołu, nazwę komputera, nazwę dokumentu oraz parametry. Nazwa komputera i numer portu służą jej do ustanowienia połączenia z serwerem, na którym udostępniono daną stronę. Nazwa dokumentu i jego parametru umożliwiają z kolei wskazanie określonej strony do pobrania.

Odsyłacz do określonego dokumentu jest w standardzie HTML definiowany za pomocą znacznika *<a>* zawierającego właściwy adres URL. Poniższy przykład przedstawia kod źródłowy dokumentu HTML, w którym odsyłaczem jest tekst „Wydawnictwo Helion”:

```
Ta książka została wydana przez .  
<a href="http://www.helion.pl">  
Wydawnictwo Helion</a>, jedno z  
największych wydawnictw książek informatycznych.
```

Odsyłacz zawiera adres URL *http://www.helion.pl*. Przekazanie kodu do przeglądarki skutkuje natomiast wyświetleniem następującego tekstu:

Ta książka została wydana przez Wydawnictwo Helion, jedno z największych wydawnictw książek informatycznych.

## 4.7. Dostarczanie dokumentów za pomocą protokołu HTTP

Protokół transferu dokumentów hipertekstowych (HTTP) jest podstawowym protokołem komunikacyjnym wykorzystywanym przez przeglądarki w interakcjach z serwerami WWW. Odnosząc się do modelu klient-serwer, należy stwierdzić, że przeglądarka jest w tym przypadku klientem, który kontaktuje się z serwerem dzięki nazwie tegoż serwera, zapisanej w adresie URL. Większość adresów URL zawiera bezpośrednie odniesienie do protokołu HTTP (w sekcji *http://*). Niemniej pominięcie specyfikatora protokołu powoduje automatyczne przyjęcie, że odwołanie dotyczy *http*.

Oto kilka cech protokołu HTTP:

- Bazuje na tekstowych komunikatach sterujących.
- Umożliwia przekazywanie binarnych plików z danymi.
- Umożliwia pobieranie danych z serwera i przesyłanie danych do serwera.
- Uwzględnia mechanizmy buforowania danych.

Po ustanowieniu połączenia przeglądarka wysyła do serwera **żądanie** HTTP. Cztery podstawowe rodzaje żądań HTTP zostały przedstawione w tabeli 4.3.

**Tabela 4.3.** Cztery najważniejsze żądania HTTP

Żądanie	Opis
GET	Żądanie dostarczenia dokumentu. Serwer odpowiada, odsyłając kod statusu oraz kopię samego dokumentu.
HEAD	Żądanie dostarczenia informacji statusowych. Serwer odsyła kod statusu, ale bez dołączania kopii dokumentu.
POST	Przesłanie danych do serwera. Serwer dodaje przesłane dane do wskazanego elementu (na przykład wpis do listy komunikatów).
PUT	Przesłanie danych do serwera. Serwer wykorzystuje dostarczone informacje do zastąpienia określonego elementu (tj. nadpisuje wcześniejsze dane).

Najczęściej wykorzystywana formą interakcji jest żądanie dostarczenia strony do przeglądarki. W ramach ustanowionego wcześniej połączenia przeglądarka wysyła żądanie GET, na które serwer odpowiada, odsyłając nagłówek, pusty wiersz oraz treść wskazanego dokumentu. Zgodnie ze standardem HTTP żądanie oraz nagłówek mają format informacji tekstowej. Składnia żądania GET jest następująca:

```
GET /dokument wersja CRLF
```

Ciąg dokument wskazuje plik do pobrania z serwera (którego nazwa jest zawarta w adresie URL). Element wersja odpowiada wersji protokołu (zazwyczaj HTTP/1.0 lub HTTP/1.1). Natomiast człon CRLF symbolizuje dwa znaki ASCII — powroto na początek wiersza (ang. *carriage return*) oraz zmiany wiersza (ang. *linefeed*) — wyznaczające koniec wiersza tekstowego.

Informacja o wersji jest bardzo ważna, ponieważ pozwala na zachowanie zgodności z wcześniejszymi implementacjami protokołu HTTP. Na przykład gdy przeglądarka obsługująca starszą wersję protokołu komunikuje się z nowszym serwerem, serwer przełącza się na starszą wersję protokołu i dostosowuje odpowiednio generowane odpowiedzi. Podsumowując:

*Korzystając z protokołu HTTP, przeglądarki dostarczają informację na temat obsługiwanej wersji protokołu, dzięki czemu serwer może wybrać najwyższą wersję zaimplementowaną po obydwu stronach.*

Pierwszy wiersz odpowiedzi zawiera kod statusu, który informuje przeglądarkę o tym, czy serwer przetworzył żądanie. Jeśli żądanie zostało błędnie sformatowane lub żądany element nie jest dostępny, kod statusu wskaże zainstniały problem. Na przykład jeśli wymieniony w żądaniu plik nie jest dostępny, serwer odeśle kod o wartości 404 (wartość ta jest zdefiniowana w standardzie HTTP). W przypadku odebrania poprawnego żądania serwer odsyła kod statusowy 200. Pozostałe wiersze nagłówka zawierają dodatkowe informacje na temat pobieranego elementu (rozmiar, czas ostatniej modyfikacji oraz typ zawartości). Ogólny format nagłówka odpowiedzi został przedstawiony na rysunku 4.3.

```
HTTP/1.0 kod_statusu tekst_statusu CRLF
Server: identyfikator_serwera CRLF
Last-Modified: data_modyfikacji_dokumentu CRLF
Content-Length: rozmiar_danych CRLF
Content-Type: typ_danych CRLF
CRLF
```

Rysunek 4.3. Ogólny format nagłówka odpowiedzi

Pole `kod_statusu` jest przeznaczone na wartość liczbową zapisaną w formie tekstuowej. Z kolei `tekst_statusu` to odpowiadające tej wartości tekstowe wyjaśnienie znaczenia kodu. W tabeli 4.4 zamieszczony został wykaz najczęściej przesyłanych kodów i ich wersji tekstowych wraz z wyjaśnieniem. Element `identyfikator_serwera` zawiera tekstowy opis serwera (łatwy do zapamiętania przez człowieka), zazwyczaj odpowiadający jego nazwie domenowej. Wartość `rozmiar_danych` w nagłówku `Content-Length` określa rozmiar zasadniczej treści przekazywanego dokumentu liczony w bajtach. Pole `typ_danych` przechowuje wartość tekstową stanowiącą dla przeglądarki informację o tym, jakiego rodzaju treść jest przekazywana. Wartość pola składa się z dwóch elementów rozdzielonych znakiem ukośnika — typu dokumentu oraz jego reprezentacji. Na przykład podczas przekazywania strony HTML pole to zawiera ciąg `text/html`. Natomiast w przypadku przesyłania pliku jpeg typ dokumentu jest określany jako `image/jpeg`.

Tabela 4.4. Przykłady kodów statusowych protokołu HTTP

Kod statusu	Tekst statusu	Opis
200	OK	Poprawne żądanie
400	Bad Request	Błędne żądanie
404	Not Found	Brak dokumentu

Na rysunku 4.4 przedstawiono przykładową odpowiedź serwera WWW Apache. Żądanym dokumentem był plik tekstowy składający się z 24 znaków (zliczane są znaki *To jest strona testowa.* oraz znak nowego wiersza). Mimo że żądanie zostało dostarczone w wersji HTTP 1.0, odpowiedź serwera została wygenerowana zgodnie z protokołem HTTP 1.1, ponieważ taka wersja jest obsługiwana przez serwer. Wynikiem jest dziewięć wierszy nagłówkowych, jeden pusty wiersz oraz zasadnicza treść pliku.

```
HTTP/1.1 200 OK
Date: Sat, 15 Mar 2008 07:35:25 GMT
Server: Apache/1.3.37 (Unix)
Last-Modified: Tue, 1 Jan 2008 12:03:37 GMT
ETag: "78595-81-3883bbe9"
Accept-Ranges: bytes
Content-Length: 24
Connection: close
Content-Type: text/plain
To jest strona testowa.
```

Rysunek 4.4. Przykład odpowiedzi HTTP wygenerowanej przez serwer WWW Apache

## 4.8. Buforowanie stron w przeglądarkach

Buforowanie danych jest istotnym elementem optymalizacji ruchu sieciowego, gdyż użytkownicy często odwiedzają te same strony. Duża część dokumentów składa się z obrazów GIF lub JPEG stanowiących elementy tła lub banerów, które nie zmieniają się zbyt często. Idea buforowania bazuje na następującym założeniu:

*Przeglądarka może istotnie zmniejszyć czas pobierania dokumentu przez zapisanie kopii każdego pliku graficznego w pamięci podręcznej (na dysku lokalnym użytkownika) i wykorzystanie jej podczas prezentacji strony.*

Co się jednak stanie, jeśli dokument przechowywany na serwerze WWW zostanie zmieniony po zbuforowaniu go w pamięci podręcznej przeglądarki? W jaki sposób przeglądarka może stwierdzić, czy zarejestrowana kopia jest przestarzała? Pewna wskazówka jest widoczna na rysunku 4.4 — nagłówek `Last-Modified`. Za każdym razem, gdy przeglądarka pobiera dokument z serwera WWW, otrzymuje również informację o dacie ostatniej modyfikacji pliku. Wartość pola `Last-Modification` jest przechowywana wraz z treścią dokumentu. Dzięki temu przed wykorzystaniem treści zbuforowanej w pamięci podręcznej przeglądarka może przesyłać do serwera żądanie `HEAD` i porównać wartość pola `Last-Modification` z analogiczną wartością pliku zapisanego lokalnie. Jeśli zarejestrowana uprzednio treść uległa przedawnieniu, przeglądarka pobiera nowy dokument. Mechanizm ten opisuje algorytm 4.1.

**Algorytm 4.1.** Skrócenie czasu pobierania dokumentów dzięki buforowaniu ich zawartości

Dane:

Adres URL elementu strony internetowej

Wynik:

Kopia strony

Realizacja:

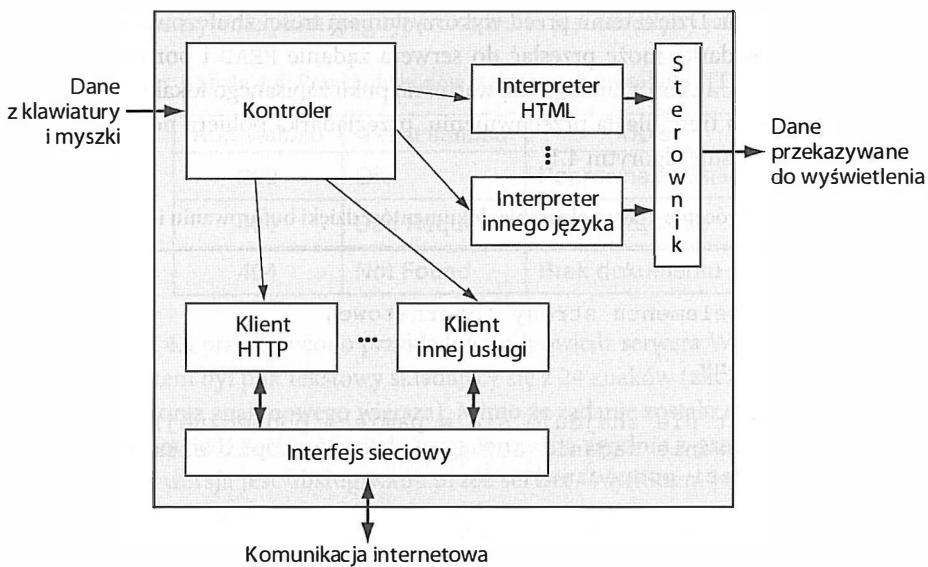
```
if (element nie znajduje się w pamięci podręcznej) {  
    Wygenerowanie żądania GET i zapisanie kopii elementu  
    ↳w pamięci podręcznej;  
} else {  
    Wygenerowanie żądania HEAD;  
    if (zbuforowany element jest aktualny) {  
        wykorzystanie zbuforowanego elementu;  
    } else {  
        Wygenerowanie żądania GET i zapisanie kopii elementu  
        ↳w pamięci podręcznej;  
    }  
}
```

W przedstawionym algorytmie pominięto kilka mniej istotnych szczegółów, na przykład to, że protokół http umożliwia serwerom dołączanie nagłówków `No-cache`, które zabraniają buforowania określonych treści. Ponadto przeglądarki nie zapisują w pamięci

podręcznej elementów o niewielkim rozmiarze, ponieważ czas potrzebny na pobranie ich za pomocą żądania GET jest porównywalny z czasem realizacji żądania HEAD, a przechowywanie dużej liczby niewielkich plików wydłuża czas dostępu do pamięci podręcznej.

## 4.9. Budowa przeglądarki

Przeglądarka jest skomplikowanym programem. Przede wszystkim dlatego, że realizuje wiele usług i udostępnia graficzny interfejs użytkownika. Oczywiście, każda przeglądarka musi mieć zaimplementowaną obsługę protokołu HTTP, ale większość z nich pozwala na korzystanie również z innych protokołów. Z uwagi na fakt, że adres URL obejmuje pole specyfikatora protokołu, przeglądarka musi zawierać kod kliencki odpowiadający wszystkim wykorzystywanym wartościom tego typu. Jej twórcy muszą więc implementować mechanizmy zapewniające poprawną interakcję z każdym rodzajem serwera oraz algorytmy właściwej interpretacji odbieranych odpowiedzi. Na przykład przeglądarka musi zawierać kod korzystania z usługi FTP (opisanej w kolejnym punkcie). Poszczególne komponenty składowe tego rodzaju oprogramowania zostały przedstawione na rysunku 4.5.



Rysunek 4.5. Budowa przeglądarki korzystającej z wielu usług

## 4.10. Protokół transferu plików (FTP)

Plik jest podstawową jednostką składowania danych, ponieważ pliki mogą przechowywać obiekty dowolnego formatu (na przykład dokumenty tekstowe, arkusze kalkulacyjne, programy komputerowe, obrazy graficzne, dane itp.), a funkcja przesyłania kopii pliku z jednego komputera do drugiego okazuje się bardzo użytecznym mechanizmem wymiany danych. Zadania tego typu często określa się skrótnie **transferem plików**.

Transfer plików w internecie jest skomplikowaną operacją, ponieważ komputery są jednostkami heterogenicznymi. Wynika to z tego, że każdy system operacyjny w inny sposób reprezentuje pliki, operuje innymi typami plików i wykorzystuje różne mechanizmy dostępu do nich. W niektórych systemach do opisu obrazów JPEG stosuje się rozszerzenie *.jpg*, a w innych *.jpeg*. W pewnych systemach zakończeniem wiersza tekstowego jest pojedynczy znak przejścia do kolejnego wiersza. W innych do tego celu są wykorzystywane dwa znaki — powrotu na początek wiersza i przejścia do kolejnego wiersza. W niektórych systemach separatorem katalogów w ścieżce dostępu do pliku jest znak ukośnika (/), a w innych jest to znak odwrotnego ukośnika (\). Ponadto system operacyjny może przypisywać do poszczególnych plików konta użytkowników o określonych prawach do operowania tymi plikami. Wspomniane konta nie są jednak różne w różnych komputerach. Zatem użytkownik X w jednym systemie nie odpowiada użytkownikowi X w drugim systemie.

Najczęściej wykorzystywaną usługą transferu plików w internecie jest **protokół transferu plików** (FTP — *File Transfer Protocol*). Oto kilka cech usługi FTP:

- **Dowolna zawartość plików.** Protokół FTP przenosi dane każdego rodzaju, włącznie z dokumentami, obrazami, muzyką i filmami.
- **Dwukierunkowy transfer.** Usługę FTP można wykorzystać do pobierania plików (przesyłania ich z serwera do klienta) lub wysyłania plików (przesyłania z jednostki klienckiej do serwera).
- **Obsługa uwierzytelniania i praw dostępu.** Protokół FTP umożliwia ustalenie właściciela pliku oraz zapewnia weryfikację praw dostępu do plików.
- **Możliwość przeglądania katalogów.** Usługa FTP pozwala użytkownikom na przeglądanie zawartości katalogów (folderów).
- **Tekstowe komunikaty sterujące.** Podobnie jak w przypadku wielu innych usług internetowych wymiana poleceń między klientem i serwerem bazuje na komunikatach tekstowych, zapisanych w formacie ASCII.
- **Uwzględnienie heterogeniczności jednostek.** Protokół FTP jest niezależny od szczegółów implementacyjnych określonego systemu operacyjnego i może przekazywać kopie plików między dowolnymi jednostkami sieciowymi.

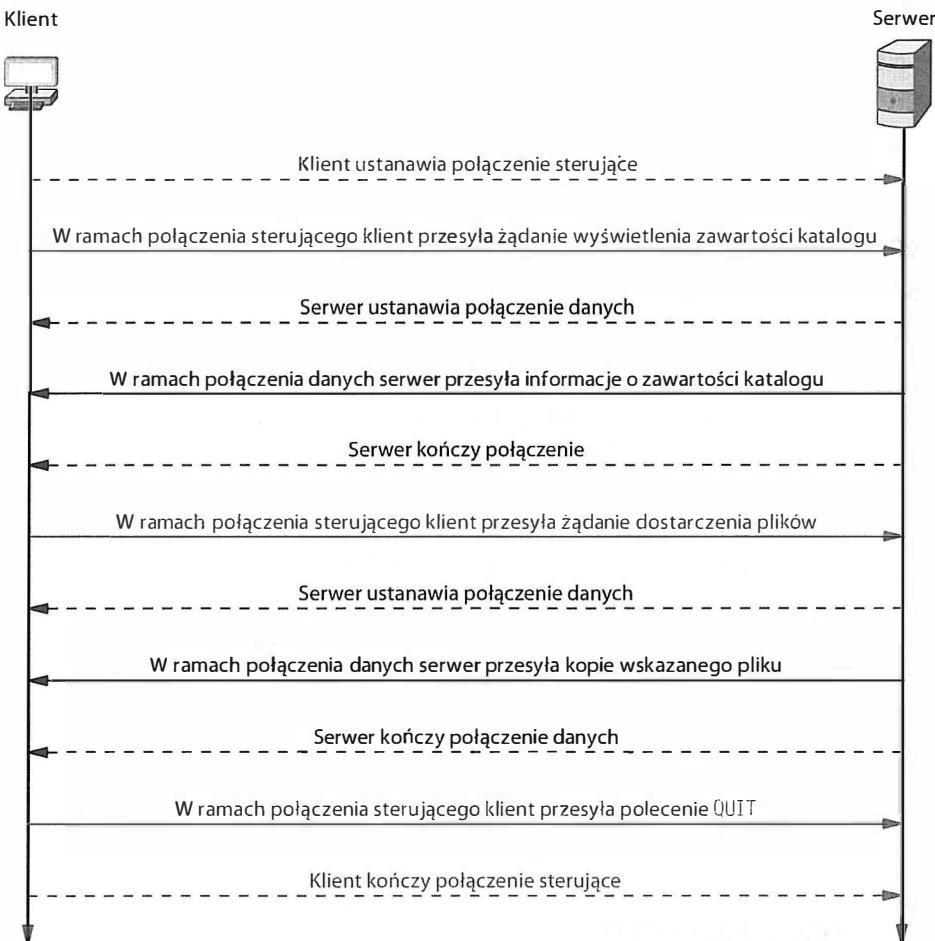
Ponieważ niewiele osób korzysta ze specjalnych aplikacji FTP, wykorzystanie tego protokołu zazwyczaj pozostaje niezauważone przez użytkowników. Klient FTP jest jednak często automatycznie uaktywniany, gdy użytkownik zażąda dostarczenia pliku z serwisu internetowego.

## 4.11. Komunikacja FTP

Jednym z najciekawszych aspektów działania mechanizmu FTP jest sposób wymiany danych między klientem i serwerem. W ogólnym ujęciu zagadnienie wydaje się nieskomplikowane. Klient ustanawia połączenie z serwerem FTP i generuje żądania, na które serwer odpowiada. Jednak w przeciwieństwie do usługi HTTP protokół FTP nie zawsze przesyła

odpowiedzi w tym samym połączeniu, w którym otrzymuje żądania. Za każdym razem, gdy następuje przesyłanie pliku (do serwera lub do klienta), serwer ustanawia nowe połączenie. Aby odróżnić je od połączeń sterujących, połączenia przeznaczone do transmisji plików są nazywane **połączniami danych**.

Co ciekawe, w wymianie informacji w połączeniu danych protokół FTP odwraca rolę jednostek modelu klient-serwer. Oznacza to, że wraz z otwarciem połączenia danych klient rozpoczyna pracę w charakterze serwera (tj. oczekuje na ustanowienie połączenia), a serwer działa jak klient (tj. inicjuje połączenie). Po zakończeniu przesyłania danego pliku połączenie danych jest przerywane. Jeżeli użytkownik życzy sobie dostarczenia kolejnych plików, serwer ustanawia nowe połączenia. Doskonałą ilustracją opisanego mechanizmu jest rysunek 4.6.



Rysunek 4.6. Połączenia w czasie typowej sesji FTP

W zestawieniu pominięto kilka istotnych szczegółów, między innymi to, że po ustanowieniu połączenia sterującego użytkownik musi się zalogować na serwerze. Protokół FTP uwzględnia polecenie USER, które umożliwia wprowadzenie nazwy konta, oraz pole-

cenie PASS służące do przekazania treści hasła. Serwer odsyła do jednostki klienckiej kod statusowy z informacją o tym, czy uwierzytelnienie zakończyło się powodzeniem. Inne polecenia mogą być dostarczane do serwera tylko po poprawnym zalogowaniu się użytkownika<sup>14</sup>.

W przypadku ustanawiania połączenia danych istotny jest również numer portu protokołu. Jaki numer powinien zostać wykorzystany w czasie zestawiania połączenia z klientem? W specyfikacji FTP można znaleźć ciekawą odpowiedź — przed przesaniem żądania do serwera klient wyznacza port lokalnego systemu operacyjnego i przesyła jego numer do serwera. Klient kojarzy w ten sposób port z usługą i rozpoczyna oczekiwanie na połączenie. Informacja o numerze portu jest dostarczana do serwera w ramach połączenia sterującego za pomocą polecenia PORT. Działanie opisanego mechanizmu podsumowuje algorytm 4.2.

**Algorytm 4.2.** Operacje wykonywane po stronie klienta i serwera w trakcie połączenia FTP

Dane:

Połączenie sterujące FTP

Wynik:

Przesłanie danych w ramach połączenia danych usługi FTP

Realizacja:

Za pomocą połączenia sterującego klient przesyła do serwera żądanie dostarczenia określonego pliku.

Serwer odbiera żądanie.

Klient wybiera lokalny port o numerze X.

Klient przypisuje port X do usługi i przygotowuje się do zaakceptowania połączenia.

Klient przesyła polecenie „PORT X” do serwera, wykorzystując połączenie sterujące.

Serwer odbiera polecenie PORT i żądanie dostarczenia pliku.

Klient oczekuje na ustanowienie połączenia danych na porcie X.

Serwer ustanawia połączenie danych z portem X komputera klienckiego.

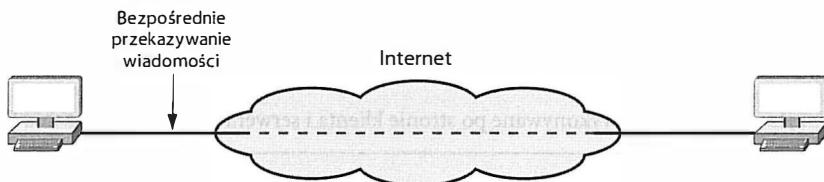
Serwer przesyła żądanego plik w ramach połączenia danych. Serwer zamyka połączenie danych.

Przekazywanie informacji o porcie pomiędzy dwoma aplikacjami wydaje się bezproblemową operacją. W praktyce jednak tak nie jest, a opisana technika zawodzi w niektórych sytuacjach. Ustanawianie połączenia z nowym portem jest niemożliwe, jeśli jedno z urządzeń końcowych jest przesłonięte przez urządzenie dokonujące translacji adresów sieciowych (NAT — *Inetwork Address Translation*), takich jak router dostępowy w sieci domowej. W rozdziale 23. został szczegółowo opisany wyjątkowy sposób obsługi protokołu FTP (urządzenie wykonujące funkcję NAT rozpoznaje połączenie sterujące FTP i analizuje przekazywane informacje, a po zarejestrowaniu polecenia PORT zmienia przesyłaną wartość).

<sup>14</sup> Korzystając z publicznie dostępnych zasobów serwera, klient uwierzytelnia się za pomocą konta anonimowego, używając nazwy *anonymous* oraz hasła *guest*.

## 4.12. Poczta elektroniczna

Mimo wzrostu popularności usług związanych z bezpośrednią komunikacją poczta elektroniczna nadal pozostaje jedną z najczęściej wykorzystywanych aplikacji internetowych. Usługa ta została opracowana przed upowszechnieniem się komputerów osobistych i przed pojawieniem się na rynku urządzeń PDA. Umożliwia ona przekazanie wiadomości utworzonej w jednym komputerze bezpośrednio do użytkownika innej jednostki. Architektura systemu została przedstawiona na rysunku 4.7, a poszczególne etapy komunikacji są opisane w algorytmie 4.3.



**Rysunek 4.7.** Pierwotna konfiguracja systemu poczty elektronicznej, w której zakładano bezpośrednie przekazywanie wiadomości z komputera nadawcy do komputera odbiorcy

**Algorytm 4.3.** Operacje podejmowane w celu dostarczenia poczty zgodnie z pierwotnymi założeniami odnośnie sposobu działania usługi

Dane:

System poczty elektronicznej zapewniający przekazywanie wiadomości między użytkownikami.

Wynik:

Przekazanie wiadomości do wskazanego odbiorcy.

Realizacja:

Użytkownik uruchamia aplikację pocztową i tworzy wiadomość e-mail przeznaczoną dla odbiorcy o adresie x@cel.pl.

Program pocztowy nadawcy zapisuje wiadomość w kolejce nadawczej.

Program dostarczania poczty w komputerze nadawcy przegląda kolejkę wyjściową i odczytuje wiadomość.

Program dostarczania poczty nawiązuje połączenie z jednostką o adresie cel.pl.

Program dostarczania poczty wykorzystuje protokół SMTP do przekazania wiadomości.

Program dostarczania poczty kończy połączenie z jednostką odbiorczą.

Serwer pocztowy pracujący pod adresem cel.pl odbiera wiadomość i umieszcza jej kopię w skrzynce pocztowej użytkownika x.

Użytkownik x uruchamia w systemie cel.pl program pocztowy, który wyświetla zawartość skrzynki pocztowej, a w niej dostarczoną wiadomość.

Jak wynika z analizy algorytmu, nawet wczesne programy pocztowe były podzielone na dwa niezależne komponenty:

- aplikację stanowiącą interfejs użytkownika dla systemu pocztowego,
- program dostarczania poczty.

**Aplikacja interfejsu** jest wywoływana bezpośrednio przez użytkownika. Udostępnia ona funkcje tworzenia i edytowania wychodzących wiadomości, a także odczytu i przetwarzania odbieranych listów elektronicznych. Nie ma jednak na celu działania w charakterze klienta lub serwera i nie przekazuje wiadomości do innych użytkowników. Odczytuje natomiast listy pozostawione w **skrzynce pocztoowej** danego użytkownika (tj. pliki w komputerze użytkownika) i zapisuje nowo utworzone wiadomości w **kolejce poczty wychodzącej** (zazwyczaj w odpowiednim folderze dysku użytkownika). Samym transferem wiadomości zajmowały się inne programy — **program dostarczania poczty** oraz **serwer pocztowy**. Program dostarczania poczty pełnił funkcję klienta komunikującego się z serwerem pocztowym działającym w komputerze docelowym. Zadania serwera pocztowego ograniczają się do odbierania nadchodzących listów i składowania ich w skrzynkach pocztowych odpowiednich użytkowników.

Specyfikację systemu dostarczania poczty za pośrednictwem internetu można podzielić na trzy kategorie, wymienione w tabeli 4.5.

Tabela 4.5. Trzy rodzaje protokołów wykorzystywanych w systemie pocztowym

Typ	Opis
Transfer poczty	Protokół wykorzystywany do przekazywania poczty z jednego komputera do drugiego.
Dostęp do wiadomości	Protokół umożliwiający użytkownikom korzystanie z ich skrzynek pocztowych oraz wysyłanie nowych listów.
Reprezentacja wiadomości	Protokół opisujący format wiadomości pocztowych zapisywanych na dysku komputera.

## 4.13. Prosty protokół dostarczania poczty (SMTP)

**Prosty protokół dostarczania poczty** (SMTP — *Simple Mail Transfer Protocol*) jest standardowym protokołem przesyłania wiadomości e-mail do serwerów pocztowych działających w internecie. Oto kilka cech mechanizmu SMTP:

- Bazuje na komunikacji strumieniowej.
- Wykorzystuje tekstowe komunikaty sterujące.
- Przekazuje jedynie wiadomości tekstowe.
- Pozwala nadawcy na określenie nazw odbiorców i weryfikuje każdą nazwę.
- Przekazuje pojedynczą kopię danej wiadomości.

Najbardziej zaskakującym aspektem funkcjonowania protokołu SMTP jest to, że jest ono ograniczone do przekazywania treści tekstowych. W dalszych punktach zostanie omówiony

standard MIME, umożliwiający dołączanie do listów elektronicznych załączników (takich jak pliki graficzne lub pliki binarne), niemniej standardowy mechanizm SMTP jest przy stosowany do przekazywania tekstu.

Drugą szczególną cechą jest możliwość przesyłania pojedynczej wiadomości do wielu użytkowników danego komputera. Protokół pozwala bowiem nadawcy na zdefiniowanie listy odbiorców i przekazanie jednej kopii wiadomości do wszystkich osób występujących na liście. Klient dostarcza do serwera komunikat „mam wiadomość dla użytkownika A”, a serwer odsyła odpowiedź „OK” albo „nie ma takiego użytkownika”. W praktyce komunikatom opisowym towarzyszą kody numeryczne. Odpowiedzi mają więc treść typu 250 OK lub 550 No such user here (nie ma takiego użytkownika). Przykładowa sesja SMTP została przedstawiona na rysunku 4.8. Odnosi się ona do przypadku przesyłania wiadomości e-mail od użytkownika Jan\_Kowalski dysponującego kontem w komputerze przyklad.pl do dwóch użytkowników, których konta są założone w systemie gdzies.pl.

```

Serwer: 220 gdzies.pl Simple Mail Transfer Service Ready
Klient: HELO przyklad.pl
Serwer: 250 OK

Klient: MAIL FROM:<Jan_Kowalski@przyklad.pl>
Serwer: 250 OK

Klient: RCPT TO:<Zenon_Nowak@gdzies.pl>
Serwer: 550 No such user here

Klient: RCPT TO:<Stefan_Iksinski@gdzies.pl>
Serwer: 250 OK

Klient: DATA
Serwer: 354 Start mail input; end with <CR><LF>.<CR><LF>
Klient: ...treść wiadomości składającej się z dowolnej
Klient: ...liczby wierszy
Klient: <CR><LF>.<CR><LF>
Serwer: 250 OK

Klient: QUIT
Serwer: 221 gdzies.pl closing transmission channel

```

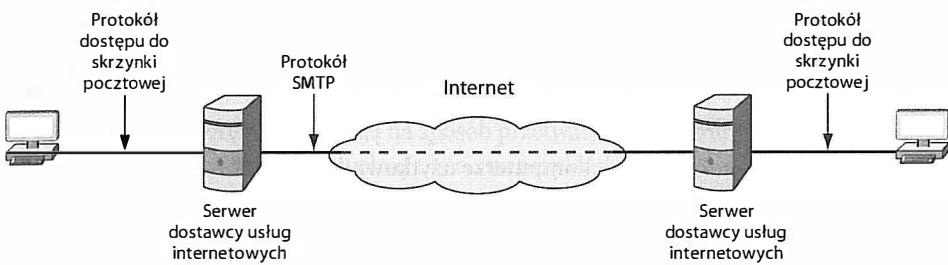
Rysunek 4.8. Przykładowa sesja SMTP

Każdy wiersz przykładu jest oznaczony etykietą *Klient:* lub *Serwer:* w celu wskazania jednostki, która generuje dany wiersz. Sam protokół, oczywiście, nie dodaje tego typu oznaczeń. Polecenie HELO umożliwia klientowi uwierzytelnienie się z podaniem własnej nazwy domenowej. Zapis <CR><LF> odpowiada znakowi powrotu na początek wiersza, po którym następuje znak przejścia do następnego wiersza (czyli symbolowi końca wiersza). Treść wiadomości e-mail jest zakończona wierszem składającym się tylko z pojedyńczej kropki (bez jakiegokolwiek dodatkowego tekstu lub znaków spacji).

Słowo *prosty* w pełnej nazwie protokołu SMTP oznacza, że stosowane obecnie rozwiązanie zostało niegdyś uproszczone. Mechanizm poprzedzający standard SMTP rzeczywiście był niezwykle skomplikowany. Dlatego twórcy nowego rozwiązania usunęli większość niepotrzebnych funkcji i skoncentrowali swoje działania na rzeczach najważniejszych.

#### 4.14. Dostawcy usług internetowych, serwery pocztowe i dostęp do poczty elektronicznej

Wraz z rozwojem internetu i udostępnieniem go zwykłym odbiorcom sposób przesyłania wiadomości e-mail uległ istotnej zmianie. Większość użytkowników komputerów nie wiedziała, w jaki sposób należy skonfigurować serwer pocztowy oraz jak można zarządzać jego pracą. Dostawcy usług internetowych zaczęli więc oferować usługi pocztowe swoim abonentom. Rozwiązanie polegało na utrzymywaniu serwerów pocztowych, na których założone były skrzynki pocztowe klientów firmy. Zamiast tradycyjnego oprogramowania pocztowego każdy dostawca zapewniał interfejs dostępowy, pozwalający użytkownikom na zarządzanie własną skrzynką. Konfigurację opisywanego systemu zaprezentowano na rysunku 4.9.



Rysunek 4.9. Konfiguracja systemu pocztowego, w którym dostawca usług internetowych oferuje dostęp do serwera pocztowego i zarządza kontami użytkowników końcowych

Dostęp do poczty elektronicznej gwarantują dwa rozwiązania:

- wykorzystanie specjalnych aplikacji pocztowych,
- użycie przeglądarki internetowej odwołującej się do serwisu WWW systemu pocztowego.

Rozwiązanie bazujące na przeglądarce internetowej jest mniej skomplikowane. Bazuje ono na założeniu, że dostawca usług internetowych zapewnia dostęp do specjalnego serwisu WWW, w którym są prezentowane wiadomości pochodzące ze skrzynek pocztowych użytkowników. Zadanie użytkownika sprowadza się więc jedynie do uruchomienia przeglądarki i wpisania adresu dostawcy. Serwis internetowy wymaga podania loginu i hasła, na podstawie których identyfikowana jest właściwa skrzynka pocztowa. Serwer WWW pobiera wiadomości ze skrzynki i wyświetla na stronie internetowej. Główną zaletą korzystania ze stron internetowych do obsługi poczty jest możliwość odczytywania wiadomości z dowolnego komputera. Użytkownik nie musi bowiem uruchamiać specjalnego programu pocztowego.

Zaletą stosowania dedykowanych aplikacji pocztowych jest z kolei możliwość pobierania całej zawartości skrzynki poczтовej na dysk lokalny komputera. Jest to szczególnie wygodne dla osób, które często się przemieszczają i korzystają z laptopów. Po podłączeniu laptopa do sieci użytkownik może uruchomić program pocztowy i pobrać zawartość skrzynki do swojego komputera. Dzięki temu przetwarzanie wiadomości można przeprowadzić po odłączeniu jednostki od sieci (na przykład w samolocie). Przywrócenie połączenia internetowego powoduje, że zainstalowane w laptopie oprogramowanie komunikuje się z serwerem dostawcy usług internetowych w celu przesłania do niego listów utworzonych w międzyczasie przez użytkownika oraz pobrania nowych wiadomości, które zostały zapisane w skrzynce poczтовej danej osoby.

#### **4.15. Protokoły dostępu do poczty (POP, IMAP)**

**Dostęp** do poczty zapewniają odpowiednie protokoły. Protokół dostępowy jest niezależny od protokołu dostarczania poczty, ponieważ jego zadanie ogranicza się do obsługi interakcji pojedynczego użytkownika z jedną skrzynką pocztową. Protokoły dostarczania poczty umożliwiają natomiast użytkownikom wysyłanie wiadomości do innych odbiorców. Oto cechy protokołu dostępowego:

- Zapewnia dostęp do skrzynki poczowej użytkownika.
- Umożliwia użytkownikowi przeglądanie nagłówków wiadomości, a także ich pobieranie i usuwanie.
- Program kliencki działa w komputerze użytkownika.
- Program serwerowy pracuje w komputerze, w którym jest utrzymywana skrzynka pocztowa użytkownika.

W przypadku niezbyt wydajnego łącza między komputerem klienckim a serwerem pocztowym bardzo użyteczna okazuje się funkcja przeglądania wiadomości bez konieczności pobierania ich treści. Na przykład użytkownik posługujący się telefonem komórkowym może przejrzeć nagłówki listów, a następnie usunąć niechcianą pocztę bez oczekiwania na pobranie całej zawartości skrzynki poczowej.

Dostęp do poczty zapewnia wiele różnych mechanizmów. Niektórzy dostawcy usług internetowych oferują klientom własne (darmowe) programy pocztowe. Ponadto obowiązują dwa standardowe protokoły dostępu do serwerów e-mail. Nazwy tych protokołów zostały wymienione w tabeli 4.6.

**Tabela 4.6. Dwa standardowe protokoły dostępu do skrzynek pocztowych**

Skrót	Nazwa
POP3	Post Office Protocol (protokół pocztowy) wersja 3
IMAP	Internet Mail Access Protocol (internetowy protokół dostępu do poczty)

Mimo że w założeniu realizują te same zadania, ich działanie różni się w wielu szczegółach. Zastosowano w nich na przykład różne mechanizmy uwierzytelniania, za których pomocą użytkownik identyfikuje się po stronie serwera (dzięki uwierzytelnieniu dana osoba nie ma dostępu do skrzynek pocztowych innych użytkowników serwera).

## 4.16. Standardy zapisu wiadomości e-mail (RFC2822, MIME)

Format wiadomości e-mail regulują dwa standardy:

- RFC2822 — standard formatu wiadomości pocztowych;
- wielozadaniowe rozszerzenia poczty internetowej (MIME — ang. *Multi-purpose Internet Mail Extensions*).

**Standard formatu wiadomości pocztowych RFC2822.** Standard ten jest opisany w dokumencie IETF nazywanym **prośba o komentarze 2822** (RFC — *Request for Comments*). Sam format wiadomości wydaje się nieskomplikowany. Zgodnie z nim list odpowiada pojedynczemu plikowi tekstowemu, na który składa się **nagłówek**, pusty wiersz oraz **treść**. Wiersze nagłówka mają format:

Słowo-kluczowe: informacja

Przykładami słów kluczowych są **From:** (od), **To:** (do), **Subject:** (temat), **Cc:** (do wiadomości) itp. Ponadto możliwe jest dołączanie wierszy nagłówka rozpoczynających się od dużej litery X, które nie wpływają na sposób przetwarzania wiadomości. Nagłówek listu może więc zawierać wiersz o dowolnej treści, na przykład:

X-Najgorsze-Programy-TV: wszystkie reality show

**Wielozadaniowe rozszerzenia poczty internetowej (MIME).** Zgodnie z wcześniejszymi informacjami protokół SMTP obsługuje jedynie wiadomości tekstowe. Standard MIME rozszerza jednak funkcjonalność systemu poczty w taki sposób, aby możliwe było przekazywanie również danych innych niż tekst. Specyfikacja MIME definiuje zasady kodowania plików binarnych za pomocą znaków drukowanych, które następnie są dołączone do komunikatu i rozkodowywane po stronie odbiorczej.

Choć wraz ze stosowaniem rozszerzeń MIME upowszechnił się standard kodowania **Base64**, sama specyfikacja nie narzuca żadnych określonych formatów kodowania. Pozwala natomiast nadawcy i odbiorcy na wybranie mechanizmu kodowania najlepszego w danej sytuacji. W celu określenia zastosowanego kodowania nadawca musi dołączyć do nagłówka wiadomości odpowiedni wiersz informacyjny. Ponadto standard MIME umożliwia nadawcy podzielenie wiadomości na kilka części i zastosowanie niezależnych mechanizmów kodowania w każdej z nich. Dzięki temu użytkownik może przesyłać czysty tekst z dołączonymi do niego plikami graficznymi, arkuszami kalkulacyjnymi, utworami muzycznymi itp. Każdy z wymienionych elementów jest wówczas kodowany niezależnie od pozostałych. Jednostka odbiorcza może natomiast wybrać właściwy sposób przetwarzania każdego z załączników (na przykład zapis na dysku lub wyświetlenie na ekranie).

W praktyce zastosowanie mechanizmu MIME wiąże się z dodaniem dwóch wierszy do nagłówka wiadomości. Jeden informuje o wykorzystaniu standardu MIME, a drugi opisuje sposób dołączenia informacji w formacie MIME do treści wiadomości. Oto przykładowe wiersze nagłówka:

```
MIME-Version: 1.0  
Content-Type: Multipart/Mixed; Boundary=Separator_MIME
```

Informuję one o tym, że wiadomość została utworzona zgodnie z wersją 1.0 specyfikacji MIME oraz że przed każdym niezależnym elementem listu będzie występował ciąg separatora o treści Separator\_MIME. W przypadku wykorzystania mechanizmu MIME do przesyłania klasycznej wiadomości tekstowej drugi wiersz ma treść:

```
Content-Type: text/plain
```

Standard MIME gwarantuje zachowanie zgodności z systemami pocztowymi, które nie obsługują tego rodzaju rozszerzeń lub podanego kodowania. Choć, oczywiście, nie ma wówczas możliwości wyodrębnienia z treści listu binarnych załączników — cała treść wiadomości jest traktowana jak pojedynczy blok tekstu. Podsumowując:

*Standard MIME zakłada dodanie nadmiarowych wierszy nagłówka, które umożliwiają przesyłanie wraz z wiadomością nietekstowych załączników. Załączniki są kodowane za pomocą liter drukowanych, a każdy z nich jest poprzedzany specjalnym wierszem separatora.*

## 4.17. System nazw domenowych (DNS)

**System nazw domenowych** (DNS — ang. *Domain Name System*) odpowiada za odzyskiwanie nazw symbolicznych (łatwy do zapamiętania dla ludzi) na adresy komputerów. Przeglądarki, programy pocztowe oraz większość aplikacji internetowych wykorzystują system DNS w bieżącym działaniu. Rozwiążanie to stanowi jednocześnie ciekawy przykład interakcji klient-serwer, ponieważ odwzorowanie adresów nie jest realizowane przez pojedynczy serwer. Informacje na temat nazw jednostek są rozproszone na wielu serwerach rozmieszczeniach w różnych obszarach internetu. Za każdym razem, gdy aplikacja musi przetłumaczyć nazwę na adres komputera, pełni funkcję klienta systemu DNS. Klient wysyła żądanie do serwera nazw, który wyszukuje właściwy adres i odsyła komunikat z odpowiedzią. Jeśli ustalenie odpowiedzi nie jest możliwe, dany serwer staje się klientem innego serwera nazw. Operacja się powtarza aż do odnalezienia serwera, który może udzielić odpowiedzi.

Każda nazwa komputera składa się z sekwencji segmentów alfanumerycznych rozdzielonych znakami kropki. Na przykład komputer Wydziału Elektroniki i Telekomunikacji Politechniki Poznańskiej dysponuje nazwą domenową:

et.put.poznan.pl

Z kolei komputer pracujący w Cisco, Incorporated jest dostępny pod nazwą:

anakin.cisco.com

System nazw domenowych ma strukturę hierarchiczną. W każdej z nazw najbardziej znaczący człon znajduje się na jej końcu. Segment występujący po lewej stronie nazwy (et lub anakin w podanych przykładach) odpowiada nazwie konkretnego komputera. Pozostałe człony nazwy domenowej odnoszą się do grupy, w której skład wchodzi nazwa danego komputera. Na przykład `put` odpowiada nazwie politechniki, a `cisco` jest nazwą firmy. Standard DNS nie ogranicza liczby segmentów nazwy. Każda z organizacji we własnym zakresie ustala, ile segmentów jest wykorzystywanych do opisu komputerów w sieci wewnętrznej oraz co dane segmenty oznaczają.

System nazw domenowych wyznacza wartości jedynie dla najbardziej znaczącego segmentu nazwy, nazywanego **domeną najwyższego poziomu** (TLD — ang. *Top-level domain*). Zarządzaniem domenami najwyższego poziomu zajmuje się agencja Internet Corporation for Assigned Names and Numbers (ICANN). To ona wyznacza agencje **rejestrujące domeny**, które administrują konkretnymi domenami najwyższego poziomu i zatwierdzają konkretne nazwy. Część domen najwyższego poziomu jest ogólnie dostępna. Pozostałe są przeznaczone dla określonych grup podmiotów lub agencji rządowych. Wykaz domen najwyższego poziomu został zamieszczony w tabeli 4.7.

Tabela 4.7. Przykłady domen najwyższego poziomu wraz z opisem grupy użytkowników

Nazwa domeny	Przeznaczenie
<code>aero</code>	Organizacje transportu powietrznego
<code>arpa</code>	Domena odwzorowania odwrotnego
<code>asia</code>	Organizacje azjatyckie lub związane z Azją
<code>biz</code>	Biznes
<code>com</code>	Organizacje komercyjne
<code>coop</code>	Organizacje zajmujące się współpracą
<code>edu</code>	Instytucje edukacyjne
<code>gov</code>	Instytucje rządowe
<code>info</code>	Informacja
<code>int</code>	Organizacje międzynarodowe
<code>jobs</code>	Zarządzanie zasobami ludzkimi
<code>mil</code>	Wojsko
<code>mobi</code>	Dostawcy treści dla urządzeń mobilnych
<code>museum</code>	Muzea
<code>name</code>	Nazwy indywidualne
<code>net</code>	Duże ośrodki sieciowe
<code>org</code>	Organizacje niekomercyjne
<code>pro</code>	Osoby o uznany dorobku zawodowym
<code>travel</code>	Turystyka
<code>kod kraju</code>	Niepodległe kraje

Organizacje występują o przydzielenie nazwy w ramach jednej z domen najwyższego poziomu. Na przykład wiele firm stara się o uzyskanie nazwy w domenie *com*. Przedsiębiorstwo o nazwie *Superex* mogłoby wystąpić o zarejestrowanie domeny *superex* w domenie najwyższego poziomu *com*. Po pozytywnym rozpatrzeniu wniosku firma Superex otrzymałaby domenę:

*superex.com*

Gdyby okazało się, że wspomniana domena została wcześniej przydzielona innej organizacji, firma mogłaby wystąpić o przydział domeny *superex.biz* lub *superex.org* (ale nie *superex.com*). Ponadto po uzyskaniu prawa do domeny *superex.com* przedsiębiorstwo mogłoby samo zdecydować, ile ustanowi dodatkowych poziomów podziału domeny oraz jakie będzie znaczenie poszczególnych nazw. Na przykład gdyby oddziały firmy znajdowały się we wschodniej i zachodniej Polsce, nazwy komputerów mogłyby mieć następującą postać:

*komputer1.zachod.superex.com*

Oczywiście, nic nie stałoby na przeszkodzie, aby opracować plan nazewnictwa pozbawionego hierarchii i identyfikować komputery jedynie za pomocą nazwy jednostki i domeny przedsiębiorstwa:

*komputer1.superex.com*

Poza podziałem na domeny funkcjonalne system DNS umożliwia również dobieranie domen zależnie od regionu geograficznego. Na przykład Urząd Miasta Bydgoszcz dysponuje domeną:

*um.bydgoszcz.pl*

Ponieważ urząd znajduje się w mieście Bydgoszcz w Polsce, nazwy komputerów należących do urzędu mają końcówkę *.bydgoszcz.pl* zamiast *.com*.

Często również stosuje się połączenie nazw domen geograficznych i funkcjonalnych. Wiele szkół w Polsce posługuje się domenami:

*edu.pl*

w których człon *edu* oznacza instytucję edukacyjną, a *.pl* pochodzi, oczywiście, od kodu kraju.

#### **4.18. Nazwy domenowe rozpoczynające się od www**

Wiele organizacji przypisuje komputerom nazwy domenowe odzwierciedlające usługę wykonywaną przez tę jednostkę. Na przykład komputer realizujący zadanie serwera FTP prawdopodobnie nazywałby się:

*ftp.firma.pl*

Analogicznie, jednostka funkcjonująca jako serwer WWW zapewne otrzymałaby nazwę:

*www.firma.pl*

Takie nazwy są łatwe do zapamiętania, ale nie są obowiązkowe. Użycie członu *www* w nazwie komputera wykonującego zadanie serwera WWW jest jedynie konwencją.

Usługę WWW można udostępnić w dowolnym systemie, nawet jeśli jego nazwa domenowa nie zawiera sekcji *www*. Ponadto komputer, którego nazwa domenowa obejmuje człon *www*, nie musi wcale pełnić funkcji serwera WWW.

*Pierwszy element nazwy domenowej nie definiuje usługi realizowanej przez dany komputer (np. www), jest on jedynie zgodny z konwencją, która ułatwia użytkownikom zapamiętywanie nazw.*

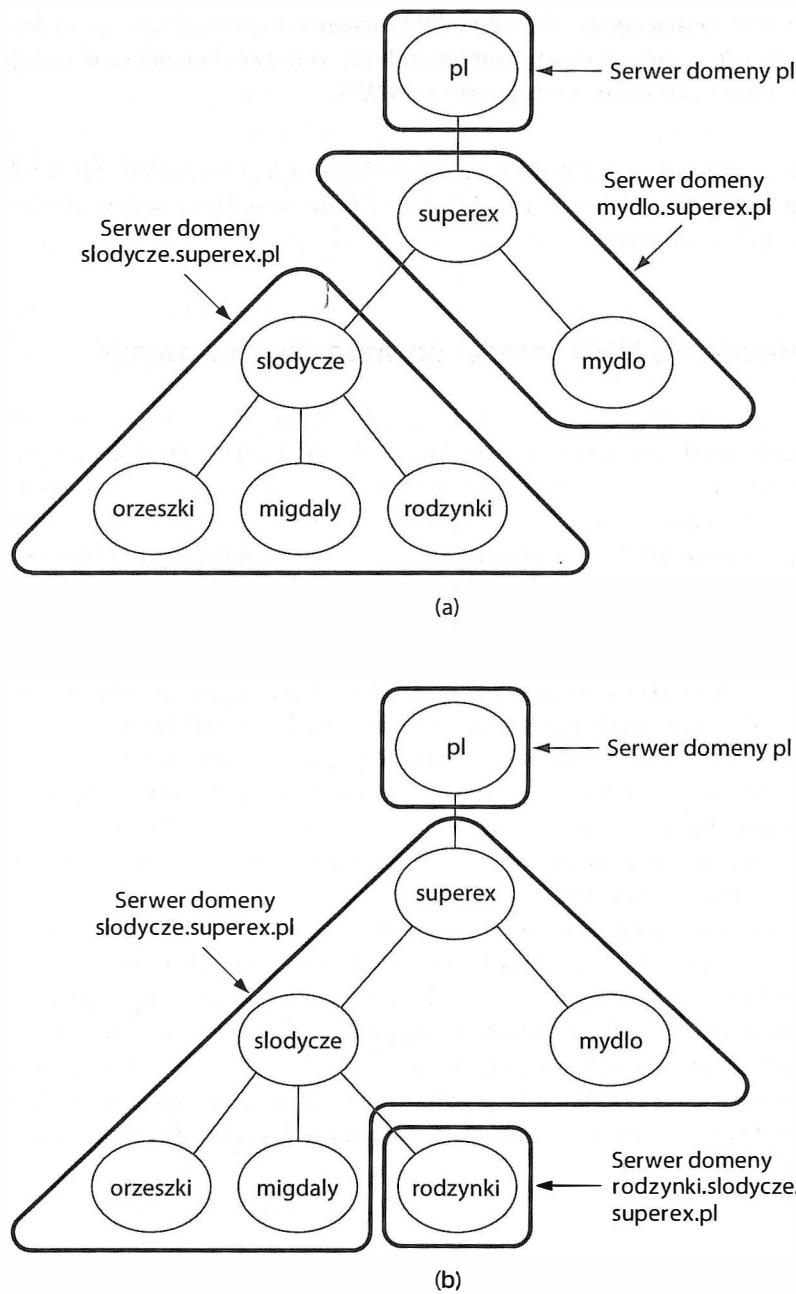
## 4.19. Hierarchia DNS i model powiązań serwerowych

Jedną z najważniejszych cech systemu nazw domenowych jest autonomia jednostki. Mechanizm został zaprojektowany w taki sposób, aby każda organizacja mogła przypisywać nazwy własnym komputerom bez potrzeby informowania o tym fakcie centralnego nadzorcy. Aby zapewnić wspomnianą autonomię, każda organizacja odpowiada za konfigurację serwerów DNS we własnej gałęzi drzewa DNS. Zatem Politechnika Poznańska zarządza serwerem obsługującym nazwy kończące się na *put.poznan.pl*. Z kolei do zadań firmy IBM należy utrzymanie serwerów odpowiedzialnych za tłumaczenie nazw kończących się na *ibm.com*. Każdy serwer DNS zawiera informacje, które umożliwiają wskazanie kolejnego serwera nazw domenowych zarówno na niższym, jak i na wyższym poziomie hierarchii. Ponadto informacje danego serwera mogą być **replikowane**, tak aby istniało kilka fizycznych kopii tych samych definicji. Replikacja okazuje się niezwykle użyteczna w przypadku mocno obciążonych jednostek, takich jak **serwery główne** (ang. *root servers*), udostępniające informacje na temat domen najwyższego poziomu. Administrator systemu nazw musi jednak zagwarantować, że wszystkie kopie są ze sobą zsynchronizowane i dostarczają jednakowych informacji.

Szczegóły konfiguracji serwerów zawsze pozostają w gestii organizacji będącej właścicielem domeny. Niewielkie firmy (wykorzystujące kilka komputerów) często zawierają umowy z dostawcami usług internetowych na utrzymanie zewnętrznych serwerów DNS. Natomiast duże przedsiębiorstwa, które korzystają z własnych serwerów, często zapisują nazwy na jednym ze swoich komputerów lub dzielą zbiór nazw, rejestrując je na większej liczbie serwerów. Na rysunku 4.10 przedstawiono hipotetyczną sytuację, w której firma Superex sp. z o.o. opracowała podział nazw domenowych między działy zajmujące się produkcją słodyczy i mydła.

## 4.20. Odwzorowanie nazw

Proces zamiany nazw domenowych na odpowiadające im adresy komputerów jest nazywany **odwzorowaniem nazw**. Zadanie to realizuje specjalny program systemu operacyjnego (potocznie określany jako *resolver*). W przypadku stosowania interfejsu API gniazd odwołanie do programu odwzorowującego nazwy następuje w chwili wywołania funkcji *gethostbyname*. Program ten staje się wówczas klientem, który kontaktuje się z serwerem DNS i przekazuje uzyskaną odpowiedź do aplikacji wywołującej.



Rysunek 4.10. Drzewa nazw DNS z dwoma przykładowymi sposobami powiązania serwerów

Każdy program odwzorowujący nazwy jest skonfigurowany w taki sposób, aby dysponował co najmniej jednym adresem **lokalnego** serwera nazw<sup>15</sup>. Jego zadanie polega na przygotowaniu **żądania DNS**, przesłaniu go do lokalnego serwera DNS i zaczekaniu na odesłanie przez serwer **odpowiedzi DNS**. Program odwzorowujący nazwy może korzystać z transmisji strumieniowej lub posłużyć się komunikatami — większość systemów bazuje na komunikatach, ponieważ wnoszą one mniejszy narzut transmisyjny.

Jako przykład odwzorowania nazwy rozważmy drzewo zależności przedstawione na rysunku 4.10a. Założymy, że komputer pracujący w dziale mydeł generuje żądanie zamiany nazwy *czekolada.slodycze.superex.pl* na adres jednostki. Program odwzorowujący nazwy może być skonfigurowany w taki sposób, by wysyłał żądania do lokalnego serwera DNS o nazwie *superex.pl*. Choć wybrany serwer nie może samodzielnie udzielić odpowiedzi, wie, że musi się skontaktować z serwerem obsługującym domenę *slodycze.superex.pl*, który dostarczy niezbędnych informacji.

## 4.21. Buforowanie danych w systemie DNS

Zasada **preferowania lokalnego serwera**, stanowiąca podstawę działania wszelkich mechanizmów buforujących, w przypadku systemu nazw domenowych sprawdza się z dwóch powodów:

- Aspekt przestrzenny: użytkownicy znacznie częściej poszukują nazw komputerów lokalnych niż jednostek zdalnych.
- Aspekt czasowy: dany użytkownik zazwyczaj wielokrotnie poszukuje nazw tych samych serwerów.

Aspekt przestrzenny został omówiony we wcześniejszym przykładzie — program odwzorowujący nazwy kontaktuje się w pierwszej kolejności z lokalnym serwerem. Aspekt czasowy jest uwzględniany w mechanizmie buforowania wszystkich odpowiedzi serwerów DNS (algorytm 4.4).

Zgodnie z algorytmem odebranie przez serwer żądania dotyczącego nazwy, która nie jest na nim zarejestrowana, wiąże się z koniecznością wykonania kolejnych operacji klient-serwer. Dany serwer staje się na chwilę klientem w relacji z innym serwerem nazw. Gdy otrzyma odpowiedź z serwera zdalnego, buforuje ją w pamięci podręcznej, a następnie przesyła odpowiedź do jednostki, która dostarczyła żądanie. Z tego względu poza adresami wszystkich serwerów niższego poziomu hierarchii każdy serwer DNS musi również dysponować adresem serwera głównego.

W każdym mechanizmie buforowania informacji istotny jest czas przechowywania danych w pamięci podręcznej. Jeśli jest on zbyt długi, informacje stają się przedawnione. W systemie DNS problem ten został rozwiązany w ten sposób, że serwer, w którym nazwa jest zarejestrowana, określa czas jej przechowywania w pamięci podręcznej innych serwerów. Dzięki temu, gdy lokalny serwer poszukuje adresu dla danej nazwy, otrzymuje

<sup>15</sup> Zalety odwoływanego się w pierwszej kolejności do lokalnych serwerów nazw zostaną wyjaśnione w podrozdziale dotyczącym buforowania odwzorowań.

**Algorytm 4.4.** Postępowanie serwera DNS w czasie odwzorowywania nazwy na adres jednostki

Dane:

Żądanie dostarczone przez moduł odwzorowania nazw

Wynik:

Odpowiedź zawierająca adres jednostki

Realizacja:

Wyodrębnienie nazwy z żądania

```
if (nazwa jest zarejestrowana w serwerze) {
```

Przygotowanie odpowiedzi i przesłanie jej do jednostki,  
która wygenerowała żądanie

```
} else if (nazwa jest w pamięci podręcznej) {
```

Przygotowanie odpowiedzi i przesłanie jej do jednostki,  
która wygenerowała żądanie

```
} else { /*Konieczne jest ustalenie odpowiedzi */
```

```
if (znany jest serwer, w którym nazwa jest  
zarejestrowana) {
```

Przesłanie żądania do serwera, w którym nazwa jest  
zarejestrowana

```
} else {
```

Przesłanie żądania do serwera głównego

```
}
```

Odebranie odpowiedzi i zapisanie jej w pamięci podręcznej

Przygotowanie odpowiedzi i przesłanie jej do jednostki,  
która wygenerowała żądanie

```
}
```

**rekord zasobu**, zawierający zasadniczą odpowiedź oraz informację o czasie jej buforowania. Korzystanie z pamięci podręcznej oznacza obowiązek respektowania tego ustawienia. Ogólna zasada stanowi, że:

*Ponieważ każdy rekord zasobów generowany przez upoważniony do tego serwer DNS zawiera informacje o czasie ważności, dane przechowywane w pamięci podręcznej DNS mogą być usuwane w chwili, gdy staną się nieważne.*

Buforowanie informacji nie ogranicza się jedynie do działania serwerów DNS. Pamięcią podręczną dysponują również programy odpowiedzialne za odwzorowywanie nazw. W praktyce większość systemów operacyjnych zawiera moduły odwzorowania nazw, które buforują wyniki wcześniejszych operacji. Dzięki temu podczas poszukiwania adresów odpowiadających wcześniej przetwarzanym nazwom nie muszą obciążać sieci. Wystarczy, że odwołają się do pamięci podręcznej utworzonej na lokalnym dysku.

## 4.22. Rodzaje wpisów DNS

Każdy wpis występujący w bazie danych DNS składa się z trzech elementów — nazwy domenowej, typu rekordu oraz wartości. Typ rekordu wyznacza sposób interpretacji wartości (informując na przykład o tym, że dana wartość jest adresem IP). Zapytania dostar-

czane do serwerów DNS zawierają zarówno nazwę domenową, jak i informację o typie rekordu. Serwer zwraca jedynie te wyniki, które pasują do typu zapytania.

Podstawowy typ wpisów odpowiada za odwzorowanie nazwy domenowej na adres IP. W systemie DNS powiązania tego rodzaju są klasyfikowane jako wpisy typu A. Rekordy typu A są wykorzystywane przez takie programy jak klient FTP, przeglądarka lub polecenie ping. Wśród wielu innych typów zdefiniowanych w specyfikacji DNS znajduje się również typ MX, odpowiadający serwerowi poczty (*Mail eXchanger*). Podczas analizowania nazwy domenowej zawartej w adresie e-mail moduł SMTP definiuje zapytanie jako żądanie typu MX. Odpowiedź zwracana przez serwer musi odpowiadać wskazanemu rodzajowi wpisów. Zatem komponent dostarczania poczty otrzymuje odpowiedź zgodną z typem MX. Należy więc zapamiętać, że:

*Każdy wpis w bazie danych serwera DNS ma określony typ. Gdy jednostka kliencka żąda odwzorowania nazwy, określa pożądaną typ odpowiedzi, a serwer DNS zwraca jedynie te wartości, które odpowiadają podanemu typowi.*

Uzależnienie zwracanych wyników od typu wpisu może spowodować, że system DNS zwróci niespodziewany dla użytkownika wynik. Założymy na przykład, że przedsiębiorstwo postanowiło wykorzystać nazwę *firma.pl* zarówno do świadczenia usług WWW, jak i przekazywania poczty. System DNS umożliwia rozłożenie obciążenia między dwa niezależne komputery przez zdefiniowanie rekordu A wskazującego jeden serwer oraz rekordu MX odnoszącego się do drugiej jednostki. Wadą takiego rozwiązania jest to, że może się ono okazać niezrozumiałe dla części osób. Może się bowiem okazać, że dostarczanie poczty do odbiorców w domenie *firma.pl* będzie przebiegało bezproblemowo, mimo że dostęp do serwera WWW lub wykonanie polecenia ping w odniesieniu do tego serwera będą niemożliwe.

## 4.23. Aliasy nazw i rekordy CNAME

System DNS zawiera definicję typu CNAME, którego znaczenie jest zbliżone do dowiązania symbolicznego w systemie plików. Wpis ten jest bowiem aliasem innego wpisu w bazie danych DNS. Aby uświadomić sobie użyteczność aliasów, przeanalizujmy przykład dwóch komputerów o nazwach *hobbes.domena.pl* i *kalwin.domena.pl*. Przyjmijmy, że właściciel domeny postanowił uruchomić serwer WWW w systemie *hobbes* i chce zachować konwencję nazewniczą, przypisując komputerowi z usługą WWW nazwę *www*. Choć można by zmienić nazwę jednostki *hobbes*, istnieje znacznie łatwiejsze rozwiązanie problemu. Wystarczy utworzyć rekord CNAME z adresem *www.domena.pl*, który będzie odnosił się do komputera *hobbes*. Dzięki temu po każdorazowym odebraniu żądania odwzorowania nazwy *www.domena.pl* serwer będzie mógł zwrócić adres jednostki *hobbes*.

Aliasy nazw są niezwykłe użytkiczne, gdyż pozwalają organizacjom na zmianę komputerów udostępniających poszczególne usługi bez konieczności modyfikowania ich nazw lub adresów. W poprzednim przykładzie właściciel domeny mógłby przenieść usługę WWW z serwera *hobbes* do komputera *kalwin* i zmienić jedynie treść rekordu CNAME w serwerze

DNS. Obydwa komputery zachowałyby wówczas pierwotne nazwy i adresy IP. Aliasy umożliwiają również kojarzenie wielu nazw z pojedynczym komputerem. Gdyby analizowana firma uruchomiła serwer FTP i serwer WWW w systemie tego samego komputera, mogłaby zdefiniować następujące rekordy CNAME:

www.domena.pl  
ftp.domena.pl

## 4.24. Skróty w systemie DNS

System DNS nie obsługuje skrótów — serwer odpowiada jedynie na zapytania o pełne nazwy. Niemniej większość programów realizujących odwzorowanie można skonfigurować w taki sposób, aby operowały zbiorem sufiksów, umożliwiając jednocześnie użytkownikom posługiwanie się skróconymi wersjami nazw. Na przykład odwzorowanie nazw w firmie Superex może zostać skonfigurowane tak, aby operacja pozyskania adresu była realizowana dwukrotnie — raz w odniesieniu do podanej wartości i raz z dołączonym sufiksem *superex.pl*. Jeśli użytkownik wprowadzi pełną nazwę domenową, lokalny serwer zwróci odpowiedni adres i zadanie będzie kontynuowane. Jeżeli jednak podana zostanie skrócona forma nazwy, moduł odwzorowania najpierw spróbuje zamienić wprowadzoną wartość na adres IP, a gdy otrzyma informację o niemożności ustalenia adresu, ponownie próbę z nazwą uzupełnioną o sufiks. Ponieważ moduł odwzorowania nazw działa w komputerze użytkownika końcowego, rozwiązywanie to umożliwia zdefiniowanie listy sufiksów, które będą wykorzystywane w odpowiedniej kolejności.

Oczywiście, umożliwienie każdemu użytkownikowi konfigurowania mechanizmu odwzorowania nazw ma pewną wadę — wartości wprowadzane przez jednego użytkownika często różnią się od ustawień zdefiniowanych przez innego użytkownika. Jeśli więc dwie osoby wymieniają się nazwami serwerów (na przykład przesyłając ciąg nazwy domenowej w treści wiadomości e-mail), muszą pamiętać o tym, żeby podawać pełne nazwy, a nie ich skrócone wersje.

## 4.25. Znaki narodowe w nazwach domenowych

System DNS rejestruje nazwy domenowe za pomocą zestawu znaków ASCII. Nie może więc przechowywać ciągów zawierających znaki spoza zestawu ASCII. Problem dotyczy takich języków, jak polski, rosyjski, grecki, chiński czy japoński. W każdym z nich występują litery, które nie mają swojej reprezentacji w zestawie ASCII.

Przez wiele lat organizacja IETF pracowała nad zmianami i rozszerzeniami do standardu DNS uwzględniającymi nazwy ze znakami narodowymi. Po rozważeniu wielu propozycji zdecydowała się ostatecznie na zastosowanie rozwiązania znanego jako **internacjonalizacja nazw domenowych w aplikacjach** (IDNA — *Internationalizing Domain Names in Applications*). Zamiast modyfikowania bazowego systemu DNS wykorzystano mechanizm (IDNA), który posługuje się znakami ASCII do zapisywania nazw o dowolnej treści. Jeśli dana nazwa domenowa zawiera litery spoza zestawu ASCII, algorytm IDNA przekształca

ją w sekwencję znaków ASCII i zapisuje wynik operacji w bazie danych systemu DNS. Analogiczne przekształcenie jest realizowane w chwili, gdy użytkownik zleci odwzorowanie nazwy. Jej treść jest zamieniana na ciąg ASCII, a uzyskana wartość zostaje przekazana w zapytaniu do serwera DNS. Rozwiążanie IDNA działa dzięki aplikacjom, które dokonują translacji znaków diakrytycznych, którymi posługuje się użytkownik, na sekwencje znaków ASCII, zrozumiałe dla systemu DNS.

Zasady przekształcania nazw domenowych zawierających znaki narodowe są skomplikowane i bazują na wykorzystaniu standardu Unicode<sup>16</sup>. Tłumaczeniem są objęte wszystkie elementy nazwy domenowej, a poszczególne człony ciągu wynikowego mają format:

$$xn--\alpha-\beta$$

Ciąg  $xn--$  jest zarezerwowanym czteroznakowym łańcuchem, informującym o tym, że dany element reprezentuje niestandardową nazwę. Symbol  $\alpha$  reprezentuje podzbior znaków z oryginalnego członu nazwy, które można przedstawić w formacie ASCII. Natomiast symbol  $\beta$  odpowiada ciągowi tekstowemu złożonemu ze znaków ASCII, utworzonemu w sposób, który pozwala aplikacji IDNA na wstawienie znaków spoza zestawu ASCII do sekcji  $\alpha$  i uzyskanie pierwotnej wersji elementu.

Najnowsze wersje powszechnie stosowanych przeglądarek internetowych (Firefox i Internet Explorer) umożliwiają wprowadzanie i wyświetlanie znaków spoza zestawu ASCII, ponieważ ich kod obejmuje algorytm IDNA. Jeśli jednak dany program nie został wyposażony w mechanizm IDNA, prezentowane w nim treści mogą się wydawać użytkownikowi dość nieczytelne. W takim przypadku nazwa domenowa zawierająca znaki diakrytyczne zostanie przedstawiona w formie pokazanej powyżej, wraz z ciągiem początkowym ( $xn--$ ) oraz sekcjami  $\alpha$  i  $\beta$ .

Podsumowując:

*Standard IDNA koduje każdy element nazwy domenowej w formie ciągu znaków ASCII i zakłada, że operacje przekształcania treści widzianych przez użytkownika na ciągi tekstowe zapisywane w systemie DNS są realizowane przez aplikacje sieciowe.*

## 4.26. Rozszerzalne formaty reprezentacji danych (XML)

Omówione w tym rozdziale tradycyjne protokoły warstwy aplikacji bazują na stałej reprezentacji. Oznacza to, że protokół aplikacji operuje stałym zbiorem komunikatów, które klient i serwer mogą wymieniać, a także dokładnie sprecyzowanym formatem danych, który towarzyszy tym komunikatom. Największą wadą zamkniętego rozwiązania jest to, że bardzo trudno jest wprowadzić jakiekolwiek zmiany w jego działaniu. Doskonałym przykładem jest tutaj standard poczty elektronicznej, który ograniczał treść przekazywanych wiadomości do czystego tekstu. Aby wprowadzić rozszerzenia MIME, konieczna była istotna zmiana w sposobie funkcjonowania mechanizmu.

<sup>16</sup> Algorytm używany do kodowania opisów ze znakami spoza zestawu ASCII jest nazywany algorytmem Puny, a generowane przez niego ciągi tekstowe określa się mianem punycode (ang. punycode).

Alternatywą dla zamkniętego algorytmu reprezentacji danych jest system rozszerzalny, który pozwala nadawcy definiować własne formaty danych. Jeden ze standardów rozszerzalnego formatu przekazywania informacji — rozszerzalny język znaczników (XML — ang. *Extensible Markup Language*) — stał się wyjątkowo popularnym rozwiązaniem. Format XML jest zbliżony do formatu HTML, gdyż podobnie jak on zawiera znaczniki w tekstuowej treści dokumentu. Jednak w przeciwieństwie do specyfikacji HTML znaczniki standardu XML nie zostały wstępnie zdefiniowane i nie reprezentują poleceń formatowania. Format XML służy do opisu struktury danych i uwzględnia nazwy poszczególnych pól zbioru danych. Znaczniki XML są odpowiednio rozłożone w dokumencie — każdemu wystąpieniu znacznika `<x>` musi odpowiadać jedno wystąpienie znacznika `</x>`. Ponadto dzięki temu, że w standardzie nie określono znaczników o specjalnym znaczeniu, nazwy znaczników można dobierać dowolnie. Na przykład jeśli dwie firmy będą chciały wymieniać informacje o numerach telefonów służbowych, mogą zdefiniować format dokumentu XML, w którym zostaną zawarte takie elementy jak dane pracownika, numer telefonu, lokalizacja biura itp. Dane pracownika mogą następnie zostać podzielone na pola imienia i nazwiska, tak jak to zostało pokazane na rysunku 4.11.

```
<adres>
  <pracownik>
    <imie>Jan</imie>
    <nazwisko>Kowalski</nazwisko>
  </pracownik>
  <biuro>Pokój 320</biuro>
  <telefon>52 997 997 99</telefon>
</adres>
```

Rysunek 4.11. Przykład dokumentu XML odpowiadającego firmowej książce telefonicznej

## 4.27. Podsumowanie

Protokoły warstwy aplikacji, niezbędne w pracy standardowych usług, odpowiadają za definiowanie sposobu reprezentowania danych oraz za właściwe działanie mechanizmów transportu danych. Reprezentacja danych w systemie WWW jest opisana w standardach hipertekstowego języka znaczników (HTML) oraz ujednoliconego formatu adresowania zasobów (URL). Komunikacja między przeglądarką a serwerem WWW (niezbędna do pobierania i wysyłania informacji) jest natomiast zdefiniowana w protokole transferu dokumentów hipertekstowych (HTTP). Aby przyspieszyć wyświetlanie stron, przeglądarka buforuje treść dokumentów i wykorzystuje polecenie HEAD protokołu HTTP do pozyskiwania informacji na temat bieżącego stanu strony. Jeśli zbuforowana wersja dokumentu jest aktualna, dane są pobierane z pamięci podręcznej. W przeciwnym przypadku przeglądarka wysyła żądanie typu GET, aby pobrać bieżącą treść.

Protokół HTTP bazuje na tekstowych komunikatach. Każda odpowiedź serwera rozpoczyna się od nagłówka, który opisuje tę odpowiedź. Wiersze nagłówka rozpoczynające się od wartości liczbowych zapisanych w formacie ASCII są kodami statusowymi (przekazują na przykład informację o błędnej składni żądania). Dane następujące po nagłówku mogą zawierać dowolne wartości binarne.

Do pobierania plików z serwerów często wykorzystywana jest usługa FTP. Wymaga ona od użytkownika zalogowania się w systemie serwera, choć uwzględnia również dostęp anonimowy z wykorzystaniem loginu *anonymous* i hasła *guest*. Najciekawszym elementem w działaniu serwera FTP jest jego niestandardowy sposób nawiązywania połączeń. Jednostka kliencka ustanawia bowiem połączenie sterujące, które jest wykorzystywane do przekazywania instrukcji. Jednak w chwili gdy serwer przystępuje do przesyłania danych (na przykład pliku lub listingu z zawartością katalogu), zaczyna działać jak klient, a dotychczasowa jednostka kliencka staje się serwerem. Serwer inicjuje nowe połączenie z klientem, które zamyka natychmiast po przesłaniu danych (pliku lub listingu).

Do obsługi poczty elektronicznej wykorzystywane są trzy rodzaje protokołów warstwy aplikacji, odpowiedzialne za przesyłanie wiadomości, reprezentowanie danych oraz dostęp do skrzynek pocztowych. Podstawowym standardem transferu informacji jest protokół SMTP, który obsługuje jedynie komunikaty tekstowe. Reprezentacja wiadomości została opisana w dwóch specyfikacjach — RFC2822 oraz MIME. Zalecenie RFC2822 stanowi, że wiadomość e-mail składa się z nagłówka i treści rozdzielonych pustym wierszem. Standard MIME definiuje natomiast mechanizmy przesyłania danych binarnych w formie załączników do listu elektronicznego. W rozwiązaniu tym wymagane jest dodanie specjalnego wiersza nagłówka, w którym musi być zawarta informacja na temat sposobu interpretowania wiadomości. Ponadto zgodnie ze specyfikacją MIME nadawca ma obowiązek zakodować plik w formie drukowalnego tekstu.

Protokoły dostępu do poczty, takie jak POP3 i IMAP, umożliwiają użytkownikom korzystanie z ich skrzynek pocztowych. Ten rodzaj komunikacji stał się niezwykle popularny wraz z udostępnieniem skrzynek pocztowych na serwerach dostawców usług internetowych.

Za tłumaczenie nazw wykorzystywanych przez ludzi na adresy komputerów odpowiada system nazw domenowych (DNS). System DNS składa się z wielu serwerów, z których każdy kontroluje pewną część przestrzeni nazw. Między serwerami istnieje zależność hierarchiczna, a każdy węzeł drzewa dysponuje informacjami o umiejscowieniu jednostek sąsiednich w hierarchii.

Serwery DNS wykorzystują pamięć podręczną, która zwiększa wydajność odwzorowań. Każda odpowiedź udzielona przez serwer obsługujący daną domenę jest zapisywana w pamięci podręcznej przekazujących ją serwerów. Aby zapobiec propagowaniu przedawionych informacji, administrator domeny określa dopuszczalny czas przechowywania nazw w buforach serwerów pośrednich.

## ZADANIA

- 4.1. Jakiie elementy są definiowane w protokole warstwy aplikacji?
- 4.2. Dlaczego dokumentacja protokołów standardowych usług jest niezależna od ich implementacji?
- 4.3. Wymień dwa zasadnicze elementy protokołów warstwy aplikacji i opisz, co zawierają.
- 4.4. Podaj przykłady protokołów sieciowych i wskaż dwa elementy protokołu warstwy aplikacji.
- 4.5. Wymień cechy charakterystyczne formatu HTML.

- 4.6. Wymień cztery elementy adresu URL. Jaki znaki są wykorzystywane do rozdzielenia poszczególnych elementów?
- 4.7. Wymień cztery rodzaje żądań HTTP. W jakich przypadkach każde z nich jest wykorzystywane?
- 4.8. W jaki sposób przeglądarka otrzymuje informację o tym, że żądanie HTTP jest niepoprawne lub że wskazany dokument nie istnieje?
- 4.9. Jakiego rodzaju informacje są rejestrowane w pamięci podręcznej przeglądarki? Kiedy buforowanie jest wykorzystywane?
- 4.10. Opisz operacje wykonywane przez przeglądarkę w celu ustalenia, czy można wykorzystać dokument zapisany w pamięci podręcznej.
- 4.11. Czy przeglądarka może operować protokołami innymi niż HTTP? Uzasadnij odpowiedź.
- 4.12. Ile połączeń TCP jest ustanawianych w chwili, gdy użytkownik zażąda wyświetlenia zawartości katalogu FTP?
- 4.13. Gdy użytkownik posługuje się aplikacją FTP, aplikacja ta działa zarówno jako klient, jak i jako serwer. Prawda czy fałsz? Uzasadnij odpowiedź.
- 4.14. Skąd serwer FTP uzyskuje informacje o numerze portu przeznaczonego na połączenie danych?
- 4.15. Czy zgodnie z pierwotną zasadą działania poczty elektronicznej użytkownik mógł odbierać wiadomości, jeśli w jego systemie nie pracował serwer pocztowy?
- 4.16. Wymień trzy rodzaje protokołów stosowanych w systemie poczty elektronicznej. Opisz każdy z nich.
- 4.17. Wymień cechy charakterystyczne protokołu SMTP.
- 4.18. Czy protokół SMTP może przekazywać wiadomości pocztowe zawierające znak kropki w oddzielnym wierszu? Uzasadnij odpowiedź.
- 4.19. W jakich sytuacjach wykorzystywany jest protokół dostępu do poczty elektronicznej?
- 4.20. Wymień dwa główne protokoły dostępu do poczty elektronicznej.
- 4.21. Dlaczego opracowano mechanizm MIME?
- 4.22. Jaki jest ogólne przeznaczenie systemu nazw domenowych?
- 4.23. Ile jest domen najwyższego poziomu, zakładając, że organizacja ISO zdefiniowała  $N$  kodów krajowych?
- 4.24. Serwer WWW musi dysponować nazwą rozpoczęającą się od liter www. Prawda czy fałsz? Uzasadnij odpowiedź.
- 4.25. Międzynarodowe przedsiębiorstwo może podzielić nazwę domenową w taki sposób, aby w Europie, Azji i Ameryce Północnej funkcjonowały niezależne serwery DNS tej firmy. Prawda czy fałsz?
- 4.26. W jakich przypadkach serwer nazw przesyła żądanie do serwera zarządzającego domeną, a w jakich przypadkach odsyła odpowiedź bez kontaktowania się z serwerem zarządzającym domeną?
- 4.27. Serwer DNS może zwracać różne adresy IP w zależności od tego, czy w zapytaniu została wskazana usługa e-mail, czy WWW. Prawda czy fałsz? Uzasadnij odpowiedź.
- 4.28. Czy standard IDNA wymaga wprowadzania zmian w oprogramowaniu serwerów DNS? Klientów DNS? Uzasadnij odpowiedź.
- 4.29. Poszukaj w internecie informacji na temat iteracyjnego odwzorowania nazw domenowych. W jakich sytuacjach realizowane jest iteracyjne odwzorowanie?
- 4.30. W jaki sposób format XML umożliwia aplikacji definiowanie niestandardowych pól, takich jak nazwisko lub adres?

# CZĘŚĆ II

## Wymiana danych

**Podstawowe informacje na temat mediów transmisyjnych, kodowania, przesyłania informacji, modulacji, multipleksacji, połączeń i zdalnego dostępu.**

### Rozdziały:

Rozdział 5. Podstawowe informacje na temat transmisji danych	113
Rozdział 6. Sygnały i źródła informacji	121
Rozdział 7. Media transmisyjne	141
Rozdział 8. Niezawodność i kodowanie kanałowe	163
Rozdział 9. Tryby transmisji danych	179
Rozdział 10. Modulacja i modemy	191
Rozdział 11. Multipleksacja i demultipleksacja	205
Rozdział 12. Technologie łączysty dostępowych i rdzeniowych	221

## **Zawartość rozdziału**

- 5.1. Wprowadzenie 113
- 5.2. Istota transmisji danych 114
- 5.3. Założenia i zakres zagadnienia 114
- 5.4. Teoretyczne elementy systemu komunikacyjnego 115
- 5.5. Elementy modelu transmisji danych 116
- 5.6. Podsumowanie 118

# 5

## *Podstawowe informacje na temat transmisji danych*

### **5.1. Wprowadzenie**

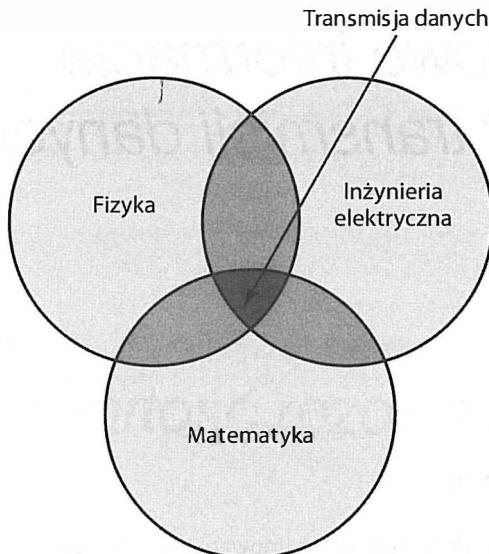
Pierwsza część książki dotyczyła programowania sieciowego oraz aplikacji internetowych. W rozdziale poświęconym gniazdom omówiony został interfejs programistyczny, który jest implementowany w systemie operacyjnym i udostępniany aplikacjom sieciowym. Z zamieszczonych tam informacji wynika, że programiści mogą z niego korzystać bez konieczności dogłębniego poznawania mechanizmów transmisji sieciowej. W dalszych częściach książki zaprezentowane zostaną jednak protokoły i technologie, które zapewniają komunikację między jednostkami sieciowymi. Zapoznając się z tymi zagadnieniami, będzie się można przekonać, że zrozumienie wszelkich niuansów w funkcjonowaniu opisywanych mechanizmów może znacznie poprawić jakość pisanej kodu.

W tej części publikacji omówione zostało zagadnienie transmisji informacji w mediach takich jak przewody, włókna optyczne i fale radiowe. Choć szczegóły implementacji poszczególnych rozwiązań są różne, ogólne zasady przekazywania informacji i zapewnienia komunikacji urządzeń są niezmienne niezależnie od formy transmisji. Transmisja danych jako dziedzina nauki zapewnia pojęciowe i analityczne narzędzia ułatwiające opracowanie spójnego modelu działania systemów komunikacyjnych. Ponadto umożliwia zestawienie tego, jakie rodzaje transmisji są teoretycznie możliwe, a jakie można praktycznie zrealizować.

Rozdział ten zawiera ogólne omówienie idei transmisji danych oraz komponentów pełnego systemu komunikacyjnego. Uszczegółowienie poszczególnych zagadnień znajduje się natomiast w kolejnych rozdziałach.

## 5.2. Istota transmisji danych

Co kryje się pod pojęciem transmisji danych? Zgodnie z rysunkiem 5.1 zagadnienie to jest niezwykle ciekawym połączeniem idei pochodzących z trzech różnych dyscyplin naukowych.



Rysunek 5.1. Transmisja danych jest elementem wspólnym fizyki, matematyki i inżynierii elektrycznej

Z uwagi na przekazywanie informacji w medium fizycznym, wymiana danych obejmuje elementy fizyki. Bazuje na technikach związanych z przepływem prądu, propagacją światła oraz innymi formami emisji fal elektromagnetycznych. Przechowywanie i przenoszenie informacji w postaci cyfrowej sprawia, że w transmisji danych niezbędne są odwoływanie do matematyki i różnych rodzajów analiz matematycznych. Ponieważ jednak ostatecznym celem każdego projektu jest opracowanie i zbudowanie systemu przesyłania danych, konieczne jest uwzględnienie w tym procesie również rozwiązań z dziedziny inżynierii elektrycznej.

*Mimo że transmisja danych jako dziedzina nauki odwołuje się do matematyki i fizyki, nie ogranicza się do formułowania abstrakcyjnych teorii. Zapewnia natomiast podstawy teoretyczne do budowy praktycznych systemów komunikacyjnych.*

## 5.3. Założenia i zakres zagadnienia

Trzy podstawowe założenia transmisji danych wyznaczają jednocześnie zakres tego zagadnienia.

- Źródła danych mogą mieć dowolny charakter.
- Transmisja bazuje na wykorzystaniu fizycznego systemu.
- Medium transmisyjne może być wykorzystywane przez wiele źródeł danych.

Pierwszy punkt jest szczególnie istotny, jeśli weźmie się pod uwagę fakt upowszechniania się aplikacji multimedialnych. Zgodnie z nim informacja nie jest ograniczona jedynie do bitów przechowywanych w komputerze. Może natomiast być pozyskiwana z otaczającego nas świata i mieć charakter przekazu dźwiękowego lub wizualnego. Konieczne wydaje się więc poznanie potencjalnych źródeł i form przekazu informacji, a także zasad przekształcania jednej formy przekazu w inną.

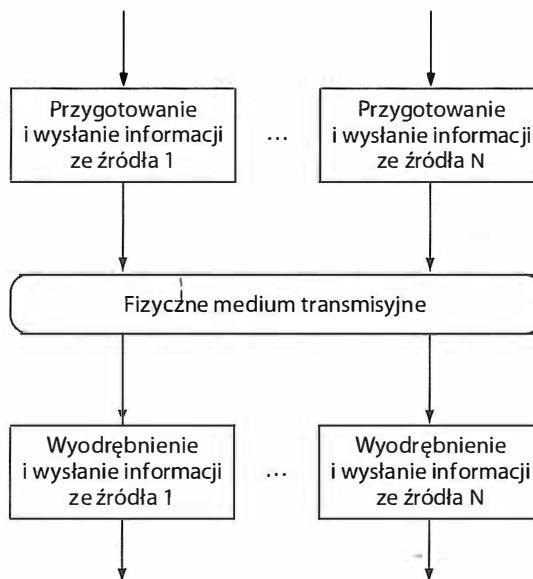
Drugi punkt stanowi, że do przekazu informacji muszą być wykorzystywane naturalne zjawiska, takie jak elektryczność i promieniowanie elektromagnetyczne. Ważne są więc rozróżnianie rodzajów mediów transmisyjnych oraz umiejętność interpretacji ich właściwości. Ponadto projektant systemu komunikacyjnego musi rozumieć sposób wykorzystania zjawisk fizycznych do przekazywania informacji w medium transmisyjnym oraz znać zależności między przepływem danych a wykorzystaną techniką transmisji. Niezbędne jest również poznanie ograniczeń fizycznych systemu, problemów, które mogą wystąpić w trakcie przekazywania danych, oraz technik, które można wykorzystać do wykrywania i rozwiązywania problemów.

Trzeci punkt wskazuje współdzielenie medium jako fundamentalny element systemu transmisji danych. W praktyce istotnie dostęp do wspólnego medium stanowi podstawę funkcjonowania większości sieci komputerowych. Sieć pozwala bowiem na to, aby wiele par jednostek komunikowało się ze sobą jednocześnie w ramach pojedynczego medium transmisyjnego. Istotne jest więc zrozumienie zasad współdzielenia komponentów odpowiedzialnych za wymianę danych, zalet i wad poszczególnych rozwiązań oraz wynikających z nich form komunikacji.

## 5.4. Teoretyczne elementy systemu komunikacyjnego

Aby zrozumieć ideę transmisji danych, należy sobie wyobrazić działający system komunikacyjny, który składa się z wielu źródeł informacji i umożliwia każdemu źródłu wysyłanie danych do innej jednostki docelowej. Wydawałoby się, że komunikacja między jednostkami nie jest szczególnie skomplikowana. Każde źródło musi dysponować mechanizmami zbierania informacji, przygotowania ich do transmisji oraz wysyłania ich za pośrednictwem wspólnego medium. Analogiczne mechanizmy są niezbędne do wyodrębnienia danych po stronie odbiorczej i dostarczenia ich do jednostki docelowej. Opisany schemat postępowania został przedstawiony na rysunku 5.2.

W rzeczywistości transmisja danych jest znacznie bardziej skomplikowana, niż można by wywnioskować z diagramu widocznego na rysunku 5.2. Z uwagi na różnorodność źródeł informacji konieczne jest stosowanie różnych technik przetwarzania danych źródłowych. Przed przekazaniem informacji do medium transmisyjnego trzeba przekształcić je do postaci cyfrowej, a następnie uzupełnić o dodatkowe dane, które zapewnią informacji ochronę przed błędami. W rozwiązaniach wymagających zachowania wysokiego poziomu



Rysunek 5.2. Uproszczony obraz systemu komunikacyjnego obejmującego wiele źródeł danych

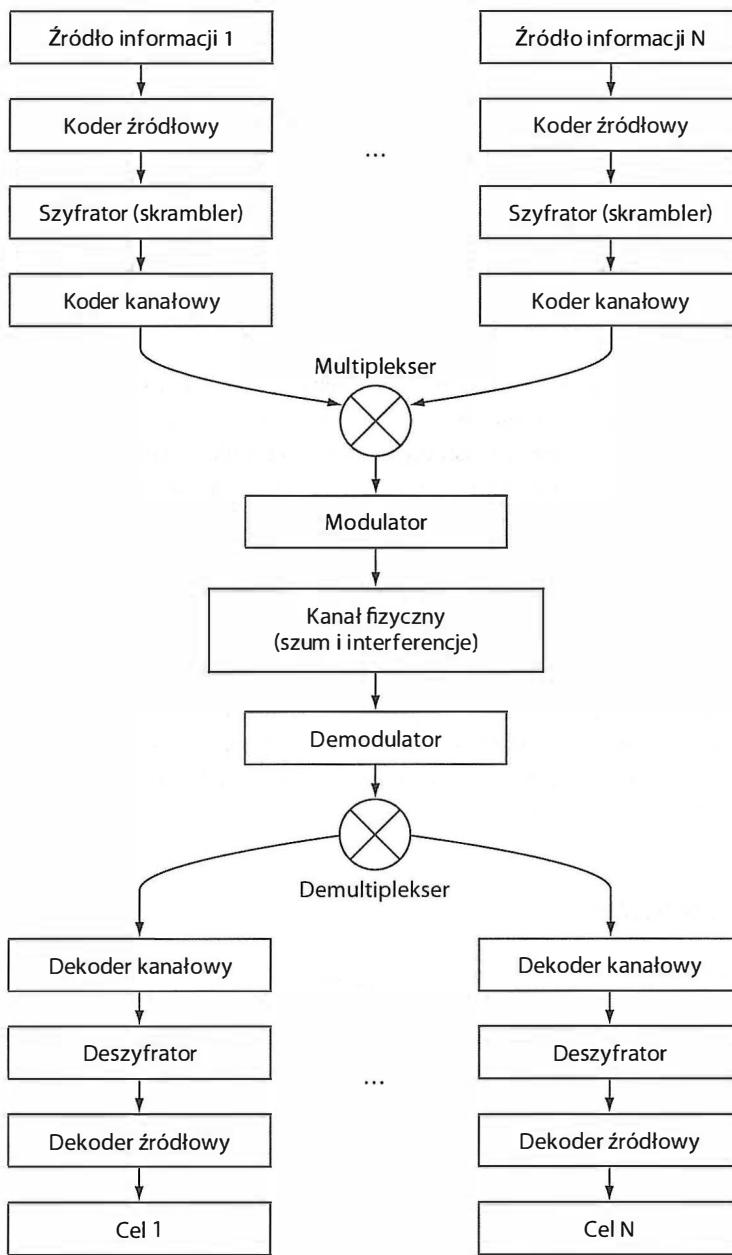
poufności konieczne okazuje się uwzględnienie szyfrowania. Z kolei możliwość przekazywania wielu strumieni danych z różnych źródeł w ramach wspólnego medium oznacza obowiązek oznaczania danych w sposób pozwalający na identyfikację źródła, a także na zaimplementowanie mechanizmów przeplatania informacji pochodzących z różnych źródeł na czas transmisji. Niezbędny jest więc system identyfikacji źródeł danych, który zagwarantuje, że informacje generowane przez określoną jednostkę nie zostaną nieodwracalnie wplecone w informacje innej jednostki.

Aby wyjaśnić najważniejsze elementy procesu transmisji danych, inżynierowie opracowali teoretyczny model systemu, który pozwala na zrozumienie funkcji pełnionych przez poszczególne elementy systemu komunikacyjnego. Każdy komponent modelu można więc analizować niezależnie, a zrozumienie zasad działania każdego z nich pozwala na zrozumienie całego mechanizmu. Wspomniany model pokazano na rysunku 5.3.

## 5.5. Elementy modelu transmisji danych

Każdy element widoczny na rysunku 5.3 odpowiada jednemu zagadnieniu z dziedziny transmisji danych. W kolejnych punktach zamieszczono wyjaśnienie wykorzystanej terminologii. Szczegółowe omówienie poszczególnych bloków znajduje się natomiast w następnych rozdziałach książki.

- **Źródła informacji.** Źródła informacji mogą mieć charakter analogowy lub cyfrowy. Do ich najważniejszych cech należy zaliczyć charakterystykę przetwarzanych sygnałów, czyli amplitudę, częstotliwość, fazę oraz przynależność do grupy sygnałów okresowych lub nieokresowych. W źródłach danych realizowana jest również konwersja danych analogowych na dane cyfrowe.



Rysunek 5.3. Teoretyczny model funkcjonowania systemu wymiany danych.  
Wiele jednostek nadawczych przekazuje informacje do wielu jednostek odbiorczych  
za pośrednictwem wspólnego kanału fizycznego

- **Koder źródłowy i dekoder źródłowy.** Po sprowadzeniu informacji do formatu cyfrowego można je poddawać dalszemu przetwarzaniu i kolejnym transformacjom. Mechanizmy implementowane w blokach kodera źródłowego i dekodera źródłowego odpowiadają za kompresję danych i jej wpływ na samą komunikację.

- **Szyfrator i deszyfrator.** Szyfrowanie informacji przed wysłaniem oraz rozszyfrowywanie ich po odbiorze pozwala na ochronę danych i zachowanie ich poufności. Do najważniejszych zagadnień związanych z tym elementem modelu należą techniki i algorytmy kryptograficzne.
- **Koder kanałowy i dekoder kanałowy.** Kodowanie kanałowe jest techniką wykorzystywaną do wykrywania i usuwania błędów transmisyjnych. Wśród najważniejszych związanych z nią zagadnień są metody detekcji i ograniczania błędów transmisyjnych oraz techniki sprawdzania parzystości, generowania sum kontrolnych oraz cyklicznych kodów nadmiarowych, które są stosowane powszechnie w sieciach komputerowych.
- **Multiplekser i demultiplekser.** Multipleksacja jest operacją przeplatania informacji pochodzących z różnych źródeł podczas przesyłania ich we wspólnym medium transmisyjnym. Szczególnie istotne zagadnienia z nią związane to techniki wspólnego korzystania z medium transmisyjnego oraz zasady wyznaczania kolejności w dostępie do medium.
- **Modulator i demodulator.** Terminem „modulacji” określa się technikę wykorzystania fal elektromagnetycznych do przenoszenia informacji. W analizie zagadnienia trzeba uwzględnić rodzaje modulacji analogowych i cyfrowych oraz urządzenia nazywane modemami, które wykonują operacje modulacji i demodulacji sygnałów.
- **Kanał fizyczny i transmisja danych.** Te określenia obejmują media transmisyjne oraz tryby transmisji danych. Do ich opisu wykorzystuje się pojęcia szerokości pasma, szumu i interferencji, pojemności kanału, a także trybów transmisji (szeregowych lub równoległych).

## 5.6. Podsumowanie

Wykorzystanie fizycznego medium transmisyjnego oraz operowanie informacjami cyfrowymi sprawia, że transmisja danych jest dziedziną zależną od matematyki i fizyki. Stanoi jednocześnie podstawę wszelkich technik, które umożliwiają inżynierom projektowanie użytkowych systemów komunikacyjnych.

Chcąc uprosić pracę projektową, inżynierowie opracowali teoretyczny model systemu transmisji danych. Dzięki temu złożony problem został podzielony na kilka niezależnych zagadnień, których szczegółowy opis znajduje się w kolejnych rozdziałach książki.

## ZADANIA

- 5.1. Jakie trzy dyscypliny naukowe stanowią podstawę transmisji danych?
- 5.2. Jakie są założenia transmisji danych?
- 5.3. Wymień elementy modelu opisującego system transmisji danych.
- 5.4. Który z elementów systemu transmisji danych przetwarza analogowe dane wejściowe?
- 5.5. Który z elementów systemu transmisji danych zabezpiecza przekaz przed błędami i przekłamaniami informacji?



# Zawartość rozdziału

- 6.1. Wprowadzenie 121
- 6.2. Źródła informacji 121
- 6.3. Sygnały analogowe i cyfrowe 122
- 6.4. Sygnały okresowe i nieokresowe 122
- 6.5. Przebieg sinusoidalny i cechy sygnału 123
- 6.6. Sygnał zespolony 124
- 6.7. Znaczenie sygnałów zespolonych i sinusoidalnych 125
- 6.8. Reprezentacja sygnału w dziedzinie czasu i częstotliwości 126
- 6.9. Szerokość pasma sygnału analogowego 127
- 6.10. Sygnały cyfrowe i ich poziomy 127
- 6.11. Body i bity na sekundę 129
- 6.12. Przekształcenie sygnału cyfrowego w sygnał analogowy 130
- 6.13. Szerokość pasma sygnału cyfrowego 131
- 6.14. Synchronizacja i uzgodnienia odnośnie sygnałów 131
- 6.15. Kodowanie liniowe 132
- 6.16. Wykorzystanie kodowania Manchester  
w sieciach komputerowych 134
- 6.17. Przekształcenie sygnału analogowego w sygnał cyfrowy 135
- 6.18. Twierdzenie Nyquista i częstotliwość próbkowania 136
- 6.19. Twierdzenie Nyquista w transmisji telefonicznej 137
- 6.20. Kodowanie i kompresja danych 137
- 6.21. Podsumowanie 138

# 6

## *Sygnały i źródła informacji*

### **6.1. Wprowadzenie**

Poprzedni rozdział należy traktować jako wprowadzenie do transmisji danych, stanowiącej podstawę pracy sieciowej. Przedstawiony w nim został teoretyczny model transmisji danych, a także najważniejsze zagadnienia z nim związane oraz zależności między blokami funkcjonalnymi. Nie zabrakło również krótkiego opisu poszczególnych bloków modelu.

Ten rozdział rozpoczyna szczegółową analizę mechanizmów transmisji danych. Omówiono w nim źródła danych i charakterystyki sygnałów przenoszących informacje. W kolejnych rozdziałach kontynuowane są rozważania związane z tym tematem.

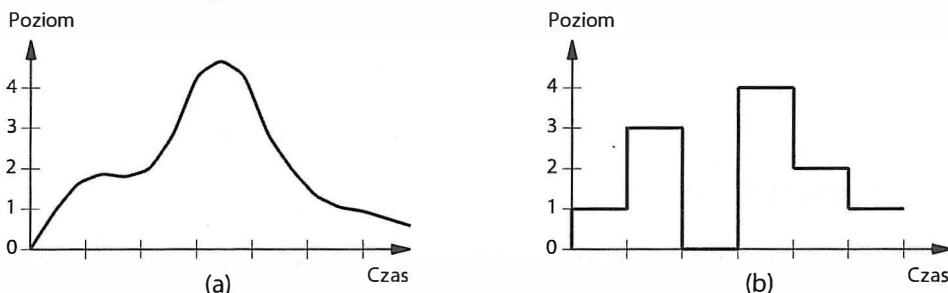
### **6.2. Źródła informacji**

Zgodnie z zamieszczonymi wcześniej informacjami system komunikacyjny umożliwia wprowadzenie danych z dowolnej liczby **źródeł**. Informacje generowane przez każde ze źródeł dostarcza następnie do określonych odbiorników **docelowych**. W przypadku sieci komputerowej, jaką jest internet, źródłami i komponentami docelowymi są aplikacje komputerowe, które przetwarzają dane. Teoria transmisji danych odnosi się jednak do niskopoziomowych systemów komunikacyjnych i znajduje zastosowanie podczas przenoszenia informacji z dowolnych źródeł danych. Poza klasycznymi urządzeniami peryferyjnymi komputerów (takimi jak klawiatura i myszka) jako źródła informacji można również rozpatrywać mikrofony, czujniki oraz urządzenia pomiarowe (na przykład termometry). Komponentami docelowymi mogą być zewnętrzne urządzenia audio, takie jak słuchawki lub głośniki, bądź urządzenia takie jak diody LED, emitujące światło.

*Zgłębiając tajniki transmisji danych, należy pamiętać, że źródłem informacji może być dowolne urządzenie, nie tylko komputer.*

### 6.3. Sygnały analogowe i cyfrowe

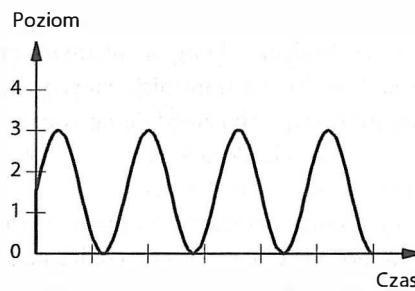
Transmisja danych dotyczy dwóch rodzajów informacji — analogowych i cyfrowych. Sygnał analogowy charakteryzuje się tym, że można go opisać funkcją ciągłą — zmiana jednej wartości na drugą odbywa się na zasadzie przejścia przez wszystkie możliwe wartości pośrednie. Natomiast sygnał cyfrowy składa się jedynie z pewnego ustalonego zbioru dopuszczalnych poziomów, a każda zmiana wartości oznacza natychmiastowe przejście z jednego poziomu do drugiego. Różnice te są widoczne na rysunku 6.1, na którym pokazano zmianę w czasie poziomu sygnałów pochodzących ze źródła analogowego i cyfrowego. Zaprezentowane przebiegi sygnału analogowego i cyfrowego można na przykład uzyskać, dokonując pomiarów odpowiednio na wyjściu mikrofonu i klawiatury.



Rysunek 6.1. Przebieg sygnału analogowego (a) i cyfrowego (b)

### 6.4. Sygnały okresowe i nieokresowe

Ogólnie sygnały podlegają podziałowi na **okresowe** i **nieokresowe** (nazywane również **periodycznymi** i **aperiodycznymi**) w zależności od tego, czy ich kształt się powiela, czy nie. Na przykład sygnał analogowy zaprezentowany na rysunku 6.1a ma charakter nieokresowy, ponieważ jego kształt nie zawiera powtarzających się fragmentów. Z kolei przebieg przedstawiony na rysunku 6.2 jest przebiegiem sygnału okresowego.



Rysunek 6.2. W sygnale okresowym wartości powtarzają się w czasie

## 6.5. Przebieg sinusoidalny i cechy sygnału

Większość analiz związanych z transmisją danych bazuje na wykorzystaniu funkcji trygonometrycznych, w szczególności na funkcji **sinus**, oznaczanej jako **sin**. Przebiegi sinusoidalne są szczególnie istotne w opisie źródeł informacji, ponieważ wynikiem zjawisk naturalnych często jest fala sinusoidalna. Na przykład dźwięki rejestrowane przez mikrofon mają charakter sinusoidalny. Również promieniowanie elektromagnetyczne można przedstawić w formie fali sinusoidalnej. W dalszych rozważaniach szczególnie istotny będzie przebieg sinusoidalny odpowiadający sygnałowi, który jest zmienny w czasie (tak jak przebieg przedstawiony na rysunku 6.2). Najważniejsze jest jednak to, że:

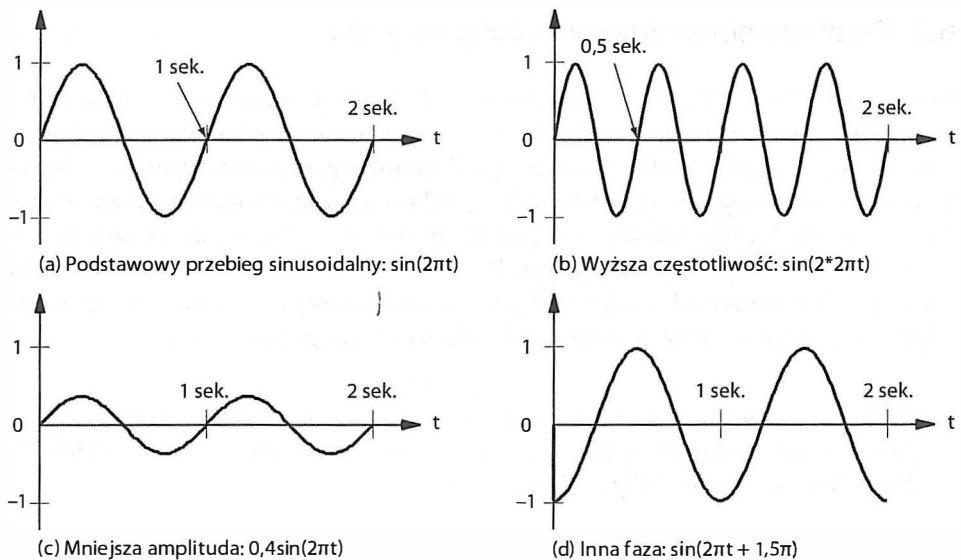
*Funkcje sinusoidalne są niezwykle istotne w przetwarzaniu informacji wejściowych, ponieważ wiele naturalnych zjawisk generuje sygnały, które zmieniając się w czasie, przypominają fale o przebiegu sinusoidalnym.*

Sygnały sinusoidalne mają cztery charakterystyczne cechy:

- Częstotliwość — liczbę oscylacji w jednostce czasu (zazwyczaj w sekundzie).
- Amplitudę — różnicę między maksymalnym i minimalnym poziomem sygnału.
- Fazę — przesunięcie punktu początkowego przebiegu sinusoidalnego względem przebiegu odniesienia.
- Długość fali — długość cyklu (okres) sygnału propagującego w medium transmisyjnym.

Długość fali wynika z prędkości propagacji sygnału (która z kolei zależy od medium transmisyjnego). Trzy pozostałe parametry można wyrazić za pomocą analizy matematycznej. Najłatwiejsza do wyjaśnienia jest amplituda. Funkcja  $\sin(\omega t)$  zwraca wartości z przedziału od -1 do 1 i ma amplitudę 1. Jeśli więc wartość funkcji zostanie przemnożona przez A, to amplituda sygnału będzie miała wartość A. Z matematycznego punktu widzenia fazę można przedstawić jako wartość dodawaną do t, która powoduje przesunięcie przebiegu funkcji w prawo lub w lewo wzduł osi X. Zatem funkcja  $\sin(\omega t + \phi)$  ma fazę  $\phi$ . Częstotliwość sygnału jest mierzona jako liczba okresów funkcji sinusoidalnej w czasie jednej sekundy i jest wyrażana w **hercach**. Pełen okres fali sinusoidalnej odpowiada  $2\pi$  radianom. Zatem jeśli t oznacza czas w sekundach, a  $\omega=2\pi$ , przebieg  $\sin(\omega t)$  ma częstotliwość 1 Hz. Trzy opisane zależności matematyczne zostały przedstawione na rysunku 6.3.

Częstotliwość można obliczyć jako odwrotność czasu potrzebnego do wykonania jednego cyklu zmian, czyli jednego **okresu**. Funkcja przedstawiona na rysunku 6.3a ma okres ( $T$ ) równy jednej sekundzie i częstotliwość wynoszącą  $1/T$ , czyli 1 Hz. Okres przebiegu widocznego na rysunku 6.3b odpowiada wartości 0,5 sekundy, a jego częstotliwość to 2 Hz. W obydwu przypadkach częstotliwość jest bardzo **niska**. Typowe systemy komunikacyjne wykorzystują **wysokie** częstotliwości, liczone w milionach okresów na sekundę.



Rysunek 6.3. Częstotliwość, amplituda i faza

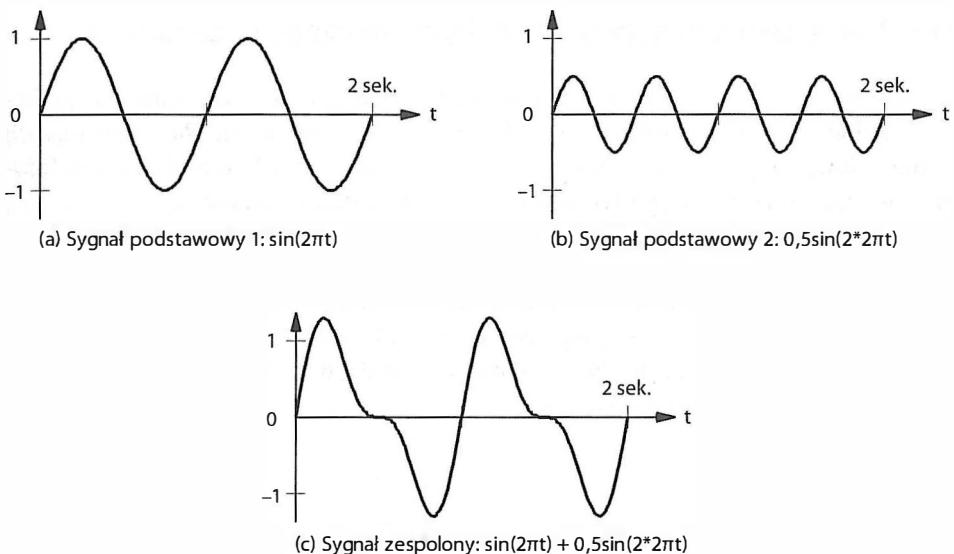
Aby ułatwić sobie zapis wartości, inżynierowie wyrażają czas w ułamkowych częściach sekundy, a częstotliwość w takich jednostkach jak **megaherce**. Zestawienie wartości czasu oraz przedrostków charakterystycznych dla wartości częstotliwości zamieszczono w tabeli 6.1.

Tabela 6.1. Przedrostki i jednostki czasu oraz częstotliwości

Jednostka czasu	Wartość	Jednostka częstotliwości	Wartość
Sekunda (s)	$10^0$ s	Herc (Hz)	$10^0$ Hz
Milisekunda (ms)	$10^{-3}$ s	Kiloherc (kHz)	$10^3$ Hz
Mikrosekunda ( $\mu$ s)	$10^{-6}$ s	Megaherc (MHz)	$10^6$ Hz
Nanosekunda (ns)	$10^{-9}$ s	Gigaherc (GHz)	$10^9$ Hz
Pikosekunda (ps)	$10^{-12}$ s	Teraherc (THz)	$10^{12}$ Hz

## 6.6. Sygnał zespolony

Sygnały o przebiegach odpowiadających tym, które są widoczne na rysunku 6.3, są klasifikowane jako sygnały **podstawowe** (ang. *simple*), ponieważ składają się z pojedynczej fali sinusoidalnej, której nie można poddać dekompozycji. W praktyce większość sygnałów jest klasifikowana jako **zespolone**, gdyż można je przekształcić w zbiór podstawowych fal sinusoidalnych. Na rysunku 6.4 przedstawiono przykładowy sygnał zespolony, który powstał po dodaniu dwóch sygnałów sinusoidalnych.



**Rysunek 6.4.** Przykład sygnału zespolonego powstały w wyniku zsumowania dwóch sygnałów podstawowych

## 6.7. Znaczenie sygnałów zespolonych i sinusoidalnych

Dlaczego transmisja danych wydaje się w tak dużym stopniu uzależniona od funkcji sinusoidalnych i sygnałów zespolonych? Odpowiedź na to pytanie stanie się oczywista po zapoznaniu się z zagadnieniami modulacji i demodulacji sygnałów. Okaże się wówczas, że wynikiem modulacji są zazwyczaj sygnały zespolone. Na bieżącym poziomie znajomości tematu ważne jest, aby pamiętać, że:

- Wynikiem modulacji jest zazwyczaj sygnał zespolony.
- Matematyk o nazwisku Fourier dowódł, że istnieje możliwość rozłożenia sygnału zespolonego na sygnały elementarne, czyli na pojedyncze przebiegi sinusoidalne o odpowiednich częstotliwościach, amplitudach i fazach.

Przeprowadzona przez Fouriera analiza udowodniła, że okresowe sygnały zespolone składają się z przebiegów elementarnych, które również mają charakter okresowy. Dzięki temu większość systemów transmisji danych wykorzystuje sygnały zespolone do przenoszenia informacji — sygnał zespolony jest generowany w jednostce nadawczej, a odbiornik rozkłada go na pojedyncze sygnały składowe.

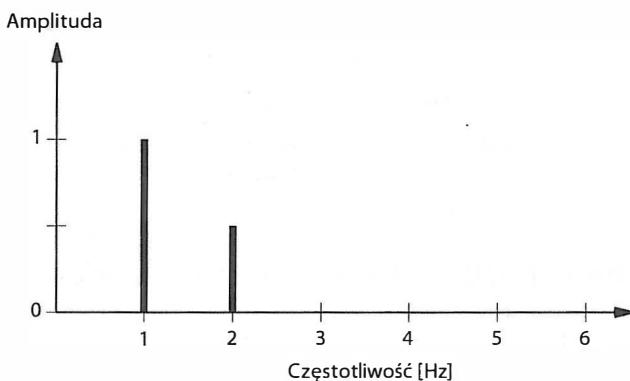
Podsumowując:

Opracowana przez Fouriera matematyczna operacja umożliwia odbiornikowi rozkładanie sygnału zespolonego na jego elementarne składowe.

## 6.8. Reprezentacja sygnału w dziedzinie czasu i częstotliwości

Sygnały zespolone, z uwagi na ich wyjątkowe znaczenie, były studiowane bardzo wnikliwie, co doprowadziło do opracowania kilku metod ich reprezentacji. Na wcześniejszych rysunkach wykorzystana została jedna z jej form — wykres sygnału w funkcji czasu. Inżynierowie nazywają taki wykres reprezentacją sygnału w **dziedzinie czasu**.

Główną konkurencją dla reprezentacji w dziedzinie czasu jest reprezentacja w **dziedzinie częstotliwości**. Wykres w dziedzinie częstotliwości prezentuje zbiór fal sinusoidalnych składających się na przebieg zespolony. Funkcja  $A \sin(2\pi t)$  jest na nim przedstawiona jako pojedyncza linia o wysokości A wykreślona na osi poziomej na pozycji t. Na przykład reprezentacja sygnału zespolonego z rysunku 6.4c w dziedzinie częstotliwości odpowiada wykresowi widocznemu na rysunku 6.5<sup>17</sup>.



Rysunek 6.5. Przedstawienie funkcji  $\sin(2\pi t)$  i  $0,5\sin(2^*2\pi t)$  w dziedzinie częstotliwości

Na rysunku widać zbiór podstawowych sygnałów okresowych. Reprezentacja w dziedzinie częstotliwości znajduje również zastosowanie do przedstawiania sygnałów nieokreślonych, jednak nie ma ona szczególnego znaczenia w przypadku transmisji danych.

Jedną z zalet wykresów w dziedzinie częstotliwości jest ich zwarty sposób przekazu informacji. W porównaniu z przebiegami charakterystycznymi dla dziedziny czasu reprezentacja w dziedzinie częstotliwości jest niewielka pod względem zajmowanego obszaru wykresu, a jednocześnie łatwa do interpretacji. Każdy przebieg sinusoidalny jest bowiem przedstawiony jako pojedynczy punkt na osi poziomej. Zaleta ta uwidacznia się przede wszystkim podczas prezentacji sygnałów zespolonych składających się z wielu sygnałów elementarnych.

<sup>17</sup> Skala osi poziomej na wykresach częstotliwościowych stosowanych w rzeczywistych systemach transmisji danych obejmuje tysiące lub miliony herców.

## 6.9. Szerokość pasma sygnału analogowego

Większość osób z pewnością słyszała o „szerokości pasma sieciowego” i wie, że rozwiązania o szerokim paśmie są rozwiązaniem pożądanym. Definicja szerokości pasma zostanie przedstawiona w dalszej części książki. Tematem tego punktu jest **szerokość pasma sygnału analogowego**. Szerokość pasma sygnału analogowego definiuje się jako różnicę między najwyższą a najniższą częstotliwością sygnałów składowych (tj. najwyższą i najniższą częstotliwością uzyskaną w wyniku przeprowadzenia analizy Fouriera). W tak prostym przypadku, jakim zostało przedstawiony na rysunku 6.4c, wynikiem są sygnały o częstotliwościach 1 i 2 Hz. Oznacza to, że szerokość pasma wynosi 1 Hz. Gdy trzeba obliczyć szerokość pasma, niezaprzeczalna wydaje się przewaga wykresów w dziedzinie częstotliwości, które wprost wyznaczają najniższą i najwyższą częstotliwość. Na przykład z rysunku 6.5 jasno wynika, że pasmo ma szerokość 1 Hz.

Na rysunku 6.6 przedstawiono wykres w dziedzinie częstotliwości, na którym wartości częstotliwości są podawane w kilohercach (kHz). Takie częstotliwości są słyszalne przez ludzkie ucho. Wyznaczona na wykresie szerokość pasma jest różnicą między najwyższą i najniższą częstotliwością i wynosi 4 kHz ( $5 \text{ kHz} - 1 \text{ kHz} = 4 \text{ kHz}$ ).



Rysunek 6.6. Wykres w dziedzinie częstotliwości przedstawiający sygnał analogowy o szerokości pasma 4 kHz

Podsumowując:

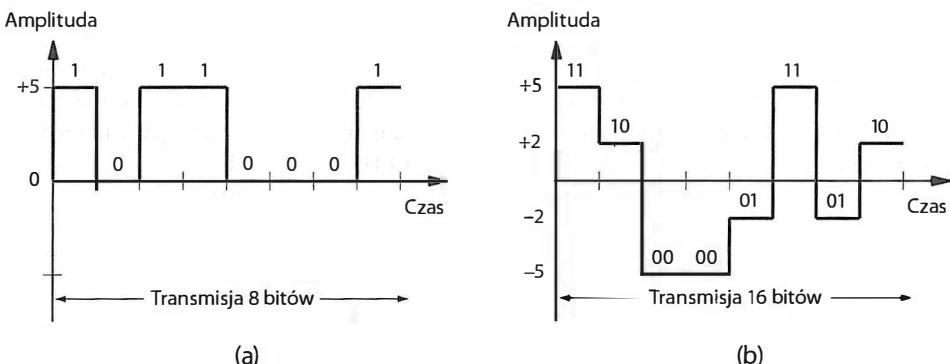
**Szerokość pasma sygnału analogowego** jest różnicą między najwyższą i najniższą częstotliwością sygnałów składowych. Wykreślenie przebiegu sygnału w dziedzinie częstotliwości znacznie ułatwia obliczenie szerokości pasma.

## 6.10. Sygnały cyfrowe i ich poziomy

Zgodnie ze wcześniejszymi informacjami, poza reprezentacją w formie sygnału analogowego informacja może być również przenoszona jako **sygnał cyfrowy**. Jak wiadomo, sygnał ma charakter cyfrowy, jeśli do jego reprezentacji służy zbiór wstępnie ustalonych poziomów, a w danej chwili wartość sygnału odpowiada jednemu z tych poziomów. W niektórych

systemach do odwzorowania wartości cyfrowych służą poziomy napięciowe. Wówczas jedynce logicznej odpowiada dodatnia wartość napięcia, a zeru logicznemu zerowa wartość napięcia. Na przykład do reprezentacji jedynek logicznych można wykorzystać napięcie +5V, a do przedstawienia zer napięcie 0V.

Jeśli wykorzystywane są jedynie dwa poziomy napięć, każdy z nich odpowiada jednemu bitowi danych (zeru lub jedynce). Niemniej w wielu systemach transmisyjnych stosowanych jest więcej poziomów napięciowych. W takich przypadkach (gdy wykorzystywanych jest więcej poziomów sygnału) każdy poziom może reprezentować kilka bitów. Jako przykład rozważmy system, który bazuje na czterech poziomach napięcia: -5 V, -2 V, +2 V oraz +5 V. Każdy z poziomów odpowiada wówczas dwóm bitom, zgodnie z rysunkiem 6.7.



Rysunek 6.7. Sygnały cyfrowe wykorzystujące dwa poziomy napięć (a) i cztery poziomy napięć (b)

Jak nietrudno zauważać na rysunku, zaletą stosowania wielu poziomów sygnału jest możliwość reprezentowania więcej niż jednego bitu w danej jednostce czasu. W przykładzie pokazanym na rysunku 6.7b napięcie -5V odpowiada sekwencji bitowej 00, napięcie -2 V reprezentuje sekwencję 01, +2 V oznacza ciąg 10, a +5 V bity 11. Dzięki zastosowaniu wielu poziomów napięciowych, w każdej szczeblinie czasowej przesyłane są dwa bity, co oznacza, że czteropoziomowa reprezentacja danych (przedstawiona na rysunku 6.7b) pozwala na przekazanie w danym czasie dwukrotnie większej liczby bitów niż dwupozycyjowa reprezentacja (widoczna na rysunku 6.7a).

Zależność między liczbą poziomów a liczbą przekazywanych bitów nie jest szczególnie skomplikowana. Każdemu poziomowi sygnału musi odpowiadać jedna kombinacja bitowa. Ponieważ na  $n$  bitach można zapisać  $2^n$  wartości, do odwzorowania  $n$  bitów system musi wykorzystywać  $2^n$  poziomów.

*System komunikacyjny wykorzystujący dwa poziomy sygnału może przekazywać tylko jeden bit w danym czasie. System obsługujący  $2^n$  poziomów sygnału przesyła  $n$  bitów w jednostce czasu.*

Mogłoby się wydawać, że wartości napięcia są wybierane w dowolnym zakresie, a podział zakresu napięciowego na dowolnie małe podzakresy pozwala na uzyskanie dowolnie dużej

liczby poziomów. Z matematycznego punktu widzenia nic nie stoi na przeszkodzie, aby w przedziale od 0 do 1 V wyznaczyć milion poziomów. Wystarczy przypisać pierwszemu poziomowi napięcie 0,000001 V, drugiemu 0,000002 V itd. Niestety, w praktyce systemy transmisyjne nie są w stanie rozróżnić sygnałów, których wartości nieznacznie odbiegają od siebie. Dlatego w użytkowych rozwiązaniach stosuje się tylko kilka poziomów sygnału.

## 6.11. Body i bity na sekundę

Ille danych można przesyłać w jednostce czasu? Odpowiedź zależy od dwóch parametrów systemu komunikacyjnego. Zgodnie z wcześniejszymi informacjami, szybkość przekazywania danych wynika z liczby poziomów sygnału. Jednak istotny jest również czas podtrzymywania przez system danego poziomu przed przejściem do kolejnego. Na rysunku 6.7a oś pozioma reprezentuje czas. Czas ten podzielono na osiem szczelin, a w każdej szczelinie czasowej transmitowany jest jeden bit. Gdyby system został zmieniony w taki sposób, aby danemu bitowi odpowiadała połowa czasu trwania szczeliny, wówczas w tym samym czasie można by przekazać dwa razy więcej bitów. Na tej podstawie można stwierdzić, że:

*Alternatywna metoda zwiększania ilości danych, które mogą zostać przekazane w określonym czasie, polega na skróceniu czasu, w którym system utrzymuje jeden poziom sygnału.*

Podobnie jak w przypadku liczby poziomów sygnału, również czas jego trwania jest ograniczony przez parametry sprzętowe. Jeśli sygnał nie pozostanie odpowiednio długo na danym poziomie, odbiornik nie będzie mógł poprawnie określić tego poziomu. Co ciekawe, opisując systemy komunikacyjne, nie stosuje się miary czasu trwania pojedynczego poziomu. Przeciwnie, inżynierowie podają liczbę zmian sygnału w sekundzie. Parametr ten jest wyrażanych w **bodach** (ang. *baud*). Na przykład jeśli analizowany mechanizm wymaga utrzymywania sygnału na danym poziomie przez 0,001 s, oznacza to, że pracuje z szybkością 1000 bodów.

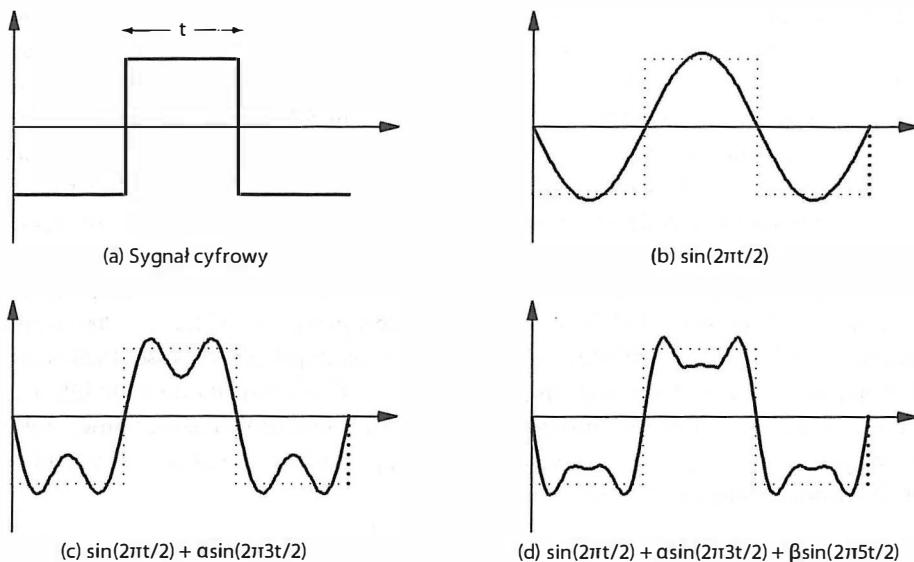
Wartość wyrażona w bodach wraz z liczbą poziomów sygnałów określają przepływność bitową. Jeśli system o dwóch poziomach sygnałów pracuje z szybkością 1000 bodów, może przesyłać dokładnie 1000 bitów w ciągu sekundy. Jeżeli jednak szybkość 1000 bodów odpowiadają cztery poziomy sygnału, szybkość przesyłania wzrasta do 2000 bitów na sekundę (ponieważ cztery poziomy sygnału umożliwiają przesyłanie dwóch bitów). Zależność między szybkością wyrażoną w bodach, liczbą poziomów sygnałów oraz przepływnością bitową opisuje równanie 6.1.

$$\text{bity na sekundę} = \text{body} \times [\log_2(\text{poziomy})] \quad (6.1)$$

## 6.12. Przekształcenie sygnału cyfrowego w sygnał analogowy

W jaki sposób można przekształcić sygnał cyfrowy w odpowiadający mu sygnał analogowy? Zgodnie z teorią Fouriera dowolną krzywą można przedstawić jako zbiór funkcji sinusoidalnych, z których każda ma odpowiednią amplitudę, częstotliwość i fazę. Ponieważ twierdzenie to odnosi się do dowolnych krzywych, ma również zastosowanie w opisie sygnałów cyfrowych. Dla inżynierów teoria Fouriera okazuje się niepraktyczna, ponieważ przedstawienie zgodnie z nią sygnału cyfrowego wymagałoby zastosowania nieskończonie dużej liczby przebiegów sinusoidalnych.

Zastosowano więc rozwiązanie kompromisowe — **aproksymację** sygnału analogowego na podstawie sygnału cyfrowego. Inżynierowie budują więc urządzenia generujące sygnały analogowe, które z dużą dokładnością aproksymują sygnały cyfrowe. Aproksymacja oznacza tworzenie sygnałów zespolonych na bazie jedynie kilku fal sinusoidalnych. Wybierając przebiegi sinusoidalne odpowiadające odpowiednim wielokrotnościom częstotliwości sygnału cyfrowego, można ograniczyć liczbę składowych do trzech. Szczegółowe rozważania na ten temat wykraczają poza zakres tematyczny książki, niemniej na rysunku 6.8 przedstawiona została zasada aproksymacji. Na rysunku widać sygnał cyfrowy (a) oraz jego aproksymacje utworzone dzięki zastosowaniu pojedynczego przebiegu sinusoidalnego (b), sygnału zespolonego złożonego z fali o częstotliwości podstawowej oraz częstotliwości o trzykrotnie większej wartości (c), a także sygnału zespolonego odpowiadającego przebiegowi z rysunku c uzupełnionego o przebieg o częstotliwości pięciokrotnie większej niż częstotliwość podstawowa.



Rysunek 6.8. Aproksymacja sygnału cyfrowego przebiegami sinusoidalnymi

## 6.13. Szerokość pasma sygnału cyfrowego

Jaka jest szerokość pasma sygnału cyfrowego? Jak wiadomo, szerokość pasma sygnału stanowi różnicę między najwyższą i najniższą częstotliwością fal składających się na sygnał zespolony. Zatem jeden ze sposobów obliczenia szerokości pasma polega na zastosowaniu analizy Fouriera do wyznaczenia fal składowych i określenia ich częstotliwości.

Przy zachowaniu matematycznej poprawności poddanie analizie Fouriera przebiegu prostokątnego (takiego jako został pokazany na rysunku 6.8a) spowodowałoby wygenerowanie nieskończonie dużej liczby przebiegów sinusoidalnych. Ponadto w nieskończoność rośłyby również zbiór wartości częstotliwości. Sporządzenie wykresu w dziedzinie czasu oznaczałoby wykreślenie nieskończonie wielu linii wzduż osi poziomej. W rezultacie:

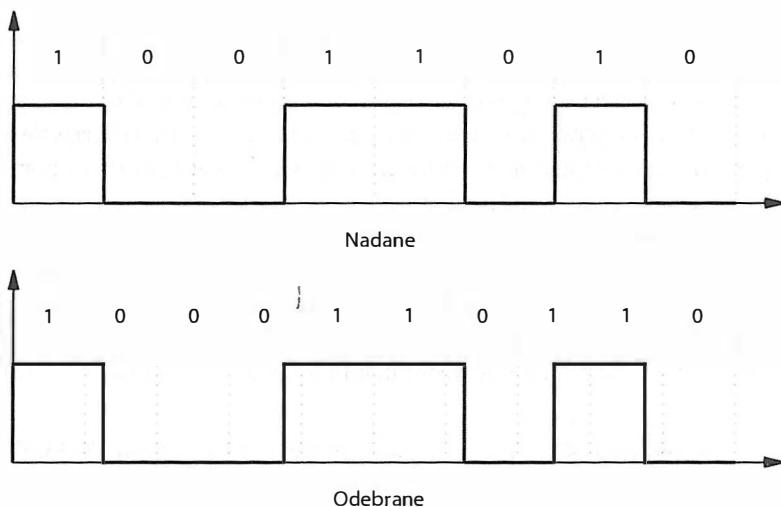
*Zgodnie z definicją szerokości pasma sygnał cyfrowy charakteryzuje się nieskończoną szerokością pasma, ponieważ analiza Fouriera sygnału cyfrowego prowadzi do użycia nieskończonie dużego zbioru fal sinusoidalnych z częstotliwościami o wartościach rosnących w nieskończoność.*

## 6.14. Synchronizacja i uzgodnienia odnośnie sygnałów

Omawiane przykłady nie uwzględniają wielu niuansów, które są istotne w tworzeniu użytkowych systemów komunikacyjnych. Na przykład aby zapewnić jednakową długość czasu na przetwarzanie każdego elementu sygnału po stronie nadajnika i odbiornika, układy elektroniczne obydwu jednostek końcowych muszą dysponować komponentami, które będą precyzyjnie odmierzały czas. Oznacza to, że jeśli jedna ze stron emittuje sygnał, nadając  $10^9$  elementów na sekundę, druga strona musi rejestrować dokładnie  $10^9$  elementów w sekundzie. Zachowanie jednakowych parametrów czasowych przy małych szybkościach transmisyjnych nie stanowi problemu. Jednak opracowanie obwodów elektronicznych, które będą utrzymywać jednakową częstotliwość przetwarzania danych przy bardzo dużych szybkościach transmisyjnych (charakterystycznych dla nowoczesnych sieci) okazuje się niezwykłym wyzwaniem.

Bardziej zasadniczy problem wynika ze sposobu reprezentacji danych w transmitowanych sygnałach. Dotyczy on operacji **synchronizacji** nadajnika z odbiornikiem. Założymy na przykład, że odbiornik pominął pierwszy nadchodzący bit i rozpoczął interpretowanie danych od drugiego lub że odbiornik jest przystosowany do odbierania danych z większą szybkością, niż jest wykorzystywana po stronie nadawczej. Niewłaściwa interpretacja może prowadzić do powstawania błędów. Sytuacja tego typu została zilustrowana na rysunku 6.9. Nadajnik i odbiornik rozpoczynają i kończą pracę w tych samych momentach, ale z uwagi na nieznacznie krótszy czas przypisany poszczególnym bitom po stronie odbiorczej odbiornik błędnie interpretuje sygnał i dostarcza mniej bitów, niż zostało wysłanych.

W praktyce błędy synchronizacji mogą być bardzo trudne do wykrycia. Założymy na przykład, że wyznaczanie czasu po stronie odbiornika jest obarczone błędem powodującym jeden niewłaściwy odczyt na  $10^8$  poprawnych. Błąd może się ujawnić dopiero po przesłaniu



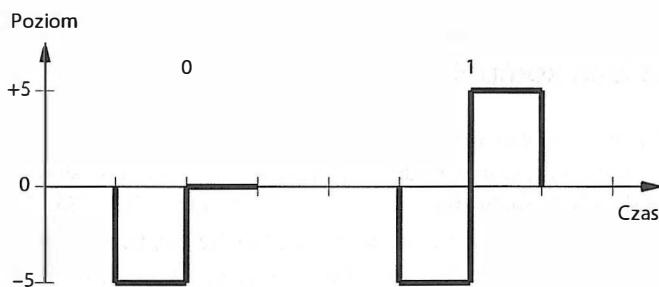
Rysunek 6.9. Błąd synchronizacji wynikający z przydania nieznacznie krótszego czasu poszczególnym bitom po stronie odbiorczej

sekwencji dziesięciu milionów bitów. Ponieważ jednak wysoko wydajne systemy komunikacyjne umożliwiają przesyłanie gigabitów danych w ciągu sekundy, tak małe błędy szybko mogą się okazać bardzo dokuczliwe.

## 6.15. Kodowanie liniowe

Naukowcy opracowali kilka technik eliminowania błędów synchronizacji. Szczególną popularność zyskały dwie z nich. Pierwsze rozwiązanie polega na wysłaniu z nadajnika wzorcowej sekwencji bitowej przed rozpoczęciem nadawania bitów danych. Zazwyczaj sekwencja ta składa się z naprzemiennie ustawionych bitów 0 i 1, które umożliwiają zsynchronizowanie odbiornika. Druga metoda zakłada odwzorowanie danych w sygnale w taki sposób, aby nie było wątpliwości odnośnie ich znaczenia. Sposób reprezentowania danych w sygnale jest określany mianem **kodowania liniowego** (ang. *line coding*).

Jako jeden ze sposobów kodowania liniowego wykluczającego niejednoznaczność można przeanalizować mechanizm transmisyjny, który wykorzystuje trzy poziomy sygnału. Aby zapewnić synchronizację, jeden z tych poziomów jest zarezerwowany na początek każdego bitu. Gdyby trzy wspomniane poziomy odpowiadały napięciu o wartościach  $-5\text{ V}$ ,  $0\text{ V}$  i  $+5\text{ V}$ , napięcie  $-5\text{ V}$  oznaczałoby początek bitu. Zero logiczne reprezentowane byłoby przez sekwencję  $-5\text{ V}$ ,  $0\text{ V}$ . Natomiast odpowiednikiem jedynki logicznej byłaby sekwencja  $-5\text{ V}$ ,  $+5\text{ V}$ . Przy założeniu, że są to jedyne dopuszczalne kombinacje wartości, wystąpienie  $-5\text{ V}$  zawsze oznaczałoby początek bitu, umożliwiając odbiornikowi zsynchronizowanie zegara z nadajnikiem. Technika ta została zilustrowana na rysunku 6.10.



Rysunek 6.10. Przykład użycia dwóch poziomów sygnału do reprezentacji jednego bitu

Oczywiście, zastosowanie większej liczby poziomów sygnału do reprezentacji pojedynczego bitu powoduje zmniejszenie liczby bitów, które można przesyłać w danej jednostce czasu. Dlatego projektanci systemów preferują mechanizmy, w których jeden element sygnału odpowiada większej liczbie bitów (zgodnie z ideą zaprezentowaną na rysunku 6.7b).

W tabeli 6.2 zestawione i pogrupowane zostały nazwy wszystkich powszechnie stosowanych technik kodowania liniowego. Szczegóły implementacyjne poszczególnych rozwiązań wykraczają poza ramy tematyczne książki. Wystarczy zapamiętać, że wybór odpowiedniego zależy od specyfiki określonego systemu komunikacyjnego.

Tabela 6.2. Nazwy powszechnie wykorzystywanych technik kodowania liniowego

Kategoria	Technika	Synchronizacja
Unipolarne	NRZ	Brak w przypadku powielających się bitów 0 lub 1
	NRZ-L	Brak w przypadku powielających się bitów 0 lub 1
	NRZ-I	Brak w przypadku powielających się bitów 0 lub 1
	Dwufazowe	Tak
Bipolarne	AMI	Brak w przypadku powielających się bitów 0
Wielopoziomowe	2B1Q	Brak w przypadku powielających się par bitów
	8B6T	Tak
	4D-PAM5	Tak
	MLT-3	Brak w przypadku powielających się bitów 0

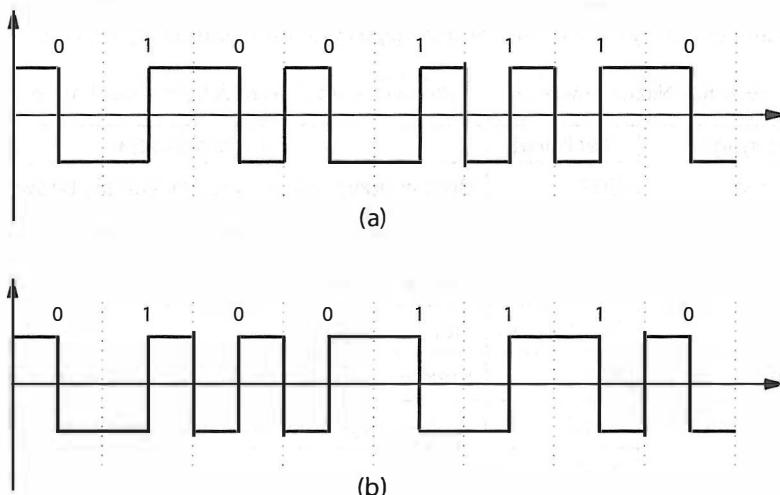
Podsumowując:

Istnieje wiele technik kodowania liniowego, które różnią się sposobem synchronizowania urządzeń, a także innymi parametrami, jak choćby wykorzystywany szerszością pasma.

## 6.16. Wykorzystanie kodowania Manchester w sieciach komputerowych

Wykaz przedstawiony w tabeli 6.2 uzupełnia jeden specjalny standard kodowania liniowego, który ma wyjątkowe znaczenie w budowie sieci komputerowych — stosowane w sieciach Ethernet<sup>18</sup> **kodowanie Manchester**.

Aby zrozumieć zasadę działania kodowania Manchester, trzeba pamiętać, że wykrywanie zmian poziomu sygnału jest znacznie łatwiejsze do zrealizowania niż mierzenie poziomu sygnału. Dlatego w praktycznie stosowanych urządzeniach do opisu bitów służą zmiany poziomu sygnału, a nie same poziomy. Zatem zamiast definiowania, że jedynce logicznej odpowiada określony poziom (na przykład +5 V), w kodowaniu Manchester zapisano, że jedynka logiczna jest reprezentowana przez przejście z zerowego poziomu napięciowego do poziomu powyżej zera. Analogicznie zmiana z wartości powyżej zera na zero oznacza zero logiczne. Ponadto zmiany zachodzą w połowie czasu trwania bitu, co zapewnia powrót do wcześniejszego poziomu w przypadku transmisji dwóch następujących po sobie zer lub jedynek. Technika ta została przedstawiona na rysunku 6.11a.



**Rysunek 6.11.** Kodowanie Manchester (a) i różnicowe kodowanie Manchester (b). W obydwu rozwiązańach zakłada się, że wcześniejszy bit kończy się na niskim poziomie sygnału

Pewną odmianą opisanego rozwiązania jest **różnicowe kodowanie Manchester**, które odzwierciedla względne zmiany, a nie wartości bezwzględne. Reprezentacja bitu zależy w nim od bitu wcześniejszego. W każdej szczerbinie czasowej występuje jedna zmiana lub dwie zmiany. W połowie czasu transmisji bitu **zawsze** następuje zmiana poziomu. Wartość logiczna bitu jest natomiast symbolizowana przez występowanie lub brak zmiany na początku szczerbiny czasowej —零u logicznemu odpowiada zmiana poziomu, natomiast jedynka logiczna nie powoduje zmiany. Różnicowe kodowanie Manchester przedstawiono

<sup>18</sup> Standard Ethernet został omówiony w rozdziale 15.

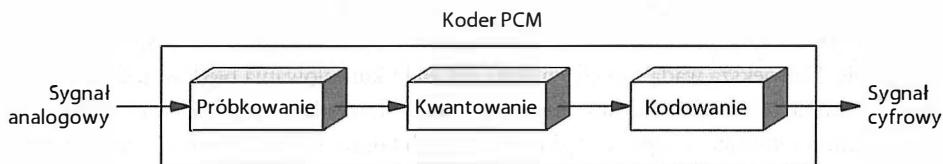
na rysunku 6.11b. Największą zaletą korzystania z kodowania różnicowego jest to, że mechanizm transmisyjny działa poprawnie nawet wtedy, gdy przewody sygnałowe zostanąomyłkowo odwrotnie połączone.

## 6.17. Przekształcenie sygnału analogowego w sygnał cyfrowy

Wiele źródeł danych ma charakter analogowy. Oznacza to, że generowane przez nie sygnały muszą zostać przekształcone do postaci cyfrowej w celu dalszego przetwarzania (na przykład przed zaszyfrowaniem). Istnieją dwa podstawowe rozwiązania tego problemu:

- modulacja impulsowo-kodowa,
- modulacja delta.

Modulacja impulsowo-kodowa (PCM — ang. *Pulse Code Modulation*)<sup>19</sup> jest techniką okresowego mierzenia poziomu sygnału analogowego i przedstawiania wartości pomiaru w formie cyfrowej. Poszczególne etapy konwersji zostały pokazane na rysunku 6.12.



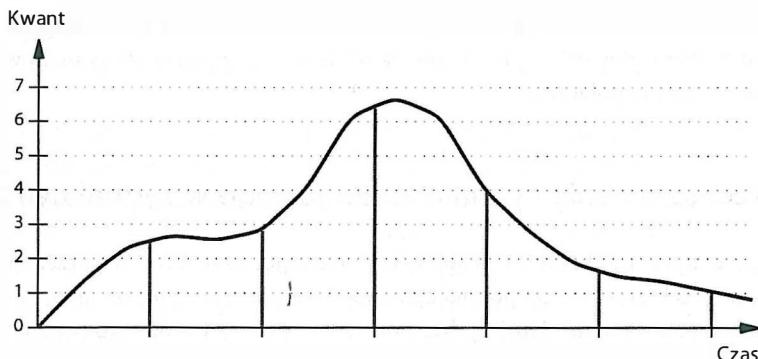
Rysunek 6.12. Trzy etapy modulacji impulsowo-kodowej

Wynik każdego pomiaru jest nazywany **próbką**, co wyjaśnia nazwę pierwszego etapu konwersji (**próbkowanie**). Po zarejestrowaniu próbki jest ona **kwantowana**, czyli zamieniana na liczbę całkowitą o niewielkiej wartości, a następnie **kodowana** w specjalnym formacie. Skwantowana wartość nie jest miarą napięcia ani innej właściwości sygnału. Zakres napięciowy sygnału (od najniższego poziomu od najwyższego) jest odwzorowany na zbiór przedziałów wartości. Zazwyczaj liczba przedziałów jest potęgą dwójki. Zasada kwantowania została zademonstrowana na rysunku 6.13, na którym sygnał został skwantowany na ośmiu przedziałach.

Szare pionowe linie reprezentują na rysunku sześć próbek. Każda z próbek jest kwantowana przez wybór najbliższej wartości kwantu. Na przykład trzecia próbka (pobrana w pobliżu wartości szczytowej) została skwantowana jako wartość 6.

W praktyce stosuje się kilka odmian mechanizmów próbkowania. Na przykład w celu uniknięcia przekłamań związanych z występowaniem chwilowych wzrostów lub spadków poziomu sygnału stosuje się uśrednianie wartości. W takim przypadku zamiast pojedynczej wartości brane są pod uwagę wyniki trzech kolejnych pomiarów, z których wyliczana jest wartość średnia.

<sup>19</sup> Skrót PCM może być nieco mylący, ponieważ odnosi się również do szczególnego wariantu modulacji impulsowo-kodowej, stosowanego w telefonii (zagadnienie to jest tematem jednego z kolejnych rozdziałów).



Rysunek 6.13. Przykład próbkowania i kwantowania stosowany w modulacji impulsowo-kodowej

Najpoważniejszą alternatywą dla modulacji impulsowo-kodowej jest **modulacja delta**. Modulacja delta również wymaga próbkowania sygnału. Jednak zamiast kwantowania każdej próbki, po wysłaniu jednej wartości przekazuje się ciąg wartości reprezentujących różnicę między wcześniejszą a bieżącą wartością sygnału. Technika bazuje na założeniu, że przekazywanie informacji o różnicach w poziomie wymaga użycia mniejszej liczby bitów niż przesyłanie pełnych wartości, co jest prawdą, jeśli sygnał nie zmienia się zbyt gwałtownie. Największą wadą modulacji delta jest efekt kumulowania błędów. Jeśli jeden element z sekwencji zostanie utracony lub przekłamany, wszystkie pozostałe wartości zostaną błędnie zinterpretowane. Z tego względu w systemach komunikacyjnych podatnych na utratę danych w czasie transmisji zazwyczaj stosuje się modulację impulsowo-kodową (PCM).

## 6.18. Twierdzenie Nyquista i częstotliwość próbkowania

Niezależnie od tego, czy wykorzystywana jest modulacja impulsowo-kodowa, czy modulacja delta, sygnał analogowy podlega próbkowaniu. Z jaką częstotliwością należy pobierać próbki sygnału analogowego? Pobranie zbyt małej liczby próbek (nazywane **podpróbkowaniem**) oznacza, że uzyskane wartości cyfrowe będą jedynie niedokładną aproksymacją pierwotnego sygnału. Z kolei pobranie zbyt dużej liczby próbek (tzw. **nadpróbkowanie**) powoduje wygenerowanie nadmiernej ilości danych i zajęcie dodatkowego pasma.

Odpowiedzi na pytanie o liczbę próbek udzielił matematyk o nazwisku Nyquist:

$$\text{częstotliwość próbkowania} = 2 \times f_{\max} \quad (6.2)$$

Czynnik  $f_{\max}$  reprezentuje w równaniu najwyższą częstotliwość sygnału zespolonego. **Twierdzenie Nyquista** stanowi praktyczne rozwiązywanie problemu. Sygnał musi być próbowany z częstotliwością dwukrotnie wyższą niż najwyższa z częstotliwości, która musi zostać poprawnie odwzorowana.

## 6.19. Twierdzenie Nyquista w transmisji telefonicznej

Jako szczególny przykład wykorzystania teorii Nyquista można rozważyć system telefonii, który został pierwotnie zaprojektowany do przenoszenia głosu. Pomiary parametrów ludzkiego głosu wykazały, że zachowanie częstotliwości z przedziału od 0 do 4000 Hz pozwala na odtworzenie go z akceptowlą jakością. Z twierdzenia Nyquista wynika, że przekształcenie sygnału głosowego z analogowego w cyfrowy wymaga rejestracji 8 000 próbek na sekundę.

Aby zapewnić odpowiednią jakość odtwarzanego sygnału głosowego, stosowany w telefonii mechanizm PCM zaprojektowano tak, aby kwantował próbki na ósmiobitowe wartości. Oznacza to, że sygnał wejściowy jest dzielony na 256 poziomów, a każda z próbek może przyjmować wartości z przedziału od 0 do 255. W rezultacie przepływność danych cyfrowych generowanych w ramach pojedynczej rozmowy telefonicznej wynosi:

$$\begin{aligned} \text{cyfrowe połączenie} &= 8000 \text{ próbek/s} \times \\ 8 \text{ bitów/próbkę} &= 64\ 000 \text{ bitów/s} \end{aligned} \quad (6.3)$$

Jak będzie się można przekonać, czytając kolejne rozdziały, systemy telefonii wykorzystują w komunikacji cyfrowej kanały o przepustowości 64 000 bitów na sekundę (64 kb/s). Obwody telefoniczne są również stosowane w internecie do realizacji połączeń długodystansowych.

## 6.20. Kodowanie i kompresja danych

Termin **kompresja danych** jest stosowany do opisu techniki, która redukuje liczbę bitów niezbędnych do reprezentowania danych. Kompresja danych jest niezwykle istotnym elementem systemów komunikacyjnych, ponieważ zmniejszenie liczby bitów skraca jednocześnie czas transmisji. Zatem kompresja danych przed transmisją stanowi sposób optymalizacji systemu komunikacyjnego.

Techniki kompresji stosowane w aplikacjach multimedialnych są tematem rozdziału 29. Na tym etapie istotna jest znajomość dwóch podstawowych rodzajów kompresji:

- Stratnej — część informacji jest tracona podczas kompresji.
- Bezstratnej — wszystkie informacje są zachowane w wersji skompresowanej.

**Kompresja stratna** znajduje zastosowanie przede wszystkim w przekazywaniu danych odbieranych przez człowieka, na przykład w transmisji obrazów, sekwencji wizyjnych lub plików audio. Założenie jest takie, że kompresja musi zagwarantować szczegóły przekazu do poziomu odpowiadającego percepcji człowieka. Zmiana w treści jest dopuszczalna, o ile człowiek nie jest w stanie jej wykryć. W dalszej części książki zostały opisane niektóre powszechnie stosowane mechanizmy kompresji stratnej, takie jak JPEG (przeznaczony do zapisu obrazów) lub MPEG-3 (opisywany skrótnie jako MP3 i stosowany do rejestracji dźwięku).

**Kompresja bezstratna** zachowuje dane wejściowe bez jakiegokolwiek zmiany. Jest więc stosowana przede wszystkim do przechowywania dokumentów oraz w przypadkach,

w których niezbędne jest wierne odwzorowanie danych. W systemach komunikacyjnych nadawca kompresuje dane przed rozpoczęciem transmisji. Natomiast odbiorca dekompresuje wyniki tej operacji. Z uwagi na bezstratność kompresji operacja może zostać wykonana na danych dowolnego rodzaju po stronie nadawcy i pozwala na uzyskanie dokładnej kopii po stronie odbiorcy.

Większość kompresji bezstratnych bazuje na technikach **słownikowych**. Algorytm kompresji wyszukuje powtarzające się ciągi danych i buduje **słownik ciągów**. Kompresje uzyskuje się przez zastępowanie ciągów danych odnośnikiem do pozycji w słowniku. Nadawca musi jednak przesyłać słownik wraz ze skompresowanymi danymi. Jeśli w danych występuje wiele powtarzających się ciągów wartości, połączenie słownika ze skompresowanymi danymi ma mniejszy rozmiar niż pierwotne dane.

## 6.21. Podsumowanie

Źródła informacji mogą dostarczać analogowych lub cyfrowych informacji. Sygnały analogowe mogą mieć charakter okresowy lub nieokresowy. Sygnały okresowe charakteryzują się określona amplitudą, częstotliwością i fazą. Fourier udowodnił, że dowolną krzywą można opisać jako sumę przebiegów sinusoidalnych. Pojedyncze fale sinusoidalne są klasifikowane jako sygnały podstawowe. Natomiast sygnały złożone z wielu fal sinusoidalnych są nazywane sygnałami zespolonymi.

Sygnały zespolone są przez inżynierów reprezentowane na dwa sposoby. Reprezentacja w dziedzinie czasu dostarcza informacji o zmienności sygnału w czasie. Z kolei przedstawienie sygnału w dziedzinie częstotliwości pozwala na ustalenie amplitudy i częstotliwości każdego sygnału składowego. Wykres w dziedzinie częstotliwości jest szczególnie użyteczny, gdy trzeba wyznaczyć szerokość pasma, czyli różnicę między najwyższą i najniższą częstotliwością sygnału.

Liczب zmian sygnału w ciągu sekundy opisuje parametr wyrażany w bodach. Sygnały o wielu poziomach mogą przenosić więcej niż jeden bit informacji na zmianę, zwiększając efektywną przepływność bitową do wartości zależnej od liczby poziomów i szybkości wyrażonej w bodach. Choć szerokość pasma sygnału cyfrowego jest nieskończona, sam sygnał cyfrowy można aproksymować za pomocą trzech przebiegów sinusoidalnych.

Istnieje wiele technik kodowania danych. Do najważniejszych z nich należy zaliczyć kodowanie Manchester, stosowane w sieciach Ethernet. Zaletą techniki Manchester jest to, że zamiast polegać na poziomach sygnałów, do odwzorowania bitów danych wykorzystywane są zmiany poziomu sygnału. Różnicowe kodowanie Manchester bazuje na względnych zmianach poziomu i może działać poprawnie nawet po zamianie przewodów sygnałowych.

Do przekształcania sygnału analogowego w sygnał cyfrowy wykorzystuje się modulację impulsowo-kodową lub modulację delta. Stosowana w telefonii technika PCM wymaga zastosowania 8-bitowej kwantyzacji i pobierania 8000 próbek na sekundę. Wynikiem jest strumień bitowy o przepływności 64 kb/s.

Kompresja danych może mieć charakter stratny lub bezstratny. Kompresja stratna znajduje zastosowanie w przekazywaniu obrazów, dźwięków i sekwencji wizyjnych, które

są odbierane przez ludzi. Pozwala bowiem na wprowadzanie modyfikacji w danych, o ile są one niezauważalne przez człowieka. Kompresja bezstratna nadaje się przede wszystkim do przesyłania dokumentów oraz danych, które muszą być dokładnie odwzorowane.

## ZADANIA

- 6.1. Podaj trzy przykłady źródeł informacji innych niż komputery.
- 6.2. Podaj nazwę typowego urządzenia domowego, które emisuje sygnały nieokresowe.
- 6.3. Dlaczego fale sinusoidalne mają kluczowe znaczenie w transmisji danych?
- 6.4. Wymień i opisz cztery podstawowe parametry przebiegu sinusoidalnego.
- 6.5. Jaki jest najszybszy sposób stwierdzenia, czy faza fali sinusoidalnej wynosi zero, jeśli przebieg ten jest przedstawiony na wykresie?
- 6.6. Jakiego rodzaju fale klasyfikuje się jako **podstawowe**?
- 6.7. Jaki jest wynik analizy Fouriera odnoszącej się do sygnału zespolonego?
- 6.8. Jakie wartości są reprezentowane przez oś pionową na wykresie właściwym dla domeny częstotliwości?
- 6.9. Co to jest analogowa szerokość pasma?
- 6.10. Czy szerokość pasma można łatwiej określić, posługując się wykresem w dziedzinie czasu, czy w dziedzinie częstotliwości?
- 6.11. Założymy, że inżynier zwiększył liczbę poziomów sygnałów z dwóch do czterech. Ile razy większa będzie liczba bitów przesłanych w tym samym czasie? Uzasadnij odpowiedź.
- 6.12. Podaj definicję jednego **boda**.
- 6.13. Dlaczego sygnały analogowe są wykorzystywane do aproksymowania sygnałów cyfrowych?
- 6.14. Jaka jest szerokość pasma sygnału cyfrowego? Wyjaśnij zagadnienie.
- 6.15. Czym jest błąd synchronizacji?
- 6.16. Dlaczego w niektórych technikach kodowania wykorzystuje się wiele elementów sygnału do reprezentowania pojedynczego bitu?
- 6.17. Jaką cechę sygnału wykorzystuje się w kodowaniu Manchester do odwzorowania bitu?
- 6.18. Jaka jest główna zaleta stosowania różnicowego kodowania Manchester?
- 6.19. Jaka operacja następuje po próbkowaniu podczas przekształcania sygnału analogowego w sygnał cyfrowy?
- 6.20. Z jaką częstotliwością musi być próbkowany sygnał pochodzący z mikrofonu, jeśli wiadomo, że ludzkie ucho słyszy dźwięki o maksymalnej częstotliwości 20 000 Hz?
- 6.21. Jaki czas upływa między kolejnymi próbkami w kodowaniu PCM, stosowanym w telefonii?
- 6.22. Opisz różnicę między kompresją strażną i bezstratną i wymień potencjalne obszary ich zastosowania.

# *Zawartość rozdziału*

- 7.1. Wprowadzenie 141
- 7.2. Transmisja przewodowa i bezprzewodowa 141
- 7.3. Podział ze względu na rodzaj energii 142
- 7.4. Zakłócenia elektromagnetyczne i szum 142
- 7.5. Skrętka miedziana 143
- 7.6. Ekranowanie — kabel współosiowy oraz skrętka ekranowana 145
- 7.7. Kategorie skrętek 146
- 7.8. Media przenoszące energię świetlną oraz włókna światłowodowe 146
- 7.9. Rodzaje włókien i transmisji światłowodowych 148
- 7.10. Porównanie włókien światłowodowych i kabli miedzianych 149
- 7.11. Technologie komunikacji w podczerwieni 150
- 7.12. Laserowa komunikacja punkt-punkt 150
- 7.13. Komunikacja z wykorzystaniem fal elektromagnetycznych (radiowa) 151
- 7.14. Propagacja sygnału 152
- 7.15. Rodzaje satelitów 153
- 7.16. Geostacjonarne satelity komunikacyjne 153
- 7.17. Pokrycie obszaru Ziemi przez satelity geostacjonarne 155
- 7.18. Satelity niskoorbitowe i ich klastry 156
- 7.19. Wybór medium transmisyjnego 156
- 7.20. Pomiary parametrów medium transmisyjnego 157
- 7.21. Wpływ szumu na komunikację 157
- 7.22. Znaczenie pojemności kanału 158
- 7.23. Podsumowanie 159

# Media transmisyjne

## 7.1. Wprowadzenie

W rozdziale 5. została omówiona transmisja danych. Tematem poprzedniego rozdziału były źródła informacji, a także sama analogowa i cyfrowa informacja oraz mechanizmy jej kodowania.

W tym rozdziale kontynuowany jest wątek transmisji danych, a dokładniej mediów transmisyjnych, w których skład wchodzą media przewodowe, bezprzewodowe i optyczne. Przedstawiono tutaj podział rodzajów mediów, podstawowe zależności w propagowaniu fal elektromagnetycznych oraz sposoby ekranowania kabli w celu zredukowania lub wyeliminowania zakłóceń i szumów. W końcowej części omówione zostało pojęcie pojemności kanału transmisyjnego. Kolejne zagadnienia związane z transmisją danych są przedstawiane również w następnych rozdziałach.

## 7.2. Transmisja przewodowa i bezprzewodowa

W jaki sposób można skategoryzować media transmisyjne? Istnieją dwie metody podziału:

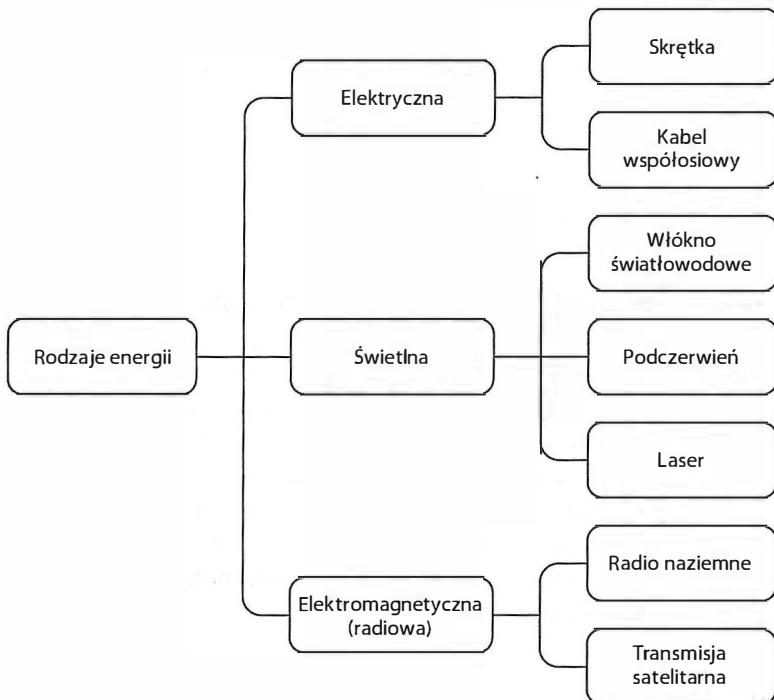
- Na podstawie swobody propagacji — komunikacja może się odbywać w ramach wstępnie ustalonej ścieżki (na przykład w przewodzie) lub bez określonej ścieżki przekazywania danych (jak w transmisji radiowej).
- Na podstawie rodzaju wykorzystywanej energii — w przewodach jest wykorzystywana energia elektryczna, w sieciach bezprzewodowych transmisja radiowa, a w światłowodach transmisja światła.

Do rozróżnienia mediów transmisyjnych wykorzystuje się określenia transmisji **przewodowej i bezprzewodowej**. Pierwsza grupa obejmuje okablowanie miedziane oraz włókna światłowodowe, które prowadzą sygnały wzdułż określonego toru. Natomiast drugie

pojęcie dotyczy transmisji radiowej, w której sygnały są emitowane we wszystkich kierunkach w wolnej przestrzeni. Należy tutaj zwrócić uwagę na fakt, że termin **przewodowe** obejmuje również włókna światłowodowe.

### 7.3. Podział ze względu na rodzaj energii

Podział fizycznych mediów ze względu na formy energii wykorzystywane do transmisji danych został przedstawiony na rysunku 7.1. Każdy rodzaj mediów transmisyjnych jest szczegółowo opisany w kolejnych punktach rozdziału.



Rysunek 7.1. Podział mediów ze względu na rodzaj energii wykorzystanej do transmisji danych

Jak w większości tego typu zestawień, dobór kategorii nie zawsze jest idealny i trzeba pamiętać o istnieniu wyjątków. Na przykład stacje kosmiczne pozostające na orbicie Ziemi mogą korzystać z komunikacji radiowej, którą trudno nazwać naziemną, ale nie jest również satelitarną. Niemniej zaproponowany podział obejmuje większość form komunikacji.

### 7.4. Zakłócenia elektromagnetyczne i szum

Z lekcji fizyki wiadomo, że prąd elektryczny przepływa w zamkniętym obwodzie. Dlatego we wszystkich rozwiązaniach bazujących na transmisji energii elektrycznej potrzebne są dwa przewody, które utworzą obwód — jednym przewodem prąd przepływa do odbior-

nika, a drugim z powrotem do nadajnika. Najprostszy schemat okablowania składa się z jednego kabla, w którym znajdują się dwa przewody miedziane. Każdy przewód jest otoczony plastikową osłoną, która izoluje przewody elektryczne. Zewnętrzna osłona kabla grupuje przewody, co ułatwia instalatorom podłączanie urządzeń.

W sieciach komputerowych stosuje się inne formy okablowania. Aby zrozumieć, dla czego tak jest, trzeba sobie uświadomić trzy fakty:

1. Cała przestrzeń jest wypełniona zmiennym promieniowaniem elektromagnetycznym, nazywanym **szumem**. W praktyce pewna część szumów elektrycznych jest generowana przez systemy komunikacyjne jako efekt uboczny normalnej pracy.
2. Promieniowanie elektromagnetyczne napotykające element metalowy indukuje sygnał o niewielkiej mocy. Oznacza to, że szum może interferować z sygnałami wykorzystywanyymi w komunikacji.
3. Dzięki zdolności do absorbowania promieniowania elementy metalowe działają jak ekrany. Dlatego umieszczenie elementu metalowego między źródłem szumu a medium transmisyjnym może zapobiec zakłócaniu przekazu użytkowego.

Dwa pierwsze stwierdzenia stanowią fundamentalny problem transmisji danych we wszystkich mediach, które wykorzystują energię elektryczną lub elektromagnetyczną. Szczerące trudności w komunikacji występują w pobliżu źródeł promieniowania elektromagnetycznego. Przykładami urządzeń emitujących tego typu zakłócenia są świetłówki i silniki elektryczne. Wyjątkowo uciążliwe są silniki elektryczne o wysokiej mocy, takie, jakie stosuje się w windach, klimatyzatorach i lodówkach. Niemniej źródłami promieniowania niekorzystnie wpływającymi na transmisję danych mogą być również mniejsze urządzenia, takie jak niszczarki do dokumentów lub elektronarzędzia.

Podsumowując:

*Przypadkowe promieniowanie elektromagnetyczne, emitowane przez takie urządzenia jak silniki elektryczne, może zakłócać komunikację bazującą na transmisji radiowej lub przesyłaniu energii elektrycznej w przewodach.*

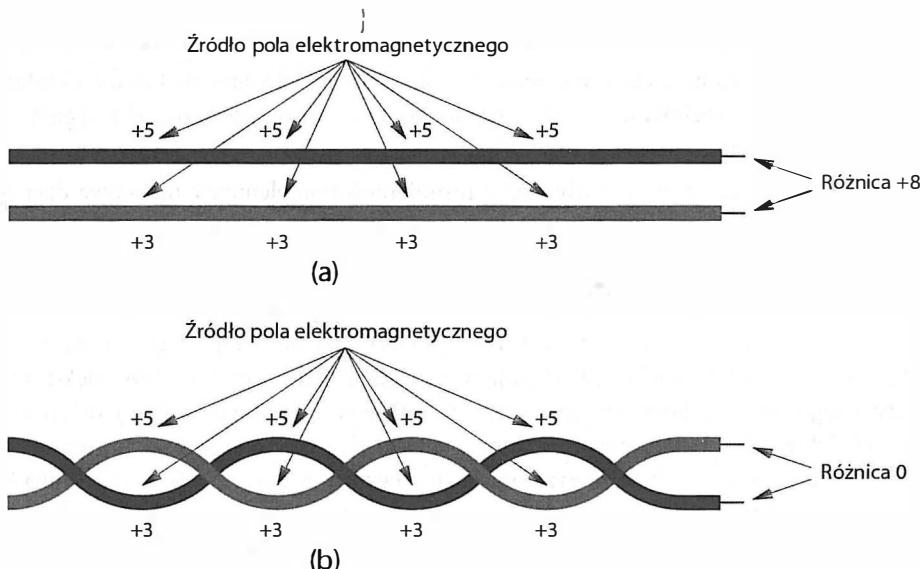
## 7.5. Skrętka miedziana

Trzecie stwierdzenie z wypunktowania zamieszczonego w poprzednim podrozdziale wyjaśnia, dlaczego w obecnych systemach komunikacyjnych stosowane są akurat takie formy okablowania. Istnieją bowiem trzy rodzaje okablowania, które zapewniają ograniczenie wpływu zakłóceń elektrycznych:

- skrętka nieekranowana (UTP — ang. *Unshielded Twisted Pair*);
- kabel współosiowy;
- skrętka ekranowana (STP — ang. *Shielded Twisted Pair*).

Pierwsza z form okablowania — **skrętka** (a dokładniej **skrętka nieekranowana**<sup>20</sup>) — jest powszechnie wykorzystywana w transmisji danych. Jak można wynieść z nazwy, w rozwiązaniu tym kabel składa się z pary przewodów, które są wokół siebie skręcone. Oczywiście, każdy z przewodów jest otoczony materiałem izolacyjnym, który uniemożliwia przepływ prądu z jednego przewodu do drugiego.

Skręcenie dwóch przewodów sprawia, że są one mniej podatne na zakłócenia niż analogiczna para przewodów równoległych. Wyjaśnienie tej zależności znajduje się na rysunku 7.2.



Rysunek 7.2. Pole elektromagnetyczne oddziałujące na dwa równoległe przewody (a) oraz na parę skręconych przewodów (b)

Z rysunku wynika, że w przypadku równoległego ułożenia przewodów jest bardzo prawdopodobne, że jeden z tych przewodów znajdzie się bliżej źródła zakłóceń niż drugi przewód. Ponadto jeden z przewodów pełni wówczas funkcję ekranu, absorbowując promieniowanie elektromagnetyczne. Drugi przewód, ukryty za pierwszym, otrzymuje mniej energii. Zgodnie z rysunkiem w obydwu przypadkach całkowita liczba jednostek wyemitowanego promieniowania wynosi 32 jednostki. W przykładzie 7.2a górny przewód absorbuje 20 jednostek, natomiast do dolnego dociera jedynie 12 jednostek. Powstaje więc różnica wynosząca 8 jednostek. W przykładzie 7.1b każdy z przewodów na takim samym odcinku znajduje się na górze i na dole. Oznacza to, że każdy absorbuje taką samą dawkę promieniowania.

Dlaczego równomierna absorpcja jest tak istotna? Jeśli z powodu zakłóceń w obydwu przewodach zaindukuje się dokładnie tylko samo energię elektryczną, nie nastąpi przepływ dodatkowego prądu. Pierwotny sygnał nie zostanie więc zniekształcony. A zatem:

<sup>20</sup> Ekranowanie zostało opisane w dalszej części rozdziału.

Aby ograniczyć zakłócenia pochodzące z promieniowania elektromagnetycznego, należy stosować skręcone pary przewodów, a nie połączenia równoległe.

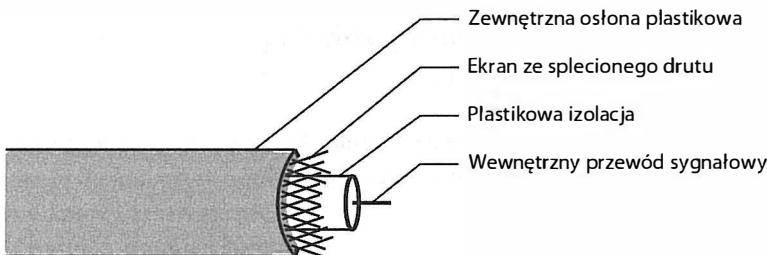
## 7.6. Ekranowanie — kabel współosiowy oraz skrętka ekranowana

Choć skrętki są względnie odporne na zakłócenia generowane przez otoczenie, w niektórych zastosowaniach okazują się zawodne. Ich użycie staje się problematyczne w następujących przypadkach:

- Szum elektryczny jest wyjątkowo silny.
- Kabel jest ułożony wyjątkowo blisko źródła zakłóceń.
- Do komunikacji są wykorzystywane wysokie częstotliwości sygnału.

Jeśli poziom zakłóceń w kablu jest wyjątkowo wysoki (na przykład w fabryce, w której wykorzystuje się urządzenia do spawania elektrycznego) lub kable są ułożone w pobliżu źródła zakłóceń, zastosowanie skrętki bywa niewystarczające. Zakłócenia mogą wystąpić również w przypadku rozłożenia okablowania w podwieszanych sufitach biur w pobliżu świetlówek. Problem staje się jeszcze poważniejszy, gdy do transmisji są wykorzystywane wysokie częstotliwości, albowiem zbudowanie urządzenia, które byłoby w stanie odróżnić sygnały o bardzo wysokich częstotliwościach od szumu, jest niezwykle trudne.

W takich sytuacjach stosuje się inne rodzaje kabli, które zawierają dodatkowy metalowy ekran. Do najczęściej wykorzystywanych kabli tego rodzaju należą kable antenowe — współosiowe. W kablach współosiowych występuje gruby metalowy ekran, utworzony z przepłecionych drutów, który otacza centralny przewód sygnałowy. Budowę takiego kabla przedstawiono na rysunku 7.3.



Rysunek 7.3. Przekrój kabla współosiowego z ekranem otaczającym przewód sygnałowy

Ekran w kablu współosiowym tworzy elastyczny cylinder otaczający przewód wewnętrznego, stanowiąc barierę dla fal elektromagnetycznych. Ponadto ekran uniemożliwia emitowanie energii elektromagnetycznej z przewodu wewnętrznego, która mogłaby oddziaływać na inne przewody. Dzięki temu kable współosiowe mogą być stosowane w pobliżu źródeł zakłóceń elektromagnetycznych, w wiązkach z innymi kablami, a także w rozwiązańach wykorzystujących wysokie częstotliwości sygnałów.

Dokładne ekranowanie i symetria konstrukcji sprawiają, że kable współosiowe nie są podatne na zakłócenia, mogą przenosić sygnały o wysokich częstotliwościach oraz uniemożliwiają przenikanie zakłóceń z danego kabla do innych kabli.

Dzięki zastosowaniu drucianego oplotu zamiast osłony z litego metalu kabel jest giętki. Niemniej ceną dokładnego ekranowania jest mniejsza elastyczność niż w przypadku skrętki. Aby wyeliminować problem niedostatecznej giętkości, opracowano kilka rozwiązań kompromisowych, w których zwiększoną elastyczność kabla, niestety kosztem większej podatności na zakłócenia. Najpopularniejszym z nich jest **skrętka ekranowana** (STP — ang. *Shielded Twisted Pair*). Kabel ten ma znacznie cieńszą i bardziej elastyczną metalową powłokę otaczającą jedną parę przewodów lub większą liczbę par. W większości odmian kabla STP ekran jest wykonany z folii metalowej, podobnej do aluminiowej folii kuchennej. Kable STP są znacznie bardziej giętkie niż kable współosiowe, a jednocześnie mniej podatne na zakłócenia niż **skrętki nieekranowane** (UTP — ang. *Unshielded Twisted Pair*).

## 7.7. Kategorie skrętek

Standard skrętki został początkowo opracowany przez firmy telekomunikacyjne, które wykorzystują tego rodzaju okablowanie w sieciach telefonicznych. Specyfikacja ta została następnie uzupełniona o dokumentację przygotowaną przez trzy organizacje normalizacyjne i dotyczącą okablowania sieci komputerowej. Wspomniane organizacje to American National Standards Institute (ANSI), Telecommunications Industry Association (TIA) oraz Electronic Industries Alliance (EIA). Przygotowane przez nie opracowanie zawiera listę kategorii okablowania z precyzyjną definicją przeznaczenia każdej z nich. Najważniejsze z kategorii przedstawiono w tabeli 7.1.

## 7.8. Media przenoszące energię świetlną oraz włókna światłowodowe

Zgodnie z podziałem przedstawionym na rysunku 7.1 do przenoszenia energii świetlnej są wykorzystywane trzy rodzaje mediów transmisyjnych:

- włókna światłowodowe,
- fale podczerwone,
- lasery.

Najważniejszym rodzajem medium transmisyjnego przenoszącego światło jest **włókno światłowodowe**. Każde włókno składa się z cienkiej nici wykonanej ze szkła lub przezroczystego plastiku, którą otacza plastikowa osłona. Zazwyczaj pojedyncze włókno służy do przekazywania danych w jednym kierunku. Na jednym z jego końców przyłączony jest laser lub dioda LED (emitujące światło). Natomiast drugi koniec jest przyłączony do fotodetektora, który odbiera docierające do niego światło. Do zapewnienia dwukierunkowej

Tabela 7.1. Kategorie skrętki wraz z ich opisem

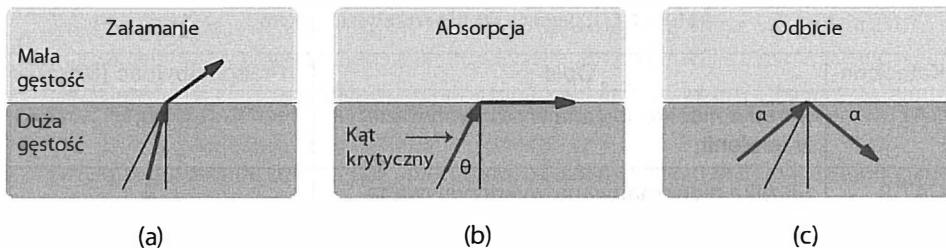
Kategoria	Opis	Przepustowość [Mb/s]
CAT 1	Skrętka nieekranowana wykorzystywana w telefonii	< 0,1
CAT 2	Skrętka nieekranowana wykorzystywana w łączach E1 i T1	4
CAT 3	Udoskonalona skrętka CAT 2 przeznaczona do wykorzystania w sieciach komputerowych	10
CAT 4	Udoskonalona skrętka CAT 3 przeznaczona do wykorzystania w sieciach Token Ring	16
CAT 5	Skrętka nieekranowana wykorzystywana w sieciach komputerowych	100
CAT 5E	Udoskonalona skrętka CAT 5 o zwiększonej odporności na zakłócenia	1000
CAT 6	Skrętka nieekranowana	10 000
CAT 7	Skrętka ekranowana z folią otaczającą cały kabel oraz każdą parę z osobna	10 000

komunikacji niezbędne jest użycie dwóch włókien, z których każde będzie przenosiło dane w jednym kierunku. Z tego względu włókna światłowodowe są często łączone w kable — przez nałożenie na nie plastikowej osłony. Kabel światłowodowy składa się z co najmniej dwóch włókien, choć w przypadku łączenia sieci o dużej liczbie urządzeń często stosuje się kable o większej liczbie włókien.

Włókna światłowodowe nie mogą być zginane pod dowolnym kątem. Niemniej są dostatecznie giętkie, aby można je było układać w łuki o średnicy mniejszej niż 5 cm bez ryzyka uszkodzeń. Rodzi się więc pytanie, jak to jest możliwe, że światło podróżuje w zagętym włókinie? Odpowiedzi na nie udziela fizyka — gdy światło napotka granicę dwóch ośrodków, jego dalsze zachowanie zależy od gęstości obydwu ośrodków oraz kąta padania samego światła. Każdej parze substancji odpowiada jeden **kąt krytyczny** ( $\theta_c$ ), wyznaczany w odniesieniu do linii prostopadłej względem granicy ośrodków. Jeżeli kąt padania światła jest równy kątowi krytycznemu, światło będzie podróżowało wzduż granicy. Jeśli ma mniejszą wartość, promień świetlny przedostanie się przez granicę i zostanie **załamany**. Gdy kąt padania jest większy od kąta  $\theta_c$ , światło zostaje odbite na granicy ośrodków (tak jak w lustrze). Zależności te zostały przedstawione na rysunku 7.4.

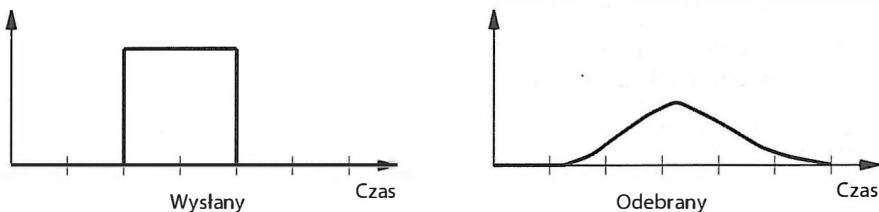
Rysunek 7.4c wyjaśnia przyczynę pozostawiania promieni świetlnych wewnątrz włókna światłowodowego. Warstwa nazywana **płaszczem** wyznacza we włóknie granicę ośrodków, która służy do odbijania światła przesyłanego przez światłowód.

Niestety, odbicie wewnątrz włókna optycznego nie jest idealne i wiąże się z nim częsciowa absorpcja energii. Ponadto występują różnice w odległościach pokonywanych przez foton, które odbijają się nieustannie na granicy rdzenia, oraz foton przemieszczające



Rysunek 7.4. Zachowanie światła na granicy ośrodków o różnych gęstościach, gdy kąt padania jest mniejszy od krytycznego (a), równy krytycznemu (b) oraz większy od krytycznego (c)

się po najkrótszej ścieżce. W rezultacie impuls świetlny wyemitowany po jednej stronie włókna dociera na drugą stronę z mniejszą energią i jest rozciągnięty w czasie (efekt dyspersji). Przypadek ten ilustruje rysunek 7.5.



Rysunek 7.5. Impuls świetlny przesłany przez włókno światłowodowe

## 7.9. Rodzaje włókien i transmisji światłowodowych

Połączenie komputera z innym pobliskim urządzeniem za pomocą kabla światłowodowego nie stanowi jakiegokolwiek problemu. Jednak z uwagi na dyspersję łączenie ze sobą dwóch miast lub kontynentów okazuje się sporym wyzwaniem, które przekłada się na wyższy koszt włókna. Aby zapewnić odbiorcom wybór między wydajnością połączenia a kosztem okablowania, opracowano trzy rodzaje włókien optycznych:

- **Włókna wielomodowe o skokowym współczynniku załamania światła.** Włókno tego typu jest najtańsze i znajduje zastosowanie w systemach, w których wpływ dyspersji nie jest istotny. Granica między rdzeniem włókna a płaszczyzną jest skokowa, co powoduje częste odbijania światła, a tym samym zwiększenie dyspersji.
- **Włókna wielomodowe o gradientowym współczynniku załamania światła.** Światłowód tego typu jest nieznacznie bardziej kosztowny niż światłowód pierwszego typu. Ma jednak tę zaletę, że gęstość materiału w pobliżu krawędzi włókna rośnie stopniowo, co ogranicza odbicia i osłabia efekt dyspersji.
- **Włókno jednomodowe.** Jest włóknem najdroższym, ale jednocześnie charakteryzuje się najmniejszą wartością dyspersji. Ograniczenie dyspersji wynika z bardzo małej średnicy rdzenia światłowodu, a także innych jego parametrów. Światłowody jednomodowe są stosowane w przekazywaniu informacji na dużych odległościach i z bardzo dużymi szybkościami.

Światłowody jednomodowe oraz współpracujące z nimi urządzenia są projektowane w taki sposób, aby skupiały światło. Dzięki temu impulsy świetlne mogą pokonywać tysiące kilometrów bez ryzyka wystąpienia dyspersji. Utrzymanie niewielkich wartości dyspersji umożliwia zwiększenie szybkości transmisji, ponieważ impuls odpowiadający jednemu bitowi nie nakłada się na impuls kolejnego bitu.

W jaki sposób światło jest doprowadzane do światłowodu i z niego odbierane? Najważniejsze jest to, żeby urządzenia transmisyjne pasowały do danego włókna światłowodowego. Do typowych rozwiązań zalicza się:

- nadawanie: dioda LED lub dioda laserowa (ILD — ang. *Injection Laser Diode*);
- odbieranie: element światłoczuły lub fotodioda.

Diody LED i elementy światłoczułe są zazwyczaj stosowane na krótszych odległościach i przy niższych przepływnościach bitowych w połączeniu ze światłowodami wielomodowymi. Światłowody wielomodowe, stosowane na dużych odległościach i w połączeniach o dużych przepływnościach, wymagają zazwyczaj użycia diod IDL i fotodiod.

## 7.10. Porównanie włókien światłowodowych i kabli miedzianych

Włókna światłowodowe mają wiele cech, które sprawiają, że światłowody są bardziej pożądane niż kable miedziane. Włókna są odporniejsze na zakłócenia elektromagnetyczne, zapewniają większą szerokość pasma, a przesyłane w nich światło nie podlega tak silnemu tłumieniu, jak sygnały elektryczne przekazywane w przewodach miedzianych. Okablowanie miedziane jest jednak znacznie tańsze. Ponadto z uwagi na konieczność odpowiedniego przygotowania zakończeń włókna optycznego instalacja kabli miedzianych nie wymaga tak dużych nakładów sprzętowych oraz specjalistycznej wiedzy. Poza tym kable miedziane są mocniejsze. Trudno je przypadkowo przerwać lub załamać. Zestawienie zalet obydwu rodzajów mediów zostało zamieszczone w tabeli 7.2.

Tabela 7.2. Zalety włókien światłowodowych i kabli miedzianych

Włókna światłowodowe
<ul style="list-style-type: none"><li>• Odporność na zakłócenia elektromagnetyczne</li><li>• Mniejsze tłumienie sygnału</li><li>• Szersze pasmo</li></ul>
Kable miedziane
<ul style="list-style-type: none"><li>• Niższa cena</li><li>• Nie wymagają specjalistycznego sprzętu ani fachowej wiedzy</li><li>• Większa odporność na uszkodzenia</li></ul>

## 7.11. Technologie komunikacji w podczerwieni

Technologie komunikacji w podczerwieni (IR — ang. *Infra Red*) bazują na tym samym rodzaju energii, który jest wykorzystywany w pilotach zdalnego sterowania — formie promieniowania elektrycznego, które ma właściwości światła widzialnego, ale obejmuje fale o długościach spoza zakresu pracy ludzkiego oka. Podobnie jak światło widzialne, fale podczerwone rozchodzą się bardzo szybko. Odbijają się od gładkich, twardych powierzchni. Jednak nieprzejrzyste obiekty, nawet tak cienkie, jak kartka papieru, mogą zatrzymać ich rozprzestrzenianie się. Podobny wpływ na rozchodzenie się fal podczerwonych ma wilgoć występująca w atmosferze.

W skrócie:

*Technologie komunikacji z użyciem promieniowania podczerwonego nadają się przede wszystkim do stosowania w pomieszczeniach, w których odległość między nadajnikiem i odbiornikiem jest niewielka, a na trasie między nimi nie ma żadnych przeszkód.*

Najczęstsze zastosowanie technologii transmisji w podczerwieni sprawdza się do połączenia komputera z urządzeniami peryferyjnymi, takimi jak drukarka. Interfejs komputera i interfejs drukarki wysyłają fale podczerwone obejmujące obszar wyznaczony przez kąt 30°. Właściwe ustawienie urządzeń umożliwia wzajemne odbieranie sygnałów. Fakt bezprzewodowej łączności jest szczególnie użyteczny w przypadku laptopów, ponieważ użytkownik może się przemieszczać w pokoju, zachowując dostęp do drukarki. W tabeli 7.3 przedstawiono trzy najczęściej wykorzystywane technologie transmisji w podczerwieni. Każdej pozycji towarzyszy informacja o obsługiwanych przepustowościach.

Tabela 7.3. Trzy technologie transmisji w podczerwieni oraz ich przepustowość

Nazwa	Przeznaczenie	Przepustowość
IrDA-SIR	Wolne połączenia podczerwone	0,115 Mb/s
IrDA-MIR	Połączenia podczerwone o średniej szybkości	1,150 Mb/s
IrDA-FIR	Szybkie połączenia podczerwone	4,000 Mb/s

## 7.12. Laserowa komunikacja punkt-punkt

Opisaną wcześniej technologię transmisji w podczerwieni można sklasyfikować jako połączenia **punkt-punkt**, gdyż do prawidłowego działania systemu konieczne jest ustawienie urządzeń w taki sposób, aby ich nadajniki i odbiorniki pozostawały w bezpośredniej widoczności. Poza tym rozwiązaniem istnieją również inne technologie komunikacji punkt-punkt. Jedną z nich jest komunikacja z użyciem koherentnej wiązki światła, wytwarzanej przez **laser**.

Podobnie jak w przypadku podczerwieni, transmisja z wykorzystaniem laserów wymaga bezpośredniej widoczności urządzeń i braku jakichkolwiek przeszkód na trasie pomiędzy

nimi. Jednak w przeciwieństwie do transmisji w podczerwieni, nadajnik sygnału nie obejmuje swoim działaniem szerszego obszaru. Wiązka światła ma jedynie kilka centymetrów średnicy. Z tego względu urządzenia nadawcze i odbiorcze muszą być bardzo precyjnie ustawiane względem siebie, aby wiązka emitowana z nadajnika trafiała w sensor odbiornika. W praktycznych systemach komunikacyjnych niezbędną jest komunikacja dwukierunkowa. Oznacza to, że po każdej stronie musi funkcjonować nadajnik i odbiornik, a każdy z tych komponentów musi być dokładnie wycelowany w urządzenia znajdujące się po drugiej stronie łącza. Ponieważ dokładność ustawienia ma kluczowe znaczenie dla działania systemu, elementy komunikacji laserowej są zazwyczaj montowane na stałe w danym miejscu.

Wiązki laserowe mają tę zaletę, że można je stosować poza zamkniętymi pomieszczeniami oraz że sprawdzają się w połączeniach na dużych odległościach. Technologia ta jest więc szczególnie użyteczna w miastach do transmisji danych między budynkami. Nietrudno sobie wyobrazić jej zastosowanie w korporacji, która zajmuje biura w dwóch sąsiadujących ze sobą budynkach. Oczywiście, firma nie uzyska zgody na przeciągnięcie kabli pomiędzy tymi budynkami. W takim przypadku wystarczy kupić urządzenia transmisji laserowej i trwale zamocować je na ścianach budynków lub na ich dachach. Po zakupie i zainstalowaniu urządzeń koszt ich użytkowania jest relatywnie niski.

Podsumowując:

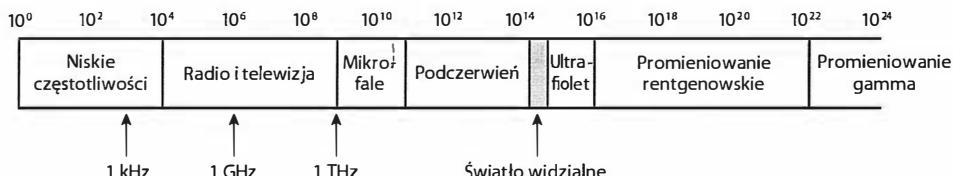
*Technologia laserowa jest wykorzystywana do budowania systemów komunikacji typu punkt-punkt. Nadajnik i odbiornik takiego systemu muszą być precyjnie ustawione względem siebie, ponieważ laser emituje bardzo wąską wiązkę światła. Zazwyczaj instalacja polega na trwałym umocowaniu urządzeń na ścianie lub dachu budynku.*

## 7.13. Komunikacja z wykorzystaniem fal elektromagnetycznych (radiowa)

Zgodnie z wcześniejszymi informacjami technologie **bezprzewodowe** to takie, które do przesyłania energii nie wymagają stałych mediów, na przykład przewodów lub włókien światłowodowych. Najpowszechniejszymi rozwiązaniami komunikacji bezprzewodowej są mechanizmy wykorzystujące energię elektromagnetyczną w zakresie **częstotliwości radiowych** (RF — ang. *Radio Frequency*). Transmisja RF ma jedną zasadniczą przewagę nad transmisją światła — może być realizowana na dużych odległościach i przekonika przez przeszkody, takie jak ściany budynków.

Konkretnie właściwości energii elektromagnetycznej zależą od częstotliwości. Zakres dostępnych częstotliwości często jest określany nazwą **spektrum**. Zakresy dostępnych częstotliwości są natomiast wyznaczane przez rządy poszczególnych państw. W Polsce przydziałem częstotliwości zajmuje się Urząd Komunikacji Elektronicznej. Do zadań tego urzędu należy również określanie mocy nadawczych, które mogą być stosowane w nadajnikach działających na określonych częstotliwościach. Całe spektrum fal elektromagnetycznych

zostało przedstawione na rysunku 7.6. Podziały towarzyszy krótka charakterystyka poszczególnych zakresów. Z rysunku wynika, że część widma częstotliwości obejmuje promieniowanie podczerwone, które zostało opisane w poprzednim podrozdziale. Spektrum odpowiadające komunikacji radiowej rozciąga się od około 3 kHz do 300 GHz i obejmuje częstotliwości przydzielone stacjom radiowym i telewizyjnym, a także nadawcom satelitarnym i operatorom łączności mikrofalowych.



Rysunek 7.6. Najważniejsze zakresy częstotliwości fal elektromagnetycznych.  
Na osi częstotliwości zastosowano skalę logarytmiczną

## 7.14. Propagacja sygnału

W rozdziale 6. zamieszczona została informacja o tym, że ilość informacji, którą może przenieść fala elektromagnetyczna, zależy od częstotliwości tej fali. Częstotliwość wpływa również na sposób **propagacji** fal elektromagnetycznych. Zestawienie trzech zasadniczych rodzajów propagacji znajduje się w tabeli 7.4.

Tabela 7.4. Propagacja fal elektromagnetycznych o różnych częstotliwościach

Kategoria	Zakres	Rodzaj propagacji
Niskie częstotliwości	< 2 Mb/s	Fale rozchodzą się zgodnie z krzywizną Ziemi, ale mogą być blokowane przez nierówności powierzchni.
Średnie częstotliwości	od 2 Mb/s do 30 Mb/s	Fale odbijają się od różnych warstw atmosfery, szczególnie od jonosfery.
Wysokie częstotliwości	> 30 Mb/s	Fale przemieszczają się wzduż linii prostej i są blokowane przez przeszkody.

Z treści zestawienia wynika, że fale elektromagnetyczne o niższych częstotliwościach przemieszczają się przy powierzchni Ziemi. Oznacza to, że w przypadku braku przeszkód terenowych istnieje możliwość umieszczenia odbiornika poza horyzontem widocznym z miejsca usytuowania nadajnika. W przypadku średnich częstotliwości nadajnik i odbiornik mogą być jeszcze bardziej oddalone od siebie, ponieważ emitowany sygnał odbija się w jonosferze. Transmisja radiowa na najwyższych częstotliwościach zachowuje się podobnie do światła. Sygnał propaguje wzduż linii prostej, a przestrzeń między urządzeniami musi być pozbawiona jakichkolwiek przeszkód. A zatem:

Częstotliwości bezprzewodowych technologii sieciowych *nie mogą być dowolnie wybierane, ponieważ przydziałem pasm częstotliwościowych zarządzają organizacje rządowe. Ponadto każda częstotliwość ma pewne szczególne cechy, takie jak sposób propagacji fal radiowych, wymagania odnośnie mocy nadawczej oraz podatność na zakłócenia.*

Technologie bezprzewodowe są klasyfikowane w dwóch ogólnych kategoriach:

- **Transmisja naziemna.** W komunikacji naziemnej wykorzystywane są nadajniki radiowe lub mikrofalowe, które zainstalowano w pobliżu powierzchni Ziemi. Urządzenia i anteny są zazwyczaj instalowane na szczytach wzgórz, masztach antenowych oraz wysokich budynkach.
- **Transmisja spoza Ziemi.** Część urządzeń używanych do komunikacji pracuje poza atmosferą ziemską (na przykład w satelitach krążących po orbitach wokół Ziemi).

Szczegółowe omówienie wybranych technologii bezprzewodowych znajduje się w rozdziale 16. Na razie wystarczą nam informacje o tym, że częstotliwość i poziom mocy wpływają na szybkość transmisji danych, maksymalną odległość między urządzeniami oraz inne cechy połączenia, takie jak zdolność sygnału do przenikania litych obiektów.

## 7.15. Rodzaje satelitów

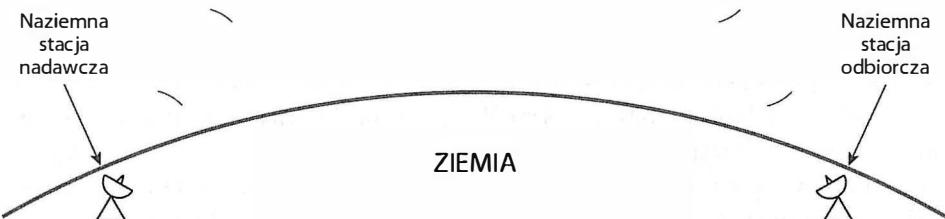
Ruchem obiektów pozostających na orbicie Ziemi (satelitów) rządzi prawa fizyki, a w szczególności **prawo Keplera**. Najważniejszą własnością każdego satelity jest czas obiegu Ziemi, który zależy od odległości obiektu od Ziemi. Parametr ten decyduje o przynależności satelity do jednej z trzech ogólnych kategorii. Wykaz wspomnianych kategorii wraz z krótkim opisem każdej z nich znajduje się w tabeli 7.5.

## 7.16. Geostacjonarne satelity komunikacyjne

Z informacji przedstawionych w tabeli 7.5 wynika, że najważniejszym i najtrudniejszym zadaniem w projektowaniu systemu satelitów komunikacyjnych jest znalezienie kompromisu między wysokością orbity a czasem obiegu Ziemi. Główna zaleta stosowania satelitów na orbicie geostacjonarnej (GEO) polega na tym, że czas, w którym wykonują one pełne okrążenie, pokrywa się dokładnie z czasem obrotu Ziemi. Zatem umieszczenie satelity nad równikiem gwarantuje, że pozostanie on przez cały czas w tym samym miejscu względem powierzchni Ziemi. Stałość pozycji oznacza również to, że po jednorazowym nakierowaniu **stacji naziemnej** na satelitę nie ma potrzeby modyfikowania ustawień sprzętu. Zależność ta została przedstawiona na rysunku 7.7.

Tabela 7.5. Trzy podstawowe kategorie satelitów komunikacyjnych

Orbita	Opis
Niska orbita ziemska (LEO — ang. <i>Low Earth Orbit</i> )	Zaletą satelitów niskoorbitowych jest krótkie opóźnienie w transmisji sygnału. Do wad trzeba jednak zaliczyć to, że z punktu widzenia obserwatora pozostającego na Ziemi satelity przemieszczają się na niebie.
Średnia orbita ziemska (MEO — ang. <i>Medium Earth Orbit</i> )	Satelity średnioorbitowe przemieszczają się po orbitach eliptycznych (a nie kołowych) w celu zapewniania komunikacji na biegunie północnym i biegunie południowym.
Orbita geostacjonarna (GEO — ang. <i>Geostationary Earth Orbit</i> )	Zaletą satelitów geostacjonarnych jest niezmienność położenia w odniesieniu do dowolnego punktu na Ziemi. Wadą jest natomiast duża odległość od powierzchni Ziemi.



Rysunek 7.7. Satelita geostacjonarny i stacja naziemna mają niezmienne położenia względem siebie

Niestety, odległość orbity geostacjonarnej od powierzchni Ziemi wynosi 35 785 km, czyli około jedną dziesiątą drogi między Ziemią a Księżycem. Aby zrozumieć konsekwencje tego faktu dla komunikacji, trzeba sobie uświadomić, że sygnał musi pokonać dystans z Ziemi do satelity i z powrotem. Przy prędkości światła wynoszącej około  $3 \cdot 10^8$  m/s podróż sygnału zajmuje:

$$\frac{2 \cdot 35,8 \cdot 10^6 \text{ m}}{3 \cdot 10^8 \text{ m/s}} = 0,238 \text{ s} \quad (7.1)$$

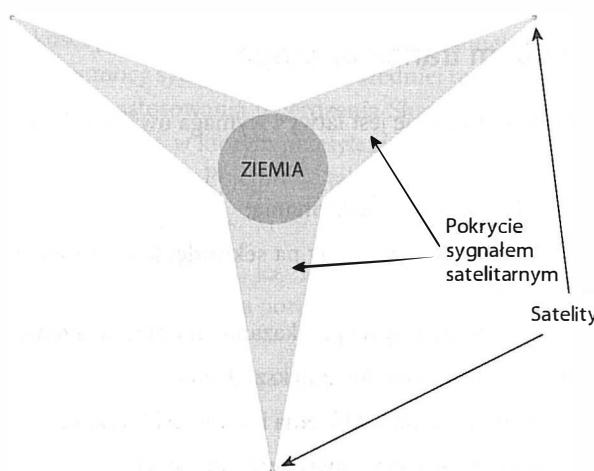
Choć mogłoby się wydawać, że opóźnienie 0,2 s to niedużo, taka wartość bywa bardzo dokuczliwa w niektórych aplikacjach. Na przykład opóźnienie transmisji dźwięku w telefonii lub obrazu w połączeniu wideokonferencyjnym jest wyraźnie zauważalne dla rozmówców. W przypadku transakcji realizowanych drogą elektroniczną, na przykład na giełdzie papierów wartościowych, czas 0,2 s może decydować o zysku lub stracie. Podsumowując:

*Mimo że sygnał radiowy przemieszcza się z prędkością światła, przesłanie go ze stacji naziemnej do satelity geostacjonarnego i z powrotem na Ziemię zajmuje ponad 0,2 s.*

## 7.17. Pokrycie obszaru Ziemi przez satelity geostacjonarne

Ile satelitów geostacjonarnych można umieścić nad Ziemią? Liczba tego typu obiektów jest ograniczona przestrzenią na orbicie geosynchronicznej, ponieważ satelity komunikacyjne pracujące na określonej częstotliwości muszą być od siebie oddzielone, aby nie zakłócały się wzajemnie. Minimalny odstęp zależy od mocy nadawczej, ale jest zazwyczaj definiowany jako kąt o wartości między cztery a osiem stopni. Bez trudu można więc obliczyć, że do pokrycia całego okręgu nad równikiem (360 stopni) wystarczy od 45 do 90 satelitów.

Jaka jest najmniejsza liczba satelitów, które są potrzebne do objęcia zasięgiem całej Ziemi? Trzy. Najlepszym wyjaśnieniem tego zagadnienia jest rysunek 7.8, który przedstawia trzy satelity geostacjonarne rozmieszczone nad równikiem co  $120^\circ$ . Widać na nim, w jaki sposób sygnał z trzech satelitów pokrywa obszar całej Ziemi. Rozmiar Ziemi i odległości do satelitów są narysowane w rzeczywistej skali.



Rysunek 7.8. Trzy satelity geostacjonarne wystarczą do objęcia sygnałem całej powierzchni Ziemi

## 7.18. Satelity niskoorbitowe i ich klastry

Najpoważniejszą alternatywą dla geostacjonarnych satelitów komunikacyjnych są satelity niskoorbitowe (LEO). Ich odległość od Ziemi nie przekracza 2000 kilometrów. Z praktycznych powodów satelita musi być umieszczony na granicy atmosfery ziemskiej, aby nie był hamowany przez gazy atmosferyczne. Z tego względu satelity są wystrzeliwane na orbity o wysokości 500 km lub wyższe. Główną zaletą systemów LEO jest krótkie opóźnienie transmisji sygnału (wynoszące zazwyczaj od 1 do 4 milisekund). Do wad należy jednak zaliczyć to, że czas obiegu Ziemi jest krótszy od czasu jej obrotu. Dlatego z punktu widzenia obserwatora znajdującego się na Ziemi satelity LEO przemieszczają się na niebie. Oznacza to również, że anteny naziemnych stacji nadawczych muszą mieć możliwość śledzenia ruchu satelitów. Samo śledzenie jest dość trudne do wykonania, ponieważ satelity przemieszczają się stosunkowo szybko. Czas obiegu Ziemi na najniższej orbicie LEO wynosi około 90 minut. Na wyższych odpowiada on kilku godzinom.

Satelity LEO są zazwyczaj rozmieszczane w formie **klastrów**, w których grupa urządzeń współdziała ze sobą. Poza komunikacją ze stacjami naziemnymi każde z nich wymienia informacje z innymi elementami grupy. Jako przykład działania systemu przeanalizujmy przypadek, w którym użytkownik w Europie chce przesłać wiadomość do odbiorcy znajdującego się w Ameryce Północnej. Europejska stacja naziemna przekazuje informacje do satelity, który w danym czasie przelatuje nad Europą. Dzięki komunikacji między elementami klastra wiadomość zostaje dostarczona do satelity znajdującego się nad Ameryką Północną, a stamtąd do stacji naziemnej w Ameryce.

*Klastry satelitów LEO współdziałają ze sobą, przekazując wiadomości. Każde z urządzeń grupy musi mieć informację o położeniu pozostałych satelitów względem powierzchni Ziemi i w razie konieczności może dostarczyć dane do satelity, który przekaże je do odpowiedniej stacji nadawczej.*

## 7.19. Wybór medium transmisyjnego

Wybór medium transmisyjnego nie jest łatwy i wymaga uwzględnienia wielu czynników. Oto kilka z nich:

- koszt — materiału, instalacji i utrzymania;
- szybkość transmisji — liczba bitów na sekundę, które mogą być przekazywane w danym medium;
- opóźnienie — czas potrzebny na przekazanie lub przetworzenie sygnału;
- wpływ na sygnał — tłumienie lub zwiększenie;
- środowisko — podatność na zakłócenia i szum elektryczny;
- bezpieczeństwo — łatwość przechwycenia informacji.

## 7.20. Pomiary parametrów medium transmisyjnego

W treści omówienia kilkukrotnie wymienione zostały dwie najważniejsze miary wydajności stosowane w kategoryzowaniu mediów transmisyjnych:

- **opóźnienie propagacyjne** — czas potrzebny na przejście sygnału przez medium;
- **pojemność kanału** — maksymalna ilość danych, które medium może obsłużyć.

Z informacji zamieszczonych w rozdziale 6. wiadomo, że w 1920 roku naukowiec o nazwisku Nyquist opracował fundamentalną zależność między szerokością pasma systemu transmisyjnego a pojemnością tego systemu, znaną jako **twierdzenie Nyquista**. Wyznacza ono w sposób teoretyczny maksymalną szybkość transmisji danych w niezaszumionym kanale. Jeśli w danym systemie transmisyjnym stosowanych jest  $K$  poziomów sygnału, a analogowa szerokość pasma wynosi  $B$ , zgodnie z twierdzeniem Nyquista maksymalną przepustowość łączą  $D$  wyrażoną w bitach na sekundę określa wzór:

$$D = 2 B \log_2 K \quad (7.2)$$

## 7.21. Wpływ szumu na komunikację

Twierdzenie Nyquista wyznacza absolutne maksimum, którego nie można osiągnąć w praktyce. Inżynierowie praktycznie sprawdzili, że z uwagi na występowanie szumu w rzeczywistej wymianie danych niemożliwe jest uzyskanie przepustowości określonej teoretycznie. W roku 1948 Claude Shannon uzupełnił pracę Nyquista, wyznaczając maksymalną przepustowość danych w systemie transmisyjnym, w którym występuje szum. Twierdzenie to, nazywane **twierdzeniem Shannona**<sup>21</sup>, można zapisać w formie następującego wzoru:

$$C = B \log_2(1 + S/N) \quad (7.3)$$

w którym  $C$  jest maksymalną pojemnością kanału wyrażoną w bitach na sekundę,  $B$  odpowiada szerokości pasma systemu, a  $S/N$  opisuje stosunek sygnału do szumu, czyli współczynnik średniej mocy sygnału względem średniej mocy szumu.

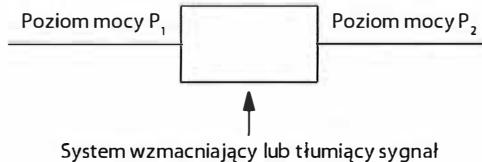
Jako przykład zastosowania twierdzenia Shannona można rozważyć medium transmisyjne o paśmie 1 kHz, w którym przesyłany jest sygnał o mocy 70 jednostek i występuje szum o mocy 10 jednostek. Pojemność takiego systemu wynosi wówczas:

$$C = 10^3 \cdot \log_2(1 + 7) = 10^3 \cdot 3 = 3\,000 \text{ bitów na sekundę}$$

Stosunek sygnału do szumu jest najczęściej wyrażany w **decybelach** (dB). Jednostka ta określa różnicę między dwoma poziomami mocy. Sposób pomiaru wartości mocy przedstawia rysunek 7.9.

---

<sup>21</sup> W niektórych opracowaniach jest ono również nazywane **twierdzeniem Shannona-Hartleya**.



Rysunek 7.9. Pomiar mocy sygnału w dwóch punktach systemu

Po wyznaczeniu poziomów mocy różnicę między nimi (wyrażoną w decybelach) wyznacza się zgodnie z poniższą zależnością:

$$\text{db} = 10 \log_{10} (P_2 / P_1) \quad (7.4)$$

Wykorzystanie decybeli jako jednostki ma dwie interesujące zalety. Po pierwsze, ujemne wartości oznaczają, że sygnał został **tłumiony**, a dodatnie odpowiadają **wzmocnieniu** sygnału. Po drugie, jeśli system komunikacyjny składa się z modułów połączonych kolejno ze sobą, wyznaczenie całkowitej wartości wzmocnienia (tłumienia) sprowadza się do zsumowania wartości decybelowych poszczególnych komponentów systemu.

System telefonii charakteryzuje się stosunkiem sygnału do szumu na poziomie 30 dB i zajmuje pasmo ok. 3000 Hz. Aby przekształcić wartość wyrażoną w decybelach w wartość, którą można wykorzystać do podstawienia do wzoru, wystarczy ją podzielić przez 10, a wynik zapisać jako wykładnik potęgi o podstavie 10 (tj.  $30/10 = 3$ , a  $10^3 = 1000$ , zatem stosunek sygnału do szumu wynosi 1000). Twierdzenie Shannona można wówczas wykorzystać do wyznaczania maksymalnej przepływności bitowej sieci telefonicznej:

$$C = 3000 \cdot \log_2 (1 + 1000)$$

Wynikiem jest 30 000 b/s. Wartość ta stanowi rzeczywisty limit w szybkości transmisji. Uzyskanie wyższej przepustowości byłoby możliwe, gdyby stosunek sygnału do szumu został zwiększyony w jakikolwiek sposób.

## 7.22. Znaczenie pojemności kanału

Twierdzenia Nyquista i Shannona mają bardzo duże znaczenie dla inżynierów projektujących sieci wymiany danych. Praca Nyquista stanowi zachętą do poszukiwania bardziej wydajnych sposobów kodowania bitów w sygnale:

*Twierdzenie Nyquista stanowi wyzwanie dla inżynierów opracowujących mechanizmy kodowania bitów w sygnale, ponieważ zastosowanie zmyślnego kodowania umożliwia przekazanie większej liczby bitów w jednostce czasu.*

Twierdzenie Shannona jest w pewnym sensie ważniejszym opracowaniem, ponieważ definiuje nieprzekraczalną granicę, wyznaczaną przez prawa fizyki. Istotna część szumu rejestrowanego w kanale transmisyjnym może być bowiem związana z promieniowaniem kosmicznym, które jest pozostałością wielkiego wybuchu. Zatem:

*Twierdzenie Shannona informuje inżynierów, że żadne wysiłki nad opracowaniem zmyślnego systemu kodowania nie pokonają praw fizyki, które nakładają ograniczenia na liczbę bitów przekazywanych w ciągu jednej sekundy w rzeczywistym systemie komunikacyjnym.*

## 7.23. Podsumowanie

Inżynierowie mają do dyspozycji wiele mediów transmisyjnych, które można ogólnie sklasyfikować jako przewodowe i bezprzewodowe, w zależności od rodzaju energii stosowanej w transmisji danych (elektrycznej, świetlnej lub radiowej). Energia elektryczna jest stosowana w połączeniach przewodowych. Aby wyeliminować zakłócenia, okablowanie miedziane składa się ze skręconych par przewodów, które mogą zostać owinięte ekranem.

Energia świetlna znajduje zastosowanie w komunikacji punkt-punkt, realizowanej z wykorzystaniem fal podczerwonych lub włókien światłowodowych i laserów. Dzięki odbiciu światła na granicy rdzenia i płaszcza pozostaje ono wewnątrz włókna optycznego. Jednak aby dochodziło do odbicia, kąt padania promienia musi być większy niż kąt krytyczny. Podczas propagowania we włóknie impuls świetlny ulega dyspersji. Problem dyspersji jest znacznie większy w przypadku światłowodów wielomodowych niż w światłowodach jednomodowych. Niestety, światłowody jednomodowe są droższe.

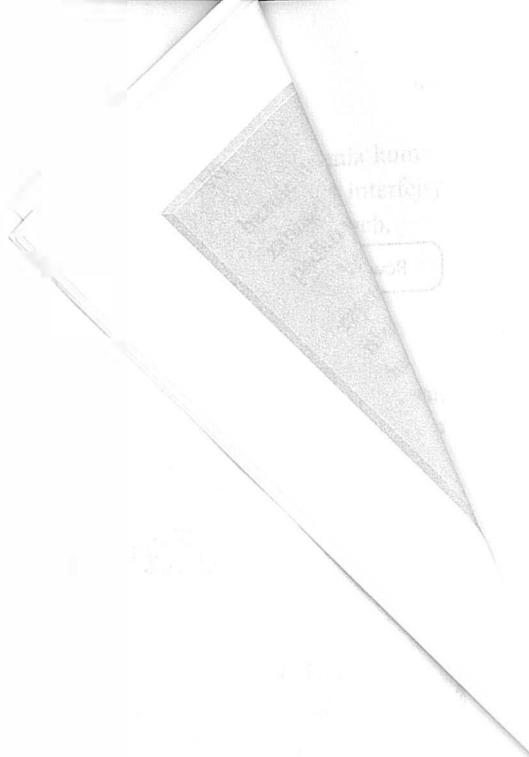
Komunikacja bezprzewodowa bazuje na przekazywaniu energii elektromagnetycznej. Wybrana częstotliwość wpływa na szerokość pasma transmisyjnego oraz na sposób propagowania sygnału. Niskie częstotliwości są charakterystyczne dla fal przyziemnych. Średnie częstotliwości zapewniają odbijanie sygnału w jonasferze. Natomiast wysokie częstotliwości sprawiają, że transmisja przypomina emitowanie światła widzialnego i jest możliwa tylko w przypadku bezpośredniej widoczności nadajnika i odbiornika oraz braku przeszkód na trasie sygnału.

Poza bezpośrednią komunikacją stacji naziemnych transmisja radiowa jest stosowana również w łączności satelitarnej. Orbity geostacjonarne zapewniają rotację satelitów GEO z taką samą prędkością, z jaką obraca się Ziemia. Niestety, duża odległość od powierzchni Ziemi jest przyczyną istotnych opóźnień, mierzonych w dziesiątkach sekundy. Satelity LEO przemieszczają się po niższych orbitach, co z kolei oznacza szybkie przesuwanie się tych obiektów na niebie. Z tego względu do przekazywania informacji stosuje się klastry satelitów LEO.

Teoretyczna pojemność kanału transmisyjnego jest wyznaczana zgodnie z twierdzeniem Nyquista. Odnosi się ona jednak tylko do mediów, w których nie występują szумy. Pojemność rzeczywistego (zaszumionego) kanału opisuje twierdzenie Shannona. Jeden z parametrów równania Shannona (stosunek sygnału do szumu) wyznacza się w decybelach.

## ZADANIA

- 7.1. Jaka jest różnica między transmisją przewodową i bezprzewodową?
- 7.2. Wymień trzy rodzaje energii stanowiące podstawę klasyfikacji fizycznych mediów transmisyjnych.
- 7.3. Co się stanie, gdy szum natrafi na metalowy obiekt?
- 7.4. Wymień trzy rodzaje okablowania stosowane w celu zmniejszenia zakłóceń.
- 7.5. Wyjaśnij, w jaki sposób skręcenie przewodów zmniejsza efekt szumu.
- 7.6. Naszkicuj przekrój kabla współosiowego.
- 7.7. Jakiej kategorii skrętki należałoby użyć podczas rozkładania sieci komputerowej w nowym budynku? Uzasadnij odpowiedź.
- 7.8. Wyjaśnij, dlaczego światło nie opuszcza rdzenia światłowodu na łukach.
- 7.9. Co to jest dyspersja?
- 7.10. Wymień trzy rodzaje włókien światłowodowych i scharakteryzuj każde z nich.
- 7.11. Jakie źródła światła i odbiorniki są stosowane w transmisji światłowodowej?
- 7.12. Jaka jest główna wada stosowania światłowodów w porównaniu do okablowania miedzianego?
- 7.13. Jaki jest średni kąt stożka wyznaczającego obszar oddziaływania nadajnika podczerwieni?
- 7.14. Czy transmisja laserowa może być stosowana w przemieszczających się obiektach? Uzasadnij odpowiedź.
- 7.15. W jaki sposób można wykorzystać fale elektromagnetyczne o niskich częstotliwościach w transmisji danych? Wyjaśnij zagadnienie.
- 7.16. Wymień dwa zasadnicze rodzaje komunikacji bezprzewodowej.
- 7.17. Wymień trzy rodzaje satelitów komunikacyjnych i scharakteryzuj każdy z nich.
- 7.18. Jeśli dane są przesyłane z Europy do Ameryki Północnej za pośrednictwem satelitów GEO, ile czasu potrzeba na dostarczenie żądania i odebranie odpowiedzi?
- 7.19. Ile satelitów GEO jest potrzebnych do objęcia ich zasięgiem całej Ziemi?
- 7.20. Czym jest opóźnienie propagacyjne?
- 7.21. Jaka jest zależność między szerokością pasma, poziomami sygnałów i szybkością transmisji?
- 7.22. Jaka jest przepustowość kabla współosiowego o szerokości pasma 6,2 MHz, jeśli w transmisji są wykorzystywane dwa poziomy sygnału?
- 7.23. Jaka jest efektywna pojemność kanału, w którym średnia moc sygnału wynosi 100, średnia moc szumów wynosi 33,33, a szerokość pasma to 100 MHz?
- 7.24. Jaki jest stosunek mocy (wyrażony w decybelach), jeśli poziom mocy wejściowej systemu wynosi 9000, a wyjściowej 3000?
- 7.25. Ile bitów można przesłać w ciągu sekundy przez sieć telefoniczną, jeśli stosunek sygnału do szumu wynosi 40 dB, a szerokość pasma równa się 3000 Hz?



}

## Zawartość rozdziału

- 8.1. Wprowadzenie 163
- 8.2. Trzy główne przyczyny błędów transmisyjnych 163
- 8.3. Wpływ błędów transmisyjnych na dane 164
- 8.4. Dwie strategie obsługi błędów 165
- 8.5. Kody blokowe i splotowe 166
- 8.6. Przykład kodu blokowego — pojedyncza kontrola parzystości 167
- 8.7. Matematyka kodów blokowych i notacja  $(n,k)$  168
- 8.8. Odległość Hamminga — miara siły kodu 168
- 8.9. Odległość Hamminga między elementami książki kodowej 169
- 8.10. Kompromis między detekcją błędów a narzutem transmisyjnym 170
- 8.11. Korekcja błędów — parzystość wierszy i kolumn 170
- 8.12. 16-bitowa suma kontrolna stosowana w internecie 171
- 8.13. Cykliczny kod nadmiarowy (CRC) 173
- 8.14. Sprzętowa implementacja algorytmu CRC 175
- 8.15. Mechanizmy automatycznego powtarzania żądań (ARQ) 175
- 8.16. Podsumowanie 176

# 8

## *Niezawodność i kodowanie kanałowe*

### **8.1. Wprowadzenie**

Każdy z rozdziałów tej części książki odnosi się do jednego z aspektów wymiany danych, stanowiących podstawę działania wszystkich sieci komputerowych. W poprzednim opisane zostały media transmisyjne oraz problemy związane z szumem elektromagnetycznym. Tematyka tego rozdziału obejmuje błędy powstające w czasie transmisji danych oraz techniki ich wykrywania i usuwania.

Prezentowane tutaj rozwiązania są kluczowe dla funkcjonowania sieci komputerowych i znajdują zastosowanie w protokołach komunikacyjnych wszystkich warstw stosu. Szczegółowe omówienie mechanizmów wykrywania błędów, które zostały zaimplementowane w protokołach internetowych, znajduje się w czwartej części rozdziału.

### **8.2. Trzy główne przyczyny błędów transmisyjnych**

Wszystkie systemy komunikacyjne są podatne na błędy. Część problemów wynika z natury wszechświata, a część jest rezultatem błędów w projektowaniu urządzeń i niedostosowania się do obowiązujących standardów. Wiele pomyłek projektowych można wyeliminować przez gruntowne testowanie rozwiązania, natomiast monitorowanie pracy urządzeń pozwala na wykrycie ich ewentualnych wad fabrycznych. Jednak wykrywanie sporadycznych błędów w trakcie transmisji jest znacznie bardziej skomplikowane niż w przypadku całkowitych uszkodzeń. Dlatego znaczną część zagadnień związanych z sieciami komputerowymi dotyczy problematyki wykrywania i korygowania błędów. Wyróżnia się trzy zasadnicze kategorie błędów transmisyjnych:

1. **Interferencje.** Zgodnie z informacjami zawartymi w rozdziale 7. fale elektromagnetyczne emitowane przez takie urządzenia jak silniki elektryczne, a także promieniowanie kosmiczne mogą zakłócać transmisję radiową, a nawet sygnały przesyłane w łączach przewodowych.
2. **Znieksztalcenia.** Wszystkie systemy fizyczne znieksztalcają sygnały. Impuls świetlny przesyłany w światłowodzie ulega dyspersji. Łącza przewodowe cechują się pewną pojemnością i indukcyjnością, które powodują blokowanie określonych częstotliwości i tłumienie innych. Zwykle umieszczenie przewodu w pobliżu dużych metalowych obiektów może spowodować zmianę zakresu częstotliwości przenoszonych przez ten przewód. Analogicznie, duże metalowe obiekty mogą blokować rozchodzenie się fal o określonych częstotliwościach i nie wpływać na propagację sygnałów o innych częstotliwościach.
3. **Tłumienie.** Wraz z przemieszczaniem się w medium sygnał staje się coraz słabszy. Inżynierowie określają taką sytuację jako **tłumienie**. Sygnały przekazywane w przewodach miedzianych lub włóknach światłowodowych są tym słabsze, im dłuższa jest pokonywana odległość. Podobnie sygnał radiowy słabnie wraz z odległością.

Z twierdzenia Shannona wynika, że jedynym sposobem na zmniejszenie ilości błędów jest zwiększenie stosunku sygnału do szumu (przez podnoszenie mocy sygnału lub obniżanie poziomu szumu). Choć rozwiązań takie jak ekranowanie pozwalają na ograniczenie poziomu szumu, system transmisyjny zawsze pozostanie podatny na błędy, gdyż nie zawsze istnieje możliwość zmiany współczynnika S/N.

Choć całkowite wyeliminowanie błędów transmisyjnych jest nierealne, istnieje możliwość wykrywania problematycznych sytuacji. W niektórych przypadkach błędy mogą być również automatycznie naprawiane. Niestety, wykrywanie błędów wiąże się z pewnym narzutem transmisyjnym. Dlatego detekcja błędów zawsze jest kompromisem projektowym. Twórca systemu musi bowiem podjąć decyzję o tym, czy uzna dany błąd za prawdopodobny, a jeśli tak, to jakie będą konsekwencje jego wystąpienia (na przykład przekłamanie jednego bitu w przelewie bankowym może się przełożyć na miliony złotych strat; z drugiej strony, zmiana jednego bitu w pliku graficznym pozostanie niezauważona dla odbiorcy).

*Błędy transmisyjne są nieuniknione, a mechanizmy ich wykrywania wprowadzają dodatkowy narzut. Z tego względu projektant systemu musi zdecydować, które mechanizmy detekcji i korekcji będą wystarczające w danym rozwiązyaniu.*

### 8.3. Wpływ błędów transmisyjnych na dane

Zamiast zgłębiania fizycznej strony problemu i analizowania przyczyn powstawania błędów osoby zajmujące się transmisją danych skupią się na szacowaniu wpływu błędów na przekazywane informacje. W tabeli 8.1 przedstawiono trzy główne rodzaje błędów występujących w transmisji danych.

Tabela 8.1. Trzy rodzaje błędów występujących w systemach komunikacyjnych

Rodzaj błędu	Opis
Błąd pojedynczego bitu	Zmianie ulega wartość pojedynczego bitu, wszystkie pozostałe bity w transmitowanym bloku pozostają bez zmian (taka sytuacja występuje zazwyczaj przy bardzo krótkich zakłócieniach).
Zbitka błędów	Zmieniane są wartości wielu bitów transmitowanego bloku (co najczęściej jest wynikiem długotrwałych zakłóceń).
Usunięcie (niejednoznaczność)	Sygnal docierający do odbiornika jest niejednoznaczny (nie można ustalić, czy odbierane bity mają wartość 1, czy 0; przyczyną może być zniekształcenie sygnału lub interferencje).

Choć dowolny błąd transmisyjny może spowodować uszkodzenie danych przypisane do poszczególnych kategorii, zazwyczaj określony rodzaj problemów w systemie transmisyjnym przekłada się na pewien rodzaj błędów w danych. Na przykład bardzo krótkie zakłócenie (nazywane **szpilką**) najczęściej powoduje przekłamanie pojedynczego bitu. Dłuższe zakłócenia lub zniekształcenie sygnału mogą doprowadzić do uszkodzenia całej zbitki bitów. Niekiedy również nie można precyzyjnie określić, czy sygnał przenosi bit o wartości 1, czy 0, powodując niejednoznaczność interpretacji.

W przypadku zbitek błędów wyznaczany jest **rozmiar zbitki** lub **długość**, które odpowiadają liczbie bitów pomiędzy początkiem a końcem uszkodzonego fragmentu, zgodnie z rysunkiem 8.1.



Rysunek 8.1. Ilustracja zbitki błędów, w której zmienione bity zostały oznaczone szarym kolorem

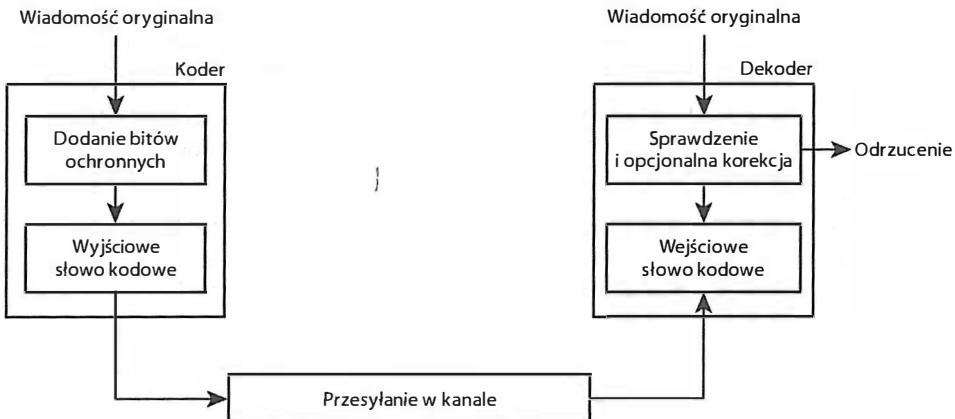
## 8.4. Dwie strategie obsługi błędów

Zwiększenie odporności transmisji na błędy i zwiększenie wiarygodności przekazu stało się przedmiotem wielu opracowań matematycznych, znanych pod ogólną nazwą **kodowania kanałowego**. Techniki te można podzielić na dwie grupy:

- kodowania korekcyjnego (FEC — ang. *Forward Error Correction*);
- automatycznego powtarzania żądania (ARQ — ang. *Automatic Repeat reQuest*).

Koncepcja kodowania korekcyjnego nie jest szczególnie skomplikowana. Srowadza się do uzupełnienia danych o dodatkową informację, która umożliwi odbiornikowi spraw-

dzenie poprawności transmisji i ewentualne naprawienie błędów. Budowa mechanizmu kodowania korekcyjnego została pokazana na rysunku 8.2.



Rysunek 8.2. Teoretyczna budowa mechanizmu kodowania korekcyjnego

Podstawowe **mechanizmy detekcji błędów** umożliwiają odbiornikom wykrywanie przypadków wystąpienia błędu. Natomiast mechanizmy korekcji błędów pozwalają na ustalenie tego, które bity zostały zmienione, oraz na wyznaczenie ich poprawnych wartości. Drugie rozwiązanie w zakresie kodowania kanałowego — mechanizm ARQ<sup>22</sup> — wymaga wymiany między nadajnikiem i odbiornikiem komunikatów, które zagwarantują, że przekazane informacje są poprawne.

## 8.5. Kody blokowe i splotowe

Różne problemy transmisyjne są rozwiązywane przez dwie techniki kodowania korekcyjnego:

- **Kody blokowe.** Działanie techniki kodowania blokowego polega na dzieleniu danych na zbiory bloków, do których następnie dołączana jest pewna **nadmiarowa** informacja. Kodowanie bitów w danym bloku zależy tylko od treści danego bloku (jest niezależne od bitów przesłanych wcześniej). Kody blokowe **nie mają pamięci**, to znaczy nie przenoszą informacji o stanie z jednego bloku do kolejnego.
- **Kody splotowe.** Technika kodowania splotowego polega na wyliczaniu kodu na podstawie całego zbioru danych. Zatem kod wyznaczony dla danego zbioru bitów zależy od zawartości tego zbioru oraz od wcześniejszych bitów strumienia. Kody splotowe są więc kodami **z pamięcią**.

Sprzętowa implementacja kodów splotowych jest trudniejsza niż kodów blokowych, ponieważ wymagają one wykonywania większej ilości obliczeń. Jednak stosowanie kodów splotowych często pozwala na wykrycie większej liczby błędów.

<sup>22</sup> Mechanizm ARQ został opisany w punkcie 8.15.

## 8.6. Przykład kodu blokowego — pojedyncza kontrola parzystości

Aby zrozumieć zasadę wykorzystania nadmiarowych informacji do zabezpieczenia danych przed błędami, rozważmy mechanizm pojedynczej kontroli parzystości (SPC — ang. *Single Parity Check*). Jeden z algorytmów SPC definiuje blok jako 8-bitowy zbiór danych (tj. jeden bajt). Urządzenie działające po stronie nadawczej tuż przed wysłaniem danych dodaje nadmiarowy bit (nazywany **bitem parzystości**). Odbiornik usuwa bit parzystości i wykorzystuje go do sprawdzenia, czy pozostałe bity mają właściwą wartość.

Prawidłowe działanie mechanizmu jest uzależnione od uprzedniego skonfigurowania nadajnika i odbiornika tak, aby obydwa urządzenia sprawdzały **parzystość** lub **nieparzystość** bitów. W przypadku sprawdzania parzystości nadajnik ustawia bit kontrolny na 0, gdy bajt składa się z parzystej liczby bitów 1, oraz na 1, gdy liczba bitów 1 jest nieparzysta. Regułę tę można bardzo łatwo zapamiętać — sprawdzanie parzystości oznacza, że w dziewięciu przesyłanych bitach musi występować parzysta liczba jedynek; z kolei sprawdzenie nieparzystości polega na ustawieniu nieparzystej liczby jedynek w dziewięciu transmитowanych bitach. Przykładowe wartości bajtu danych oraz odpowiadające im bity parzystości i nieparzystości zostały przedstawione w tabeli 8.2.

Tabela 8.2. Bajty danych oraz odpowiadające im bity parzystości i nieparzystości

Dane	Bit parzystości	Bit nieparzystości
00000000	0	1
01011011	1	0
01010101	0	1
11111111	0	1
10000000	1	0
01001001	1	0

Podsumowując:

Pojedyncza kontrola parzystości (SPC) jest podstawową formą kodowania kanałowego, w którym nadajnik dodaje do każdego bajtu nadmiarowy bit o wartości gwarantującej uzyskanie parzystej (lub nieparzystej) liczby jedynek. Odbiornik sprawdza natomiast, czy w nadchodzących danych występuje odpowiednia liczba bitów o wartości 1.

Pojedyncza kontrola parzystości jest mało efektywną wersją kodowania kanałowego. Umożliwia wykrywanie błędów, jednak nie pozwala na ich korygowanie. Ponadto mechanizmy kontroli parzystości sprawdzają się jedynie w przypadku przekłamania nieparzystej liczby bitów. Jeśli jeden z девięciu bitów zostanie zmieniony w czasie transmisji, odbiornik wykryje błąd. Jednak w przypadku wystąpienia dwóch, czterech, sześciu lub ośmiu błędów odbiornik uzna nadchodzący bajt za poprawny.

## 8.7. Matematyka kodów blokowych i notacja $(n,k)$

Mechanizmy kodowania korekcyjnego pobierają jako dane wejściowe zbiór wiadomości i umieszczają w nich dodatkowe bity. W ten sposób generowane są zakodowane wersje wysyłanych wiadomości. Zbiór wszystkich możliwych wiadomości nazywa się  **słowami danych**, natomiast zbiór wszystkich ich zakodowanych wersji określa się jako  **słowa kodowe**. Jeśli słowo danych składa się z  $k$  bitów, a w celu utworzenia słowa kodowego dodano  $r$  bitów, to wynikiem jest

$$\text{schemat kodowania } (n, k)$$

gdzie  $n=k+r$ . Najważniejszym zadaniem podczas projektowania algorytmu wykrywania błędów jest dobór odpowiednich kombinacji słów kodowych (których liczba musi wynosić  $2^n$ ). Podzbiór poprawnych wartości jest nazywany  **książką kodową**.

Jako przykład przeanalizujmy pojedynczą kontrolę parzystości. Zbiór słów danych składa się ze wszystkich wartości, które można zapisać na ośmiu bitach. Zatem  $k=8$ , a liczba słów danych wynosi  $2^8$ , czyli 256. Wysyłane dane składają się z  $n=9$  bitów, więc istnieje  $2^9$  (czyli 512) wszystkich wartości wynikowych. Jednak tylko połowa z nich to prawidłowe słowa kodowe.

Zastanówmy się nad zbiorem wszystkich  $n$ -bitowych wartości oraz podzbiorem wyznaczającym książkę kodową. Jeżeli w czasie transmisji wystąpi jakikolwiek błąd, zmieniony zostanie jeden bit słowa kodowego lub większa liczba bitów słowa. W rezultacie utworzone zostanie kolejne dopuszczalne słowo kodowe lub niedopuszczalna kombinacja bitowa. Na przykład w algorytmie sprawdzania parzystości zmiana pojedynczego bitu prowadzi do powstania niedozwolonej kombinacji bitów. Jednak przekłamanie dwóch bitów skutkuje utworzeniem innego słowa kodowego. Oczywiście, celem każdego projektanta kodu jest opracowanie takiego algorytmu, w którym każdy błąd będzie generował niedozwoloną kombinację bitową. Uogólniając:

*Idealny algorytm kodowania kanałowego to taki, w którym zmiana jakichkolwiek bitów w słowie kodowym prowadzi do powstania słowa o niedozwolonej kombinacji bitów.*

## 8.8. Odległość Hamminga — miara siły kodu

Nie ma idealnych algorytmów kodowania kanałowego. Zawsze zmiana pewnej liczby bitów doprowadzi do przekształcenia jednego słowa kodowego w inne. W praktycznych implementacjach istotne jest więc uzyskanie odpowiedzi na pytanie: „Jaka jest minimalna liczba bitów, które trzeba zmienić, aby powstało inne dozwolone słowo kodowe?”.

Aby odpowiedzieć na to pytanie, inżynierowie posługują się miarą nazywaną  **odlegością Hamminga**. Nazwa pochodzi od nazwiska pracownika Bell Laboratories, który zasłynął z pionierskich opracowań w dziedzinie teorii informacji i kodowania kanałowego. Jeśli dane są dwa ciągi o  $n$  bitach, odległość Hamminga wyraża liczbę zmian (tj. liczbę przekłamanych bitów), których trzeba dokonać, aby przekształcić jeden ciąg bitowy w drugi. Tabela 8.3 jest doskonałą ilustracją do przedstawionej definicji.

**Tabela 8.3.** Przykłady odległości Hamminga dla różnych par 3-bitowych wartości

$d(000, 001) = 1$	$d(000, 101) = 2$
$d(101, 100) = 1$	$d(001, 010) = 2$
$d(110, 001) = 3$	$d(111, 000) = 3$

Jedna z metod wyliczenia odległości Hamminga polega na wyznaczeniu wartości **sumy wyłączającej** (XOR) z dwóch ciągów i zliczeniu bitów o wartości 1 w uzyskanym wyniku. Jako przykład obliczmy odległość Hamminga między ciągami 110 i 011. Wartość sumy wyłączającej wynosi:

$$110 \oplus 011 = 101$$

W wyniku znajdują się dwa bity o wartości 1. Zatem odległość Hamminga między ciągami 011 i 101 wynosi 2.

## 8.9. Odległość Hamminga między elementami książki kodowej

Wróćmy do zasadniczego problemu. Czy błędy mogą doprowadzić do przekształcenia jednego poprawnego słowa kodowego w inne dozwolone słowo kodowe? Aby to ustalić, należy wyznaczyć odległość Hamminga między wszystkimi parami słów kodowych z danej książki kodowej. Zajmijmy się bardzo prostym przykładem, w którym do zakodowania 2-bitowych słów danych zastosowano mechanizm kontroli nieparzystości. W tabeli 8.4 przedstawiono cztery dozwolone słowa danych, cztery poprawne słowa kodowe powstałe po dodaniu bitu nieparzystości oraz wartości odległości Hamminga między poszczególnymi parami słów kodowych.

**Tabela 8.4.** Słowa danych i słowa kodowe algorytmu pojedynczej kontroli parzystości wyznaczone dla 2-bitowych ciągów danych (a) oraz odległości Hamminga dla wszystkich par słów kodowych (b)

Słowo danych	Słowo kodowe		
00	001		
01	010	$d(001, 010) = 2$	$d(010, 100) = 2$
10	100	$d(001, 100) = 2$	$d(010, 111) = 2$
11	111	$d(001, 111) = 2$	$d(100, 111) = 2$

(a)

(b)

Do oznaczenia **minimalnej odległości Hamminga** między parami wartości z książki kodowej służy symbol  $d_{min}$ . Sama operacja ma na celu obliczenie, ile błędnych bitów może doprowadzić do przekształcenia jednego dozwolonego słowa kodowego w inne. W przypadku mechanizmu sprawdzania parzystości (przedstawionego w tabeli 8.4) weryfikowane były wszystkie pary słów kodowych, co doprowadziło do wyznaczenia wartości  $d_{min} = 2$ . Z obliczenia wynika, że istnieje co najmniej jedno dozwolone słowo kodowe,

które można przekształcić w inne dozwolone słowo kodowe przez przekłamanie dwóch bitów w czasie transmisji.

Podsumowując:

*Aby wyznaczyć minimalną liczbę zmian, które spowodują przekształcenie jednego dozwolonego słowa kodowego w inne dozwolone słwo kodowe, należy obliczyć wartość minimalnej odległości Hamminga w odniesieniu do wszystkich par ciągów z książki kodowej.*

## 8.10. Kompromis między detekcją błędów a narzutem transmisyjnym

Niezależnie od rozmiaru zbioru słów kodowych istotne jest, aby wartość  $d_{min}$  była jak największa, gdyż oznacza ona, że kod jest odporny na dużą liczbę błędów — w przypadku wystąpienia mniejszej liczby błędów niż  $d_{min}$  kod zapewni wykrycie nieprawidłowości. Zależność między wartością  $d_{min}$  a maksymalną liczbą przekłamanych bitów, które mogą zostać wykryte ( $e$ ), definiuje równanie 8.1.

$$e = d_{min} - 1 \quad (8.1)$$

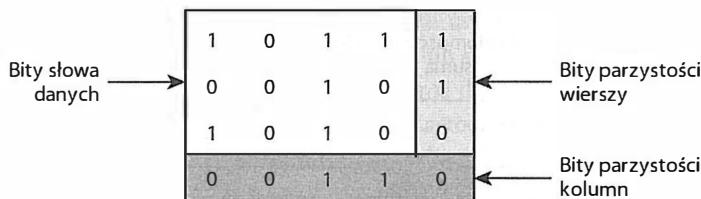
Wybór odpowiedniego kodu jest pewnym kompromisem. Większe wartości  $d_{min}$  gwarantują wykrycie większej liczby błędów. Jednak ceną za to jest konieczność przesyłania większej liczby nadmiarowych informacji niż w przypadku małych wartości  $d_{min}$ . Do określania ilości nadmiarowych danych wykorzystuje się **współczynnik kodu** (ang. *code rate*), który reprezentuje iloraz rozmiaru słowa danych do rozmiaru słowa kodowego. Równanie 8.2 definiuje współczynnik kodu ( $R$ ) w systemie kodowania ( $n, k$ ).

$$R = k/n \quad (8.2)$$

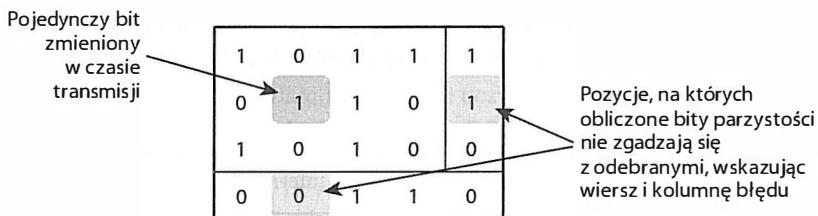
## 8.11. Korekcja błędów — parzystość wierszy i kolumn

W poprzednich punktach przedstawiono mechanizmy detekcji błędów. Tematem tego podrozdziału jest przykład kodu korekcyjnego. Założymy, że słowa danych składają się z  $k=12$  bitów. Zamiast rozpatrywać je jako pojedyncze ciągi, można je zapisać w tablicy złożonej z trzech wierszy i czterech kolumn. Do każdego wiersza i każdej kolumny można następnie dodać bit parzystości. Koncepcję tę ilustruje rysunek 8.3. W wyniku powstaje kod parzystości wierszy i kolumn (RAC — ang. *Row And Column*). W prezentowanym kodowaniu RAC parametr  $n$  ma wartość 20, co oznacza, że algorytm można opisać jako kodowanie (20, 12).

Aby zrozumieć, w jaki sposób działa mechanizm korekcji błędów, przyjmijmy, że jeden z bitów widocznych na rysunku 8.3 został zmieniony w czasie transmisji. Gdy odbiornik zapisze bity w tablicy i wyliczy bity parzystości, dwie wartości nie będą się zgadzały z odczytanymi bitami parzystości. Taki stan pokazano na rysunku 8.4.



Rysunek 8.3. Przykład kodowania wierszy i kolumn z rozmieszczeniem bitów danych w tablicy o wymiarach  $3 \times 4$  i dodaniem bitów parzystości do każdego wiersza i każdej kolumny



Rysunek 8.4. Przykład korekcji pojedynczego błędu w kodowaniu wierszy i kolumn

Jak nietrudno zauważać, pojedynczy błąd powoduje niezgodność dwóch wyliczonych bitów parzystości z bitami odebranymi. Różnice wyznaczają wiersz i kolumnę błędu. Na podstawie obliczonych bitów parzystości odbiornik ustala położenie błędnego bitu danych, a następnie poprawia jego wartość. Mechanizm RAC może więc skorygować dowolny pojedynczy błąd.

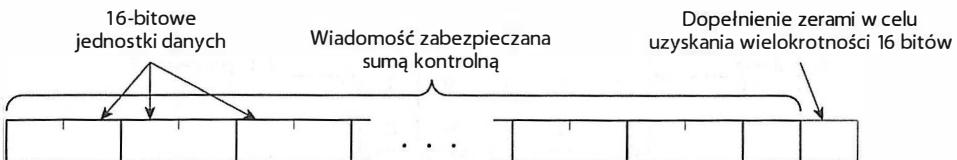
Co się stanie z kodem RAC, gdy w wyniku błędu zmienionych zostanie więcej bitów w bloku? Algorytm RAC umożliwia usuwanie tylko pojedynczych błędów. Niemniej zmiana dwóch lub trzech bitów może zostać wykryta, podobnie jak zmiana dowolnej nieparzystej liczby bitów.

Podsumowując:

*Algorytm parzystości wierszy i kolumn (RAC) umożliwia odbiornikowi korygowanie dowolnych pojedynczych błędów oraz wykrywanie zmian dwóch lub trzech bitów.*

## 8.12. 16-bitowa suma kontrolna stosowana w internecie

Jeden ze schematów kodowania jest szczególnie istotny w transmisjach internetowych. Algorytm ten, nazywany **internetową sumą kontrolną**, generuje kod składający się z 16-bitowej sumy kontrolnej zapisanej w formacie uzupełnień do jedności. W rozwiążaniu tym nie ma wstępnie narzuconych rozmiarów słowa danych. Wiadomość może mieć dowolną długość, a suma kontrolna jest wyznaczana na podstawie całej jej zawartości. W praktyce mechanizm wyznaczania wartości kontrolnej traktuje wiadomość jak zbiór 16-bitowych liczb całkowitych, tak jak to zostało przedstawione na rysunku 8.5.



Rysunek 8.5. Internetowa suma kontrolna — podział danych na 16-bitowe jednostki i dodanie zer dopełniających zbiór danych do wielokrotności 16 bitów

Aby obliczyć sumę kontrolną, nadawca dodaje do siebie wartości 16-bitowych liczb i transmisji wynik. Analogiczne obliczenie jest wykonywane po stronie odbiorczej w celu sprawdzenia wiadomości. Szczegółowy opis działania mechanizmu został przedstawiony w algorytmie 8.1.

**Algorytm 8.1.** Algorytm obliczania 16-bitowej sumy kontrolnej stosowany w wielu protokołach internetowych

Dane:

Wiadomość W o dowolnej długości

Wynik:

16-bitowa uzupełnieniowa suma kontrolna S (uzyskana dzięki 32-bitowym obliczeniom)

Realizacja:

Dopełnienie M zerowymi bitami, aż do uzyskania rozmiaru będącego wielokrotnością 16 bitów.

Przypisanie 32-bitowej sumie kontrolnej C wartości 0.

```
for (każda 16-bitowa grupa w M) {
    Potraktowanie 16 bitów jak liczby całkowitej i dodanie ich do C.
```

}

Wyodrębnienie 16 starszych bitów z C i dodanie ich do C.

Po zanegowaniu 16 młodszych bitów C jest sumą kontrolną.

Jeśli suma kontrolna ma wartość 0, zapisanie zera za pomocą samych jedynek.

Najważniejsze w zrozumieniu zasad działania algorytmu jest uświadomienie sobie, że do obliczenia wartości wynikowej stosuje się arytmetykę uzupełnień do jedności, a nie uzupełnień do dwóch (co jest typowym sposobem działania większości komputerów) oraz że zamiast 32-bitowych lub 64-bitowych liczb całkowitych wykorzystywane są liczby 16-bitowe. Algorytm został więc zapisany w taki sposób, aby wykorzystać 32-bitową arytmetykę uzupełnień do dwóch do obliczenia wartości wyrażonej w kodzie uzupełnień do jedności. W czasie wykonywania pętli `for` może nastąpić przepełnienie. Dlatego po zakończeniu pętli bity przepełnienia (najstarsze bity wyniku) są dodawane do wyniku sumowania.

Dlaczego zamiast wartości sumy przesyłana jest jej zanegowana wartość? W celu zwiększenia wydajności. Odbiornik może zastosować ten sam algorytm, jaki został wykorzystany w nadajniku. Może jednak również uwzględnić wartość samej sumy. Ponieważ kod kontrolny jest negacją wartości sumy, dodanie samej sumy kontrolnej spowoduje uzyskanie w wyniku zera. Odbiornik uwzględnia sumę kontrolną w obliczeniach i sprawdza, czy wynikiem jest zero.

Ostatni etap zadania wynika ze specyfiki kodu uzupełnień do jedności. W kodzie tym zero ma dwie postacie — same zera logiczne lub same jedynki logiczne. W interne-towej sumie kontrolnej wykorzystuje się same jedynki do zasygnalizowania, że suma kontrolna została obliczona i jej wartością jest zero. Zastosowanie samych zer oznaczałoby, że suma kontrolna nie została obliczona.

## 8.13. Cykliczny kod nadmiarowy (CRC)

W sieciach o dużych szybkościach transmisyjnych stosuje się kodowanie kanałowe generujące wartości cyklicznego kodu nadmiarowego (CRC — ang. *Cyclic Redundancy Code*). Kod CRC ma trzy wyjątkowe właściwości, które uczyniły go bardzo ważnym elementem transmisji. Lista tych właściwości widnieje w tabeli 8.5.

**Tabela 8.5.** Trzy cechy kodu CRC, które sprawiają, że jest on nieodzowny w sieciach transmisji danych

Dowolna długość wiadomości	Podobnie jak w przypadku sumy kontrolnej, rozmiar słowa danych nie jest wstępnie ustalony. Oznacza to, że algorytm CRC może być stosowany w odniesieniu do wiadomości o dowolnej długości.
Doskonała skuteczność w wykrywaniu błędów	Ponieważ obliczana wartość zależy od sekwencji bitowych wiadomości, algorytm CRC charakteryzuje się doskonałą sprawnością w wykrywaniu błędów.
Wydajna implementacja sprzętowa	Choć działanie algorytmu wynika z wyrafinowanych przekształceń matematycznych, obliczenie wartości CRC w urządzeniu może być przeprowadzone bardzo szybko.

Określenie **cykliczny** wynika z pewnej właściwości słów kodowych — przesunięcie cykliczne bitów słowa kodowego powoduje wygenerowanie nowego słowa kodowego. W tabeli 8.6 został przedstawiony cykliczny kod nadmiarowy (7, 4) zaproponowany przez Hamminga.

Przez lata nad kodami CRC prowadzono bardzo zaawansowane prace. W ich wyniku powstało wiele teorii matematycznych i technik obliczania wartości kodu. Opisy zagadnień są tak różne, że czasami trudno zrozumieć, że dotyczą tej samej koncepcji. Oto kilka najważniejszych spojrzeń na ten sam problem:

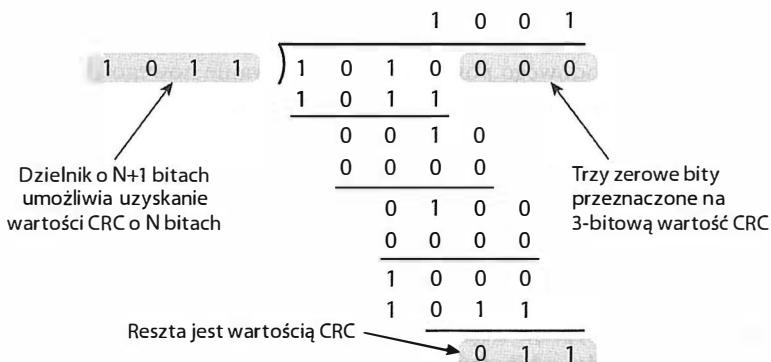
- **Matematycy** opisują algorytm CRC jako sposób na wyznaczenie reszty z dzielenia dwóch wielomianów o binarnych współczynnikach, jednego reprezentującego wiadomość i drugiego odpowiadającego stałemu dzielnikowi.
- **Informatycy** postrzegają algorytm CRC jako obliczenia reszty z dzielenia dwóch liczb binarnych, z których jedna reprezentuje wiadomość, a druga jest stałym dzielnikiem.

Tabela 8.6. Przykład cyklicznego kodu nadmiarowego (7, 4)

Słowo danych	Słowo kodowe	Słowo danych	Słowo kodowe
0000	0000 000	1000	1000 101
0001	0001 011	1001	1001 110
0010	0010 110	1010	1010 011
0011	0011 101	1011	1011 000
0100	0100 111	1100	1100 010
0101	0101 100	1101	1101 001
0110	0110 001	1110	1110 100
0111	0111 010	1111	1111 111

- **Kryptografowie** twierdzą, że obliczenie wartości CRC jest operacją matematyczną w ciele Galois drugiego rzędu — GF(2).
- **Programiści** uznają algorytm CRC za mechanizm iteracyjnego analizowania wiadomości z jednoczesnym odwoływaniem się do tabeli wartości, sumowanych w każdym kolejnym kroku.
- **Projektanci urządzeń** utożsamiają algorytm CRC z niewielkim modułem przetwarzania potokowego, który pobiera sekwencje bitowe i generuje wartości CRC bez konieczności dzielenia lub iteracyjnego przeglądania zbioru wejściowego.

Zastosujmy jedno z podejść, wykorzystując dzielenie liczb całkowitych przy założeniu braku przenoszenia. Na rysunku 8.6 pokazano dzielenie liczby 1010 (reprezentującej wiadomość) przez stałą 1011 (właściwą dla określonego mechanizmu CRC).



Rysunek 8.6. Obliczenie wartości CRC jako reszty z dzielenia binarnego bez przenoszenia

Matematyczne podejście do tego samego zadania sprowadza się do potraktowania go jako dzielenia wielomianów, w których każdy bit liczby binarnej jest odpowiednim współczynnikiem. Na przykład dzielnik 1011 (zastosowany w operacji przedstawionej na rysunku 8.6) można przedstawić jako następujący wielomian:

$$1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^3 + x + 1$$

Z kolei dzielna z rysunku 8.6 (1010000) miałaby następującą reprezentację wielomianową:

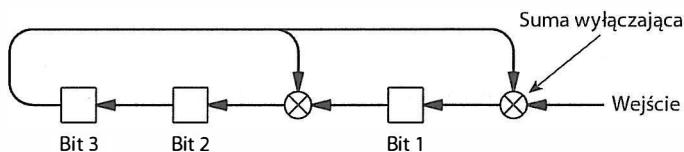
$$x^6 + x^4$$

Wielomian odpowiadający dzielnikowi jest nazywany **wielomianem generującym** kod. Wybór wielomianów generujących jest kluczową operacją podczas prac nad algorytmem CRC o dużej skuteczności wykrywania błędów. Z tego powodu wielomiany generujące kod stały się tematami wielu analiz matematycznych. Wiadomo na przykład, że idealny wielomian nie może być redukowalny (tzn. może być dzielony całkowicie jedynie przez siebie i przez 1) oraz że wielomian o liczbie niezerowych współczynników większej niż jeden pozwala na wykrywanie wszystkich jednabitowych błędów.

## 8.14. Sprzętowa implementacja algorytmu CRC

Sprzęt potrzebny do obliczenia wartości CRC jest wyjątkowo nieskomplikowany. Wystarczy rejestr przesuwny z bramkami XOR (sumy wyłączającej) dołączonymi do niektórych wyjść bitowych. Rejestr przesuwny wykonuje swoje zadanie w takt nadchodzących bitów. Na każdym etapie pobiera bit z poprzedniego etapu lub wykonuje operację XOR i odczytuje jej wynik. Gdy cały ciąg wejściowy zostanie przesunięty przez rejestr, wartość przechowywana w rejestrze jest wynikiem algorytmu CRC.

Komponenty potrzebne do implementacji 3-bitowego algorytmu CRC (przedstawionego wcześniej na rysunku 8.6) zostały pokazane na rysunku 8.7. Zarówno przesuwanie, jak i operacja XOR mogą być wykonywane z bardzo dużą szybkością, więc rozwiązanie to nadaje się do zastosowania w sieciach komputerowych o dużej przepustowości.



Rysunek 8.7. Sprzętowy sposób obliczania 3-bitowej wartości CRC z wielomianu  $x^3+x^1+1$

## 8.15. Mechanizmy automatycznego powtarzania żądań (ARQ)

Zgodnie z zamieszczonymi wcześniej informacjami mechanizm ARQ wymaga od nadajnika i odbiornika przesyłania metainformacji. Oznacza to, że po każdorazowym przesłaniu wiadomości do odległego urządzenia strona odbiorcza musi odesłać komunikat **potwierdzenia**. Na przykład jeśli jednostka A wysyła informację do B, stacja B odsyła potwierdzenie do A. Odebranie potwierdzenia przez jednostkę A stanowi informację o tym, że wiadomość została poprawnie dostarczona. Jeśli w wyznaczonym czasie (T) stacja A nie odbierze potwierdzenia, uznaże, że wiadomość nie dotarła do odbiorcy, i ją **retransmituje**.

Mechanizm ARQ jest szczególnie użyteczny, gdy wykorzystywany system zapewnia detekcję błędów, ale nie uwzględnia ich korekcji. Na przykład wiele sieci komputerowych

bazuje na algorytmie CRC jako mechanizmie wykrywania błędów. Uzupełnienie rozwiązania o mechanizm ARQ daje gwarancję poprawnego dostarczania danych — w przypadku wykrycia błędu odbiorca odrzuca wiadomość, a nadawca ją retransmituje.

W rozdziale 26. zostało omówione działanie protokołu internetowego, który wykorzystuje rozwiązanie ARQ. Poza zademonstrowaniem praktycznego zastosowania mechanizmu przedawniania i retransmisji w rozdziale tym zawarto również omówienie sposobów identyfikacji danych wymagających potwierdzenia oraz czasu oczekiwania przed ponowieniem transmisji.

{}

## 8.16. Podsumowanie

Fizyczne systemy transmisyjne są podatne na zakłócenia, zniększtalcenia sygnału i jego tłumienie. Każdy z tych czynników powoduje błędy w przekazywaniu danych. Rezultatem może być przeklamywanie pojedynczych bitów, zbitek bitów lub całkowite zablokowanie przekazu, z którym mamy do czynienia za każdym razem, gdy docierający sygnał nie jest jednoznaczny (nie można kategorycznie stwierdzić, że odbierany bit to 1 lub 0). Wymiana danych wymaga od systemów komunikacyjnych stosowania mechanizmów kodowania korekcyjnego lub automatycznego powtarzania żądań (ARQ).

Kodowanie korekcyjne polega na dodawaniu po stronie nadawczej nadmiarowych bitów, które służą do kodowania danych przed przesaniem ich w kanale. Dodatkowe bity są usuwane po stronie odbiorczej w procesie dekodowania informacji. Schemat kodowania ( $n, k$ ) uwzględnia  $k$ -bitowe słowa danych oraz  $n$ -bitowe słowa kodowe.

Jednym ze sposobów szacowania jakości algorytmów kodowania jest obliczenie prawdopodobieństwa, że zmiana bitów poprawnego słowa kodowego spowoduje utworzenie innego poprawnego słowa kodowego. Dokładną miarą tego typu problemu jest minimalna odległość Hamminga.

Niezbyt skomplikowane kody blokowe, takie jak kody parzystości, umożliwiają wykrywanie nieparzystej liczby błędów, ale nie zabezpieczają transmisji przed zmianą parzystej liczby bitów. Algorytm parzystości wierszy i kolumn (RAC) zapewnia korekcję jednego błędu, wykrycie do trzech błędów w bloku, a także każdego innego błędu, który wynika ze zmiany nieparzystej liczby bitów.

Sosowaną w internecie sumę kontrolną można wykorzystać do ochrony wiadomości o dowolnym rozmiarze. Działanie algorytmu polega na dzieleniu przesyłanych informacji na 16-bitowe bloki, obliczeniu zanegowanej wartości sumy tych bloków (w systemie uzupełnienia do jedności) i dodaniu bitów przepełnienia do wynikowej sumy kontrolnej.

W sieciach o dużych przepustowościach stosowane są cykliczne kody nadmiarowe (CRC). Ich najważniejszą zaletą jest to, że działają na wiadomościach o dowolnym rozmiarze, gwarantują efektywne wykrywanie błędów i nadają się do łatwego zaimplementowania sprzętowego. Algorytmy CRC są intensywnie badane przez matematyków. Działanie algorytmu można przedstawić jako sposób obliczania reszty z dzielenia binarnego, sposób obliczania reszty z dzielenia wielomianów lub jako operację w teorii Galois. Sprzętowe generowanie wartości CRC sprowadza się do wykorzystania rejestrów przesuwanych oraz bramek sumy wyłączającej.

## ZADANIA

- 8.1. Wymień i scharakteryzuj trzy główne źródła zakłóceń.
- 8.2. W jaki sposób błędy transmisyjne oddziałują na dane?
- 8.3. W jaki sposób mierzy się długość zbitki w błędach zbitkowych?
- 8.4. Czym jest słowo kodowe i jak można je wykorzystać w korekcji błędów?
- 8.5. Podaj przykład kodu blokowego stosowanego w danych znakowych.
- 8.6. Jaki jest efekt działania doskonałego mechanizmu kodowania kanałowego?
- 8.7. Zdefiniuj **odległość Hamminga**.
- 8.8. Oblicz odległość Hamminga w następujących parach ciągów bitowych: (0000, 0001), (0101, 0001), (1111, 1001) oraz (0001, 1110).
- 8.9. W jaki sposób wyznacza się minimalną liczbę zmian bitowych, która powoduje przekształcenie jednego dozwolonego słowa kodowego w inne słowo?
- 8.10. Wyjaśnij pojęcie **współczynnika kodu**. Czy korzystniejsza jest mała wartość współczynnika, czy duża?
- 8.11. Utwórz tablicę parzystości RAC dla schematu kodowania (20, 10) i słowa danych 100011011111.
- 8.12. Jaka jest przewaga mechanizmu RAC nad pojedynczą kontrolą parzystości?
- 8.13. Napisz program komputerowy, który obliczy 16-bitową internetową sumę kontrolną.
- 8.14. Wymień cechy charakterystyczne algorytmu CRC.
- 8.15. Zapisz dzielenie wartości 10010101010 przez 10101.
- 8.16. Zapisz dwie przedstawione powyżej wartości w formie wielomianów.
- 8.17. Napisz program komputerowy, który będzie generował wartości CRC zgodnie ze schematem (7, 4) przedstawionym w tabeli 8.6.
- 8.18. Wymień i opisz funkcje dwóch bloków urządzenia wykonującego obliczenia CRC.

# Zawartość rozdziału

- 9.1. Wprowadzenie 179
- 9.2. Podział trybów transmisji danych 179
- 9.3. Transmisja równoległa 180
- 9.4. Transmisja szeregową 181
- 9.5. Kolejność wysyłania bitów i bajtów 182
- 9.6. Zależności czasowe w transmisji szeregowej 182
- 9.7. Transmisja asynchroniczna 183
- 9.8. Asynchroniczna transmisja znaków — RS-232 183
- 9.9. Transmisja synchroniczna 184
- 9.10. Bajty, bloki i ramki 185
- 9.11. Transmisja izochroniczna 186
- 9.12. Simpleks, półsimpleks i dupleks 186
- 9.13. Urządzenia DCE i DTE 187
- 9.14. Podsumowanie 188

# 9

## *Tryby transmisji danych*

### 9.1. Wprowadzenie

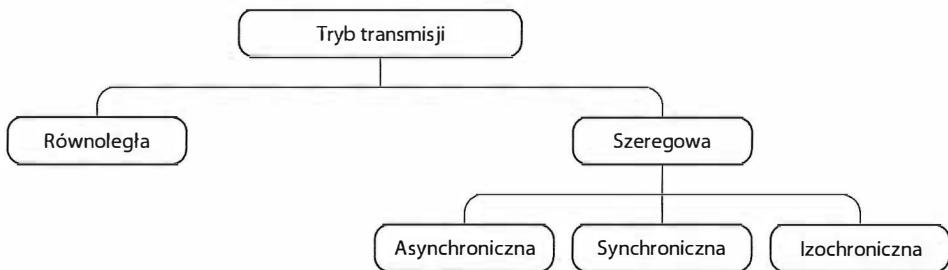
W poszczególnych rozdziałach tej części książki opisane zostały podstawowe rozwiązania z dziedziny transmisji danych. Tematem tego rozdziału są sposoby przekazywania danych. Wprowadzono tutaj powszechnie stosowaną terminologię z zakresu transmisji, przedstawiono zalety i wady równoległego przesyłania danych i omówiono koncepcje komunikacji synchronicznej i asynchronicznej. Opisane tutaj rozwiązania są stosowane w sieciach internetowych, których prezentacja znajduje się w dalszych rozdziałach książki.

### 9.2. Podział trybów transmisji danych

Termin **tryb transmisji** odnosi się do sposobu, w jaki dane są przekazywane za pośrednictwem medium transmisyjnego. Zasadniczy podział trybów transmisji sprowadza się do przypisywania ich do dwóch podstawowych kategorii:

- Szeregowe — w danym czasie przesyłany jest tylko jeden bit.
- Równoległe — w danym czasie przesyłanych jest kilka bitów.

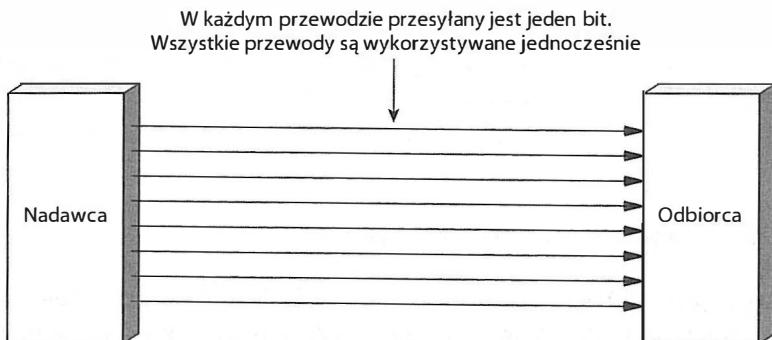
Transmisja szeregowa podlega dalszej kategoryzacji, uwzględniającej zależności czasowe. Ogólne drzewo trybów transmisji danych zostało przedstawione na rysunku 9.1.



Rysunek 9.1. Podział trybów transmisji danych

### 9.3. Transmisja równoległa

Określenie **transmisja równoległa** odnosi się do mechanizmu, który w tym samym czasie przesyła kilka bitów danych za pomocą oddzielnych mediów transmisyjnych. Zazwyczaj oznacza to przesyłanie danych za pośrednictwem kilku przewodów. Sygnały przekazywane przez każdy z przewodów są ze sobą zsynchronizowane tak, aby poszczególne bity były transmitowane we wszystkich przewodach w dokładnie tym samym czasie. Zasada działania mechanizmu została przedstawiona na rysunku 9.2. Z rysunku jasno wynika również to, dlaczego inżynierowie używają określenia **równolegle** do opisu połączenia.



Rysunek 9.2. Transmisja równoległa wykorzystująca 8 przewodów do przesłania 8 bitów

Na rysunku pominięto dwa istotne szczegóły. Po pierwsze, poza równoległymi liniami danych interfejsy równoległe zazwyczaj uwzględniają przewody, które sterują pracą nadajnika i odbiornika. Po drugie, w celu usprawnienia instalacji i utrzymania połączeń systemy równoległej transmisji danych zazwyczaj wykorzystują pojedyncze kable. Do nadajnika i odbiornika są więc przyłączane pojedyncze kable o większym przekroju, a nie poszczególne przewody.

Transmisja równoległa ma dwie zasadnicze zalety:

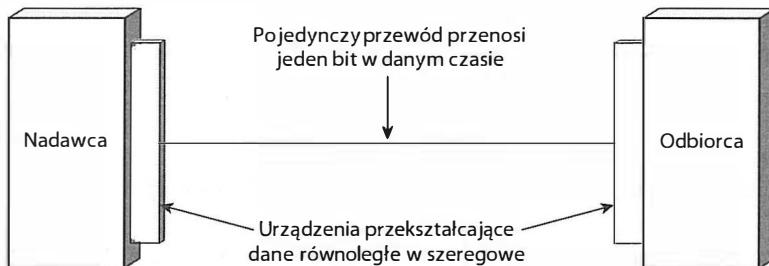
- **Wysoką przepustowość.** Dzięki możliwości przekazywania N bitów w jednostce czasu interfejs równoległy może pracować N razy szybciej niż odpowiadający mu interfejs szeregowy.

- **Dopasowanie do urządzenia końcowego.** Komputery i inne urządzenia komunikacyjne wykorzystują wewnętrznie obwody równoległe. Równoległe interfejsy wyjściowe doskonale pasują więc do wewnętrznej budowy stacji końcowych.

## 9.4. Transmisja szeregowa

Rozwiązaniem konkurencyjnym dla transmisji równoległej jest **transmisja szeregowa**, która zakłada przesyłanie jednego bitu w danym czasie. Biorąc pod uwagę dążenie do uzyskania wysokich przepustowości, wydawałoby się, że każdy projektant systemu wymiany danych zdecyduje się na zastosowanie transmisji równoległej. W praktyce jednak większość rozwiązań bazuje na komunikacji szeregowej. Są ku temu dwa powody. Po pierwsze, sieciami szeregowymi można obejmować większe obszary przy niższym koszcie (wymaganych jest mniej przewodów, a poza tym urządzenia pośredniczące są tańsze). Po drugie, użycie tylko jednego przewodu oznacza, że nigdy nie wystąpi problem dopasowania czasowego sygnałów, wynikający z tego, że jeden z przewodów jest nieznacznie dłuższy od drugiego (w systemach o dużych przepływnościach milimetrowe różnice w długości mają istotne znaczenie dla transmisji danych).

W rozwiązaniach szeregowych nadajnik i odbiornik muszą zawierać komponenty odpowiedzialne za przekształcenie danych z postaci równoległej (właściwej dla stacji końcowych) w szeregową (stosowaną podczas przesyłania danych w przewodzie). Budowę systemu przedstawiono na rysunku 9.3.



Rysunek 9.3. Transmisja danych w trybie szeregowym

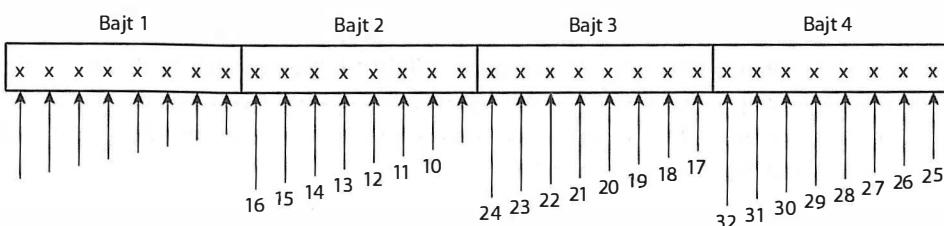
Przekształcenie danych z postaci równoległej (wykorzystywanej wewnętrznie) w szeregową wymaga zastosowania dodatkowych komponentów, których budowa może być prosta lub skomplikowana, zależnie od wykorzystywanego mechanizmu komunikacji szeregowej. W najprostszym przypadku zadanie konwersji realizuje pojedynczy układ scalony — uniwersalny odbiornik i nadajnik transmisji szeregowej (UART — ang. *Universal Asynchronous Receiver and Transmitter*). W sieciach synchronicznych tę samą funkcję pełni uniwersalny odbiornik i nadajnik transmisji synchronicznej i asynchronicznej (USART — ang. *Universal Synchronous-Asynchronous Receiver and Transmitter*).

## 9.5. Kolejność wysyłania bitów i bajtów

W przypadku transmisji szeregowej zawsze rodzi się interesujące pytanie. Który bit powinien być przesyłany jako pierwszy? Weźmy pod uwagę liczbę całkowitą. Czy jako pierwszy należy przesłać najbardziej znaczący bit (MSB — ang. *Most Significant Bit*), czy najmniej znaczący (LSB — ang. *Least Significant Bit*)?

Inżynierowie posługują się określeniem **little-endian**, odnosząc się do systemu, który jako pierwszy przesyła bit LSB, oraz pojęciem **big-endian** opisującym system, który w pierwszej kolejności nadaje bit MSB. Każde z rozwiązań jest dopuszczalne, ale wymaga jednokowego ustawienia nadajnika i odbiornika.

Ustalenie kolejności bitów nie rozwiązuje całego problemu zachowania właściwego porządku transmisji. Dane przetwarzane przez komputer są podzielone na bajty, a każdy bajt składa się z ośmiu bitów. Możliwe jest więc niezależne definiowanie kolejności bitowej i kolejności bajtowej. Na przykład w technologii Ethernet bajty są wysyłane w kolejności od najstarszego, a bity od najmłodszego. Na rysunku 9.4 przedstawiono kolejność bitową 32-bitowej porcji danych przesyłanej w sieci Ethernet.



Rysunek 9.4. Przykład transmisji, w której bajty są wysyłane w kolejności od najmniej znaczącego, a bity w kolejności od najbardziej znaczącego

## 9.6. Zależności czasowe w transmisji szeregowej

Mechanizmy transmisji szeregowej zalicza się do jednej z trzech kategorii, w zależności od sposobu doboru czasu transmisji:

- Transmisja **asynchroniczna** może zostać zainicjowana w dowolnym czasie bez względu na czas, który upłynął od poprzedniej operacji tego typu.
- Transmisja **synchroniczna** jest realizowana bezustannie — nie ma przerw między kolejnymi porcjami danych.
- Transmisja **izochroniczna** jest realizowana w regularnych interwałach — między kolejnymi porcjami danych są przerwy o określonej długości.

## 9.7. Transmisja asynchroniczna

System transmisji danych można uznać za **asynchroniczny**, jeśli pozwala na to, by medium transmisyjne pozostawało nieaktywne przez dowolny czas między dwiema kolejnymi transmisjami. Asynchroniczna wymiana danych doskonale sprawdza się więc w aplikacjach, które generują dane w sposób losowy (na przykład w wyniku naciskania klawiszy przez użytkownika lub klikania odsyłaczy na stronach internetowych).

Wadą operacji asynchronicznych jest brak koordynacji działań między nadajnikiem i odbiornikiem — odbiornik nie dysponuje informacjami o tym, przez jaki czas medium pozostanie nieaktywne. Z tego względu w rozwiązaniach asynchronicznych zazwyczaj nadajnik emituje kilka dodatkowych bitów przed przesaniem zasadniczych danych. W ten sposób informuje odbiornik o rozpoczęciu transmisji. Nadmiarowe bity umożliwiają synchronizowanie modułów odbiorczych z docierającym sygnałem. W niektórych specyfikacjach dodatkowe bity są nazywane **preambułą**, w innych **bitami startu**. Podsumowując:

*W systemach asynchronicznych medium transmisyjne może pozostawać wolne przez dowolny czas między kolejnymi transmisjami. Z tego powodu nadajnik poprzedza dane dodatkowymi bitami, które umożliwiają odbiornikowi zsynchronizowanie układów elektronicznych z rejestrowanym sygnałem.*

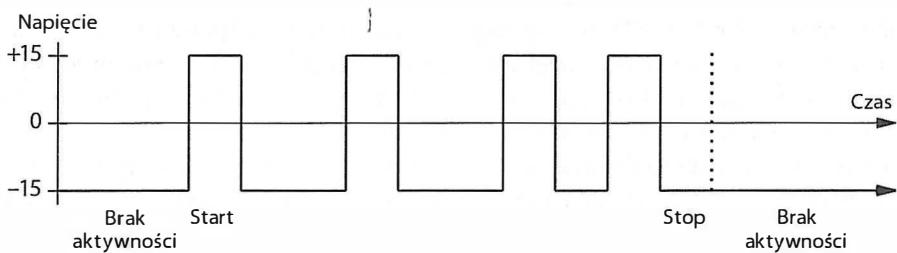
## 9.8. Asynchroniczna transmisja znaków — RS-232

Przykładem komunikacji asynchronicznej jest standard przesyłania znaków w przewodach miedzianych łączących komputer z urządzeniami takimi jak klawiatura. Technologia asynchronicznej transmisji danych, znormalizowana przez organizację Electronic Industries Alliance (EIA), stała się najpowszechniej stosowaną formą połączenia znakowego. W standardzie RS-232-C, którego nazwa często jest skracana do RS-232<sup>23</sup>, EIA zdefiniowała parametry fizyczne połączenia (na przykład maksymalną odległość między urządzeniami wynoszącą 15 m), jego właściwości elektryczne (na przykład zakres napięć od -15 V do +15 V) oraz opisała zasady kodowania liniowego (na przykład to, że jedynce logicznej odpowiada ujemna wartość napięcia, a zeru logicznemu wartość dodatnia).

Z uwagi na przeznaczenie standardu RS-232 do komunikacji z takimi urządzeniami jak klawiatura, specyfikacja przewiduje reprezentowanie każdego elementu danych za pomocą jednego znaku. Konfigurując urządzenie, użytkownik może określić szybkość transmisji danych w bitach na sekundę oraz ośmiobitowy lub siedmiobitowy sposób odwzorowania znaków. Odstęp czasowy między kolejnymi nadawanymi znakami nie jest w żaden sposób ograniczony. Niemniej po rozpoczęciu transmisji nadawca jest zobowiązany do przesłania wszystkich bitów danego znaku, bez jakichkolwiek przerw między nimi. Po zakończeniu przekazu nadajnik utrzymuje ujemne napięcie na linii danych (odpowiadające logicznej wartości 1) do czasu rozpoczęcia nowej transmisji.

<sup>23</sup> Choć późniejszy standard RS-449 jest nieco bardziej użyteczny, większość inżynierów nadal wykorzystuje pierwotne rozwiązanie.

Skąd odbiornik wie, że rozpoczyna się nadawanie nowego znaku? Standard RS-232 zakłada, że nadajnik emittuje dodatkowy bit 0 (nazywany bitem startu) przed rozpoczęciem transmisji bitów samego znaku. Ponadto zgodnie ze specyfikacją RS-232 nadawca musi utrzymać stan nieaktywności łączna między kolejnymi znakami przez czas odpowiadający przesłaniu jednego bitu. W terminologii RS-232 przerwa ta jest nazywana **bitem stopu**. Rozkład napięć w czasie transmisji bitu startu, ośmiu bitów danych i jednego bitu stopu pokazano na rysunku 9.5.



Rysunek 9.5. Zmiana napięcia w czasie transmisji 8-bitowego znaku w standardzie RS-232

Podsumowując:

*Standard RS-232, stosowany w szeregowej asynchronicznej wymianie danych na krótkich odległościach, wymusza poprzedzanie każdego znaku bitem startu, po którym następuje osiem bitów danych oraz okres nieaktywności o czasie odpowiadającym jednemu bitowi (bit stopu).*

## 9.9. Transmisja synchroniczna

Alternatywą dla transmisji asynchronicznej jest **transmisja synchroniczna**. Mechanizmy synchroniczne zapewniają ciągłą transmisję bitów danych, bez jakichkolwiek przerw między nimi. Oznacza to, że po zakończeniu nadawania jednego bajtu natychmiast rozpoczyna się emisja kolejnego.

Główną zaletą takiego rozwiązania jest stała synchronizacja nadajnika z odbiornikiem. To z kolei eliminuje narut związaną z koniecznością synchronizowania urządzeń. Aby uzmysłowić sobie skalę oszczędności czasu, wystarczy porównać mechanizm przesyłania 8-bitowych znaków w systemie asynchronicznym pokazanym na rysunku 9.5 z systemem transmisji synchronicznej zaprezentowanym na rysunku 9.6. Każdy znak przekazywany za pomocą połączenia RS-232 wymaga dodania bitu startu i bitu danych. W konsekwencji przesłanie 8-bitowego znaku zajmuje co najmniej czas 10 bitów, mimo że transmisja ma ciągły charakter. W połączeniach synchronicznych znaki są nadawane bez nadmiarowych bitów startu i stopu.



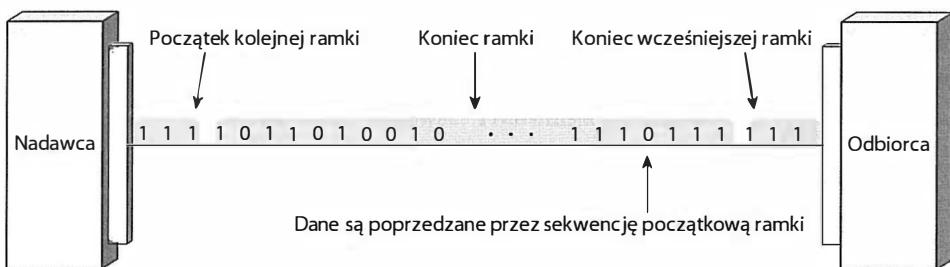
Rysunek 9.6. Transmisja synchroniczna, w której po ostatnich bitach jednego bajtu następują pierwsze bity kolejnego bajtu

Wynika z tego, że:

*Asynchroniczna transmisja w standardzie RS-232 jest o 25% bardziej czasochłonna niż transmisja synchroniczna.*

## 9.10. Bajty, bloki i ramki

Co się dzieje, gdy nadajnik pracujący w systemie synchronicznym nie dysponuje danymi, które mógłby bezustannie wysyłać? Odpowiedź jest zawarta w mechanizmie **ramkowania**. Moduły komunikacji synchronicznej są wyposażone w interfejsy, które odbierają **bloki bajtów** nazywane **ramkami**. Aby zagwarantować synchronizację między nadajnikiem i odbiornikiem, nadajnik rozpoczęta ramkę specjalną sekwencją bitową. Ponadto w większości rozwiązań synchronicznych uwzględnia się generowanie określonych **sekwencji nieaktywności** (lub **bajtów pracy jałowej**), które są transmitowane w chwilach, gdy nadajnik nie ma żadnych danych do przesłania. Koncepcja ta została zilustrowana na rysunku 9.7.



Rysunek 9.7. Ramkowanie w synchronicznym systemie transmisyjnym

Rezultaty ramkowania można podsumować w następujący sposób:

*Choć niskopoziomowe mechanizmy zapewniają ciągłe nadawanie bitów, wykorzystanie sekwencji nieaktywności i znaczników ramek umożliwia implementowanie za pomocą systemów synchronicznych interfejsów bajtowych i pozwala na wprowadzanie przerw w transmisji danych.*

## 9.11. Transmisja izochroniczna

Trzeci rodzaj transmisji szeregowej nie wprowadza żadnych nowych niskopoziomowych rozwiązań. Wręcz przeciwnie, można go rozpatrywać jako szczególną formę transmisji synchronicznej. **Transmisja izochroniczna** gwarantuje równomierny przepływ danych w aplikacjach multimedialnych, w których przekazywany jest głos i sekwencje wizyjne. Dostarczanie tego rodzaju danych z jednakową przepływnością bitową jest niezwykle istotne, ponieważ fluktuacje opóźnienia (określane jako **jitter**) mogą doprowadzić do problemów z odbiorem informacji (tj. mogą powodować generowanie trzasków w sygnale dźwiękowym lub wstrzymywanie ramek emitowanej sekwencji wideo).

Zamiast sterowania transmisją przez emitowanie danych w rozwiązańach izochronicznych implementuje się mechanizm odbierania i nadawania informacji o określonej przepływności ( $R$ ). W praktyce interfejs sieciowy jest konstruowany w taki sposób, aby **bezwarkownie** generował dane sieciowe z przepływnością  $R$  bitów na sekundę. Na przykład mechanizmy izochroniczne projektowane z przeznaczeniem do wykorzystania w transmisji głosu działają z przepływnością 64 000 bitów na sekundę. Nadajnik musi bezustannie generować cyfrowy sygnał dźwięku, a odbiornik musi odbierać i odtwarzać nadchodzące dane.

Niskopoziomowe mechanizmy sieciowe mogą stosować ramkowanie, a także przekazywać dodatkowe informacje wraz z danymi. Niemniej system izochroniczny musi być zdolny do przesyłania ciągłego strumienia danych między nadajnikiem i odbiornikiem, bez jakichkolwiek opóźnień związanych z początkiem ramki. Dlatego sieci izochroniczne gwarantujące przepustowość  $R$  bitów na sekundę zazwyczaj bazują na systemach transmisyjnych o nieznacznie większej przepustowości.

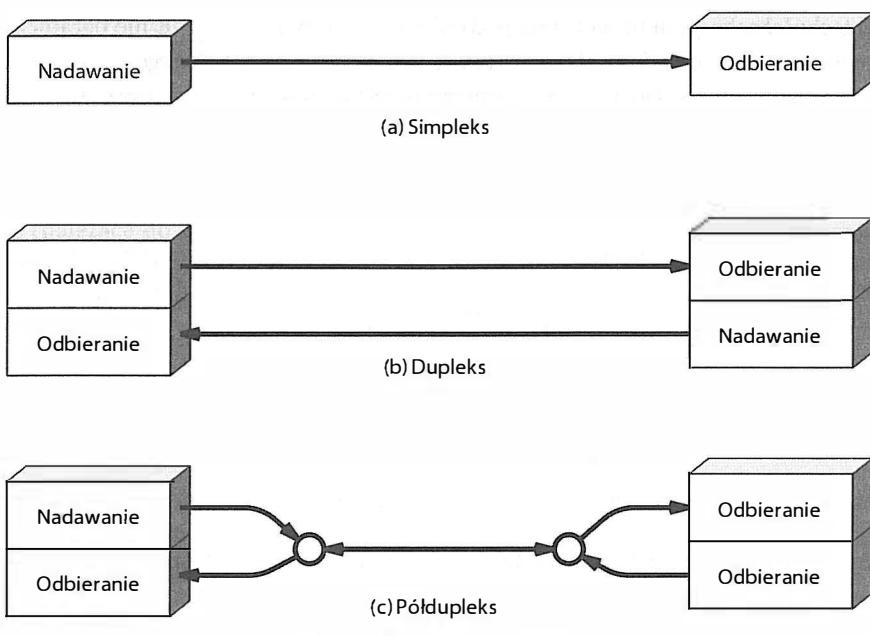
## 9.12. Simpleks, półduplek i dupleks

Kanały komunikacyjne są zaliczane do jednej z trzech kategorii zależnie od kierunku transferu danych. Oto rodzaje transmisji:

- simpleks,
- półduplek,
- dupleks.

**Simplex.** Działanie mechanizmu simpleksowego jest najłatwiejsze do zrozumienia. Dane są bowiem przekazywane tylko w jednym kierunku. Na przykład pojedyncze włókno światłowodowe działa w trybie simpleksowym, ponieważ na jednym jego końcu jest zainstalowane urządzenie nadawcze (dioda LED lub laser), a na drugim końcu działa odbiornik (element światłoczuły). Nadajniki radiowe i telewizyjne również pracują w trybie simpleksu. Działanie mechanizmów simpleksowych ilustruje rysunek 9.8a.

**Dupleks.** Działanie mechanizmów dupleksowych również nie jest szczególnie trudne do zrozumienia — system transmisyjny umożliwia przekazywanie danych w dwóch kierunkach jednocześnie. Zazwyczaj rozwiązania dupleksowe składają się z dwóch syste-



Rysunek 9.8. Trzy tryby pracy

mów simpleksowych, z których każdy odpowiada za przesyłanie informacji w jednym kierunku (zgodnie z rysunkiem 9.8b). Parę włókien optycznych można więc uznać za system komunikacji dupleksowej, jeśli będą one przekazywały dane jednocześnie w obydwu kierunkach. Komunikacja w trybie dupleksu jest analogiczna do prowadzenia rozmowy telefonicznej, której uczestnicy słyszą muzykę lub dźwięki tła nawet wówczas, gdy sami mówią.

**Półdupleks.** Mechanizmy półdupleksowe są implementowane w przypadkach współdzielenia medium transmisyjnego. Wspólne medium może być wykorzystywane do komunikacji dwukierunkowej, ale niemożliwe jest jednoczesne transmitowanie danych w dwóch kierunkach. Systemy tego typu działają podobnie do krótkofałówek — w określonym czasie tylko jeden z rozmówców może nadawać. Transmisja półdupleksowa została przedstawiona na rysunku 9.8c.

## 9.13. Urządzenia DCE i DTE

Skróty DCE (urządzenie transmisji danych) i DTE (urządzenie terminalowe) zostały wprowadzone przez firmę AT&T, która chciała odróżnić urządzenia należące do operatora telekomunikacyjnego od wyposażenia abonenckiego.

Mimo upływu czasu obydwa skróty nadal są stosowane. Gdy przedsiębiorstwo zdecyduje się na wydzierżawienie łącza od operatora telekomunikacyjnego, firma telekomunikacyjna instaluje urządzenia DCE w siedzibie klienta, a odbiorca łącza kupuje urządzenie DCE, które przyłącza do komponentów zainstalowanych przez operatora.

Z technicznego punktu widzenia podział na elementy DCE i DTE nie ogranicza się tylko do praw własności. Umożliwia natomiast wprowadzenie określonego interfejsu przyłączeniowego. Na przykład jeśli w sieci operatora telekomunikacyjnego stosowane są tylko mechanizmy transmisji synchronicznej, urządzenie DCE może zapewnić obsługę komunikacji synchronicznej lub izochronicznej obsługiwanej przez urządzenia odbiorcy. Zasada podziału urządzeń została przedstawiona na rysunku 9.9<sup>24</sup>.



Rysunek 9.9. Urządzenia DCE i DTE zapewniające połączenie między dwoma lokalizacjami

Interfejsy połączeń DCE-DTE zostały zdefiniowane w wielu standardach. Można w tym charakterze zastosować połączenia RS-232 (opisane we wcześniejszej części rozdziału) lub RS-449 (standard proponowany jako następca RS-232). Często wykorzystywany jest również standard X.21.

## 9.14. Podsumowanie

Systemy komunikacyjne wykorzystują transmisję równoległą lub szeregową. Mechanizmy równoległego przesyłania danych wymagają użycia wielu przewodów, z których każdy odpowiada za przekazanie jednego bitu w danym czasie. Systemy równoległe o  $K$  przewodach mogą więc dostarczać jednocześnie  $K$  bitów danych. Choć rozwiązania tego typu zapewniają szybszą transmisję danych, większość systemów komunikacyjnych bazuje na tańszych mechanizmach szeregowych, które umożliwiają przesyłanie pojedynczych bitów.

Komunikacja szeregowa wymaga od nadawcy i odbiorcy uzgodnienia parametrów czasowych połączenia oraz kolejności generowania bitów danych. Kolejność nadawania dotyczy tego, czy jako pierwsze są generowane najbardziej znaczące, czy najmniej znaczące bity oraz czy w pierwszej kolejności są przesyłane bajty starsze, czy młodsze.

Wymianę danych charakteryzuje również trzy rodzaje zależności czasowych. W transmisji asynchronicznej nadawanie może się rozpocząć w dowolnej chwili, a łącze może pozostawać nieaktywne między kolejnymi emisjami. Działanie synchroniczne zakłada ciągłą transmisję bitów danych, które są grupowane w ramki. Komunikacja izochroniczna zapewnia przekazywanie informacji w określonych odstępach czasowych, bez dodatkowych opóźnień związanych z ramkowaniem.

<sup>24</sup> Skróty DCE i DTE są również stosowane do rozróżnienia dwóch rodzajów złączy niezależnie od ich producenta (na przykład złącza zainstalowanego w komputerze PC i dostępnego w modemie zewnętrznym).

Systemy transmisyjne dzieli się również na systemy simpleksowe, dupleksowe i pół-dupleksowe. Mechanizm simpleksowy odpowiada za przesyłanie danych tylko w jednym kierunku. Rozwiązywanie dupleksowe umożliwia transfer informacji w dwóch kierunkach jednocześnie. Natomiast system pół-dupleksowy pozwala na dwukierunkową wymianę danych, ale z ograniczeniem do transmisji w jednym kierunku w danym czasie.

W celu wyraźnego określenia własności urządzeń wprowadzono podział na jednostki DCE (należące do operatora telekomunikacyjnego) i DTE (należące do abonentów). Jednak najważniejszą jego zaletą jest możliwość zdefiniowania interfejsu, dzięki któremu użytkownik będzie korzystał z innych usług niż oferowane bezpośrednio przez system transmisyjny.

## ZADANIA

- 9.1. Opisz różnice między transmisją równoległą i szeregową.
- 9.2. Wymień zalety transmisji równoległej. Jaka jest główna wada tego mechanizmu?
- 9.3. Kiedy nadawany jest bit znaku w transmisji 32-bitowej liczby całkowitej, zapisanej w formacie uzupełniania do dwóch, emitowanej zgodnie z kolejnością big-endian?
- 9.4. Wymień najważniejsze cechy transmisji asynchronicznej.
- 9.5. Który rodzaj (które rodzaje) transmisji szeregowej jest odpowiedni (są odpowiednie) do przekazywania strumieni wideo? A które są najlepsze do połączenia klawiatury z komputerem?
- 9.6. Co to jest bit startu? W jakich rodzinach transmisji szeregowej stosuje się bit startu?
- 9.7. Co się dzieje, gdy nadawca nie ma danych do przesłania, a wykorzystuje mechanizm transmisji synchronicznej?
- 9.8. Czy rozmowa między dwiema osobami jest realizowana w trybie simpleks, dupleks, czy pół-dupleks?
- 9.9. Czy modem jest urządzeniem DTE, czy DCE?
- 9.10. Znajdź w internecie informacje o rozmieszczeniu wyprowadzeń DCE i DTE w złączu DB-25. Podpowiedź: wyprowadzenia o numerach 2 i 3 służą do nadawania i odbierania danych. Czy w złączu DCE wyprowadzenie 2 jest przeznaczone do nadawania, czy do odbioru?

# Zawartość rozdziału

- 10.1. Wprowadzenie 191
- 10.2. Częstotliwość, fala nośna i propagacja 191
- 10.3. Modulacja analogowa 192
- 10.4. Modulacja amplitudy 192
- 10.5. Modulacja częstotliwości 193
- 10.6. Modulacja fazy 194
- 10.7. Modulacja amplitudy i twierdzenie Shannona 194
- 10.8. Modulacja, sygnał cyfrowy i kluczowanie 194
- 10.9. Kluczowanie fazy 195
- 10.10. Przesunięcie fazowe i diagram konstelacji 195
- 10.11. Kwadraturowa modulacja amplitudy 198
- 10.12. Modem — urządzenie do modulacji i demodulacji 198
- 10.13. Modemy optyczne i radiowe 200
- 10.14. Modemy telefoniczne 200
- 10.15. Modulacja QAM w telefonii 201
- 10.16. Modemy V.32 i V.32bis 201
- 10.17. Podsumowanie 202

# 10

## *Modulacja i modemy*

### 10.1. Wprowadzenie

Każdy z rozdziałów tej części książki dotyczy innego aspektu wymiany danych. W poprzednich rozdziałach opisane zostały źródła danych, sposoby reprezentowania informacji za pomocą transmitowanego sygnału oraz różne formy energii przekazywanej w mediach transmisyjnych.

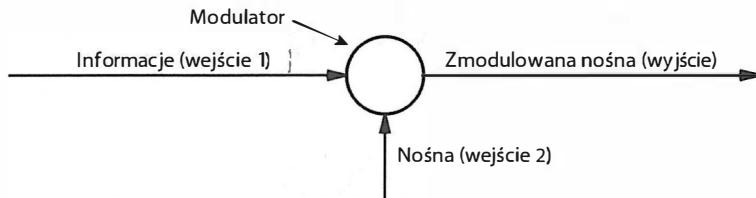
W tym rozdziale kontynuowany jest przegląd rozwiązań związanych z transmisją danych ze szczególnym uwzględnieniem technik przenoszenia informacji za pomocą sygnałów o wysokich częstotliwościach. Omówione zostały tutaj zagadnienia zmiany charakterystyki fal elektromagnetycznych w zależności od przesyłanych informacji, a także zasady wykorzystywania do tego celu analogowych i cyfrowych sygnałów informacyjnych. W kolejnych rozdziałach tematyka ta została rozszerzona o techniki komunikacyjne umożliwiające przekazywanie wielu niezależnych strumieni danych w ramach współdzielonego medium transmisyjnego.

### 10.2. Częstotliwość, fala nośna i propagacja

Wiele systemów komunikacyjnych przekazujących dane na dużych odległościach wykorzystuje nieustannie oscylującą falę elektromagnetyczną, zwaną **falą nośną** (lub po prostu **nośną**). Zadaniem nadajnika jest wprowadzanie nieznacznych zmian w kształcie fali tak, aby reprezentowały one przesyłane informacje. Aby zrozumieć, dlaczego stosowanie specjalnych fal nośnych ma tak istotne znaczenie w transmisji danych, wystarczy przypomnieć sobie wpływ częstotliwości fal elektromagnetycznych na propagację sygnału, opisany w rozdziale 7. Jedną z przyczyn generowania sygnałów nośnych jest właśnie chęć wybrania częstotliwości, która zapewni odpowiednie rozchodzenie się fal niezależnie od przepływności generowanego strumienia danych.

### 10.3. Modulacja analogowa

Modyfikowanie kształtu fali nośnej zależnie od przekazywanych informacji nazywa się **modulacją**. Teoretycznie modulacja polega na przetwarzaniu dwóch sygnałów — nośnej i sygnału danych. Celem operacji jest wygenerowanie zmodulowanego sygnału nośnego. Realizacja zadania została zilustrowana na rysunku 10.1.



Rysunek 10.1. Modulacja z wykorzystaniem dwóch sygnałów wejściowych

Zadanie nadajnika polega na zmianie jednego z podstawowych parametrów fali nośnej. Istnieją trzy ogólne techniki modulowania fali nośnej, a każda z nich bazuje na innym sposobie odwzorowywania sygnału informacyjnego. Są to:

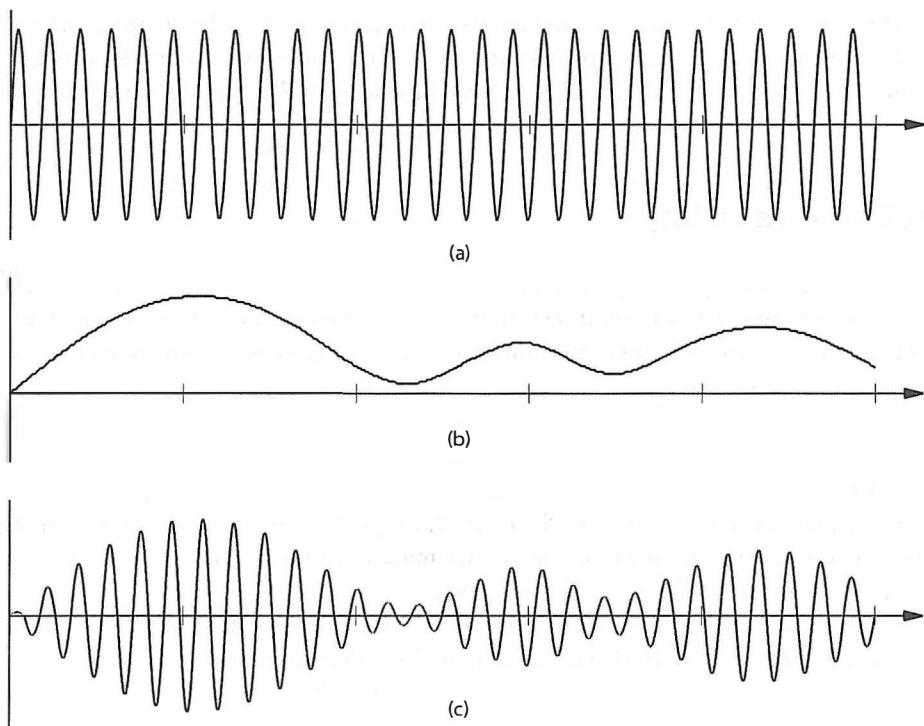
- modulacja amplitudy,
- modulacja częstotliwości,
- modulacja fazy.

Dwie pierwsze techniki modulacji są bardzo popularne i niezwykle często wykorzystywane. Nie wywodzą się jednak z sieci komputerowych. Zostały opracowane z myślą o transmitowaniu sygnałów rozgłośni radiowych i telewizyjnych.

### 10.4. Modulacja amplitudy

Technika nazywana **modulacją amplitudy** ma na celu modyfikowanie amplitudy fali nośnej zależnie od przesyłanych w danej chwili informacji (tj. w zależności od sygnału wejściowego). Częstotliwość oscylacji fali nośnej pozostaje bez zmian, ale amplituda podlega ciągłym zmianom. Na rysunku 10.2 przedstawiono niezmodulowaną falę nośną, analogowy sygnał informacyjny oraz wynik amplitudowej modulacji fali nośnej.

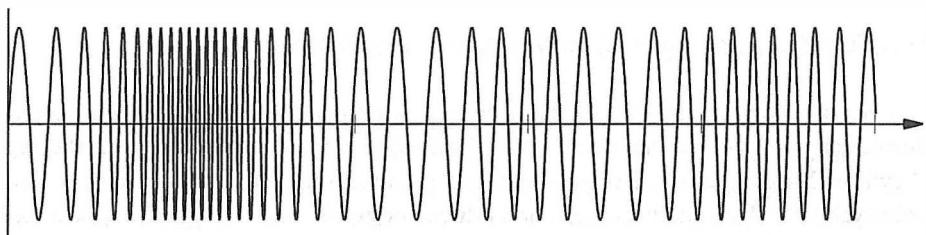
Modulacja amplitudy jest najłatwiejsza do wyjaśnienia, ponieważ zmieniana jest jedynie amplituda przebiegu sinusoidalnego. Wykres zmodulowanego sygnału nośnego przypomina swoim kształtem sygnał wykorzystany do modulacji. Wystarczy sobie wyobrazić **obwiednię** łączącą szczyty przebiegu sinusoidalnego z rysunku 10.2c, aby dostrzec w nim kształt sygnału informacyjnego, widocznego na rysunku 10.2b.



**Rysunek 10.2.** Niemodulowana fala nośna (a), analogowy sygnał informacyjny (b), wynik modulacji amplitudy (c)

## 10.5. Modulacja częstotliwości

Rozwiązańiem alternatywnym w stosunku do modulacji amplitudy jest **modulacja częstotliwości**. Zastosowanie tej techniki powoduje, że amplituda sygnału nośnego pozostaje niezmienna. Natomiast w takt sygnału danych zmienia się częstotliwość fali nośnej — gdy sygnał ma wyższy poziom, częstotliwość nośnej wzrasta, a osłabianiu się sygnału wejściowego towarzyszy zmniejszanie częstotliwości nośnej. Na rysunku 10.3 pokazano przebieg sygnału nośnego, który został zmodulowany przez sygnał informacyjny widoczny na rysunku 10.2b.



**Rysunek 10.3.** Fala nośna zmodulowana częstotliwościowo sygnałem z rysunku 10.2b

Modulacja częstotliwości jest znacznie trudniejsza do przedstawienia, ponieważ niewielkie zmiany w częstotliwości przebiegu są trudne do zauważenia. Niemniej wnikiwa analiza rysunku prowadzi do wniosku, że częstotliwość fali nośnej jest wyższa w tej części wykresu, w której sygnał modulujący ma wyższy poziom.

## 10.6. Modulacja fazy

Trzecią właściwością przebiegu sinusoidalnego jest jego **faza**, czyli przesunięcie względem czasu odniesienia, w którym fala sinusoidalna rozpoczyna oscylacje. Zmiany fazy również mogą służyć do reprezentowania sygnału informacyjnego. Zmiany takie nazywane są **modulacją fazy**.

W przypadku sygnału analogowego modulacja fazy jest stosowana bardzo rzadko, choć teoretycznie nic nie stoi na przeszkodzie, aby z niej korzystać. Chcąc zrozumieć zasadę jej działania, należałoby wyobrazić sobie sytuację, w której faza zmienia się po okresie  $k$ . Następny przebieg sinusoidalny mógłby rozpocząć się nieco później, niż następuje zakończenie okresu  $k$ . Nieznaczne opóźnienie przypomina w pewnym sensie zmianę częstotliwości. Dlatego w odniesieniu do sygnału analogowego modulację fazy można traktować jako szczególną formę modulacji częstotliwości. Zmiany fazy są jednak bardzo użyteczne w rozwiązaniach, w których fala nośna jest modulowana przez sygnał cyfrowy.

## 10.7. Modulacja amplitudy i twierdzenie Shannona

Z rysunku 10.2c wynika, że amplituda przebiegu wynikowego zmienia się od wartości maksymalnej do wartości bliskiej零. Choć taki sposób prezentacji zagadnienia ułatwia jego zrozumienie, przedstawiony rysunek jest nieco mylący. W rzeczywistości modulacja powoduje jedynie nieznaczne zmiany w amplitudzie sygnału nośnego, których zakres jest uzależniony od stałej nazywanej **indeksem modulacji**.

Powód, dla którego obwiednia sygnału zmodulowanego nie zbliża się do zero, wynika z twierdzenia Shannona. Jeśli założymy, że poziom szumu jest stały, to obniżenie poziomu sygnału do wartości bliskiej零 powoduje jednoczesne zmniejszenie stosunku sygnału do szumu. Dlatego utrzymanie poziomu nośnej w pobliżu wartości maksymalnej gwarantuje utrzymanie maksymalnego stosunku sygnału do szumu, który z kolei umożliwia przesyłanie większej liczby bitów w sekundzie.

## 10.8. Modulacja, sygnał cyfrowy i kluczowanie

Dotychczasowy opis modulacji odnosił się do modulowania nośnej przez analogowy sygnał informacyjny. Nasuwa się więc pytanie, w jaki sposób można do tego celu wykorzystać sygnał cyfrowy? Rozwiązaniem jest wprowadzenie pewnych niewielkich modyfikacji w standardowych technikach modulacji. Zamiast odwzorowywania ciągłego sygnału wejściowego mechanizmy cyfrowe operują nieciągłymi wartościami sygnału informacyjnego. Zmiana

jest zauważalna również w terminologii. W celu odróżnienia modulacji analogowej od cyfrowej stosuje się określenie **kluczowanie** zamiast modulacja.

Kluczowanie jest realizowane na podobnej zasadzie co modulacja. Jednak zamiast ciągłego przedziału wartości wykorzystuje się w nim zbiór ustalonych wartości. Na przykład modulacja amplitudy umożliwia zmianę poziomu nośnej o dowolnie małą wartość, proporcjonalną do zmian sygnału informacyjnego. Z kolei kluczowanie amplitudy bazuje na wstępnie wyznaczonym zbiorze dozwolonych poziomów amplitudy. W najprostszym przypadku maksymalna wartość amplitudy odpowiada logicznej jedynce, a nieznacznie mniejszy poziom reprezentuje logiczne zero. Kluczowanie częstotliwości działa podobnie, ale wykorzystuje dwie częstotliwości sygnału nośnego. Na rysunku 10.4 przedstawiono przykładowy przebiegi fali nośnej, cyfrowego sygnału informacyjnego oraz fali nośnej zmodulowanej amplitudowo (ASK — ang. *Amplitude Shift Keying*) i częstotliwościowo (FSK — ang. *Frequency Shift Keying*).

## 10.9. Kluczowanie fazy

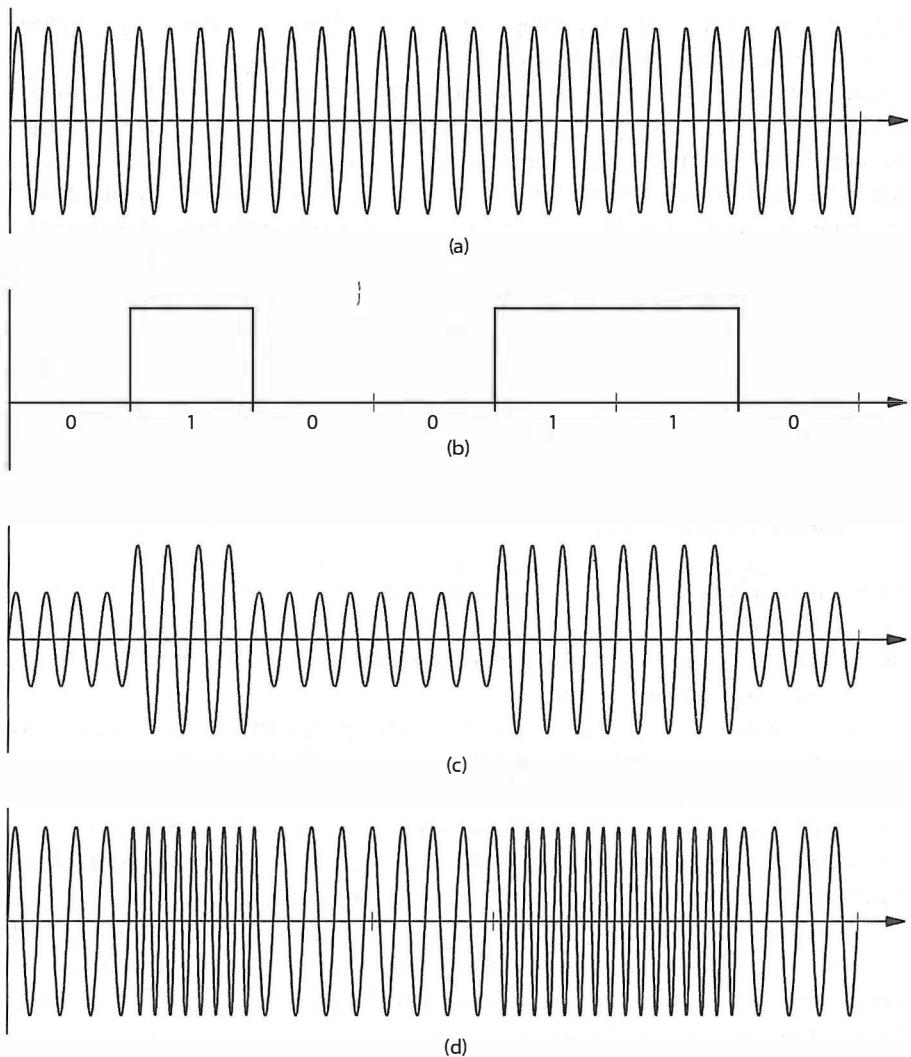
Zmiany amplitudy i częstotliwości doskonale sprawdzają się w systemach audio. Wymagają jednak co najmniej jednego pełnego okresu nośnej do przesłania pojedynczego bitu (o ile nie zastosuje się specjalnego schematu kodowania, w którym dodatnie i ujemne części przebiegu są zmieniane niezależnie).

Z zaprezentowanego w rozdziale 6. twierdzenia Nyquista wynika, że możliwe jest zwiększenie liczby bitów transmitowanych w jednostce czasu, jeśli schemat kodowania pozwala na zakodowanie większej liczby bitów w pojedynczym okresie fali nośnej. Z tego powodu w systemach transmisji danych często stosowane są rozwiązania umożliwiające przesyłanie większej liczby bitów. Szczególnie użyteczne okazuje się w tym przypadku **kluczowanie fazy**, w którym kodowanie danych polega na szybkiej zmianie fazy fali nośnej. Każda zmiana jest nazywana **przesunięciem fazy**. Po przesunięciu fazy nośna oscyluje w standardowy sposób, ale może w każdej chwili zmienić wartość na odpowiadającą dowolnemu nowemu punktowi przebiegu sinusoidalnego. Wpływ zmian fazy na falę sinusoidalną przedstawiono na rysunku 10.5.

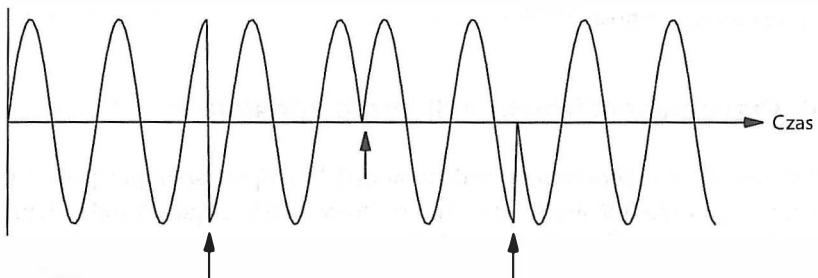
Przesunięcie fazy jest wyrażane za pomocą wartości kąta odpowiadającego zmianie. Na przykład pierwsza ze zmian przedstawionych na rysunku 10.5 ma wartość kątową  $\pi/2$  radianów, czyli  $180^\circ$ . Druga zmiana również odpowiada kątowi  $180^\circ$ , ale w trzecim przypadku przesunięcie wynosi  $-90^\circ$  (czyli  $270^\circ$ ).

## 10.10. Przesunięcie fazowe i diagram konstelacji

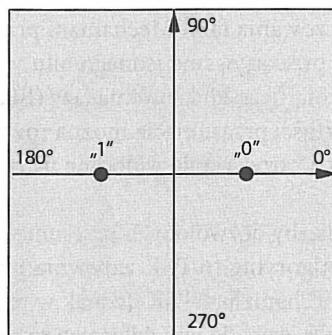
W jaki sposób dane są kodowane w zmianach fazy? W najprostszym przypadku nadawca i odbiorca mogą uzgodnić liczbę bitów transmitowanych w ciągu sekundy i reprezentować zero logiczne brakiem zmiany, a jedynkę logiczną przez zmianę fazy. W takim rozwiązaniu zmiana mogłaby wynosić  $180^\circ$ . Do przedstawienia powiązań między bitami danych a zmianami fazy wykorzystuje się **diagram konstelacji**, którego przykład jest widoczny na rysunku 10.6.



**Rysunek 10.4.** Fala nośna (a), cyfrowy sygnał informacyjny (b), kluczowanie amplitudy (c), kluczowanie częstotliwości (d)



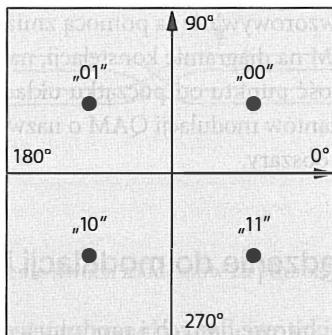
**Rysunek 10.5.** Przykład kluczowania fazy ze strzałkami wyznaczającymi chwile, w których nośna gwałtownie przeskakuje do nowego punktu przebiegu sinusoidalnego



**Rysunek 10.6.** Diagram konstelacji, który reprezentuje logiczne zero jako przesunięcie fazy o  $0^\circ$  i logiczną jedynkę jako przesunięcie o wartośći  $180^\circ$

Wyspecjalizowane moduły sprzętowe mogą wykonywać znacznie więcej zadań niż tylko wykrywanie przesunięć fazowych. Odbiornik może mierzyć wielkość zmiany. Nic nie stoi więc na przeszkodzie, aby utworzyć system komunikacyjny, który będzie rozpoznawał wiele różnych przesunięć fazowych i na ich podstawie wnioskował o wartości strumienia danych. Zazwyczaj tego typu rozwiązania bazują na przesunięciach, których liczba odpowiada potędze dwójki. Dzięki temu nadawca może wprost odwzorować bity danych w zmianach fazy.

Na rysunku 10.7 pokazano diagram konstelacji odpowiadający systemowi, w którym stosowane są cztery przesunięcia fazy (czyli  $2^2$ ). Na każdym etapie transmisji nadajnik odwzorowuje dwa bity danych na jedną spośród czterech dozwolonych wartości przesunięcia.



**Rysunek 10.7.** Diagram konstelacji odpowiadający systemowi o czterech wartościach przesunięcia fazy, z których każde reprezentuje dwa bity danych

Podsumowując:

Największą zaletą takich mechanizmów jak kluczowanie fazy jest możliwość reprezentowania większej liczby bitów za pomocą jednej zmiany fali nośnej. Powiązanie między bitami danych a zmianami fazy przedstawia diagram konstelacji.

Istnieje wiele odmian kluczowania fazy. Mechanizm przedstawiony na rysunku 10.6, który umożliwia nadajnikowi przekazywanie jednego bitu w danym czasie, zostałby sklasyfikowany jako mechanizm binarnego kluczowania fazy (BPSK — ang. *Binary Phase Shift Keying*). Dwie dozwolone wartości przesunięcia można również zapisać jako 2-PSK. Stosując tę samą zasadę klasyfikacji, rozwiązanie widoczne na rysunku 10.7 należałoby opisać jako 4-PSK.

Teoretycznie zwiększanie liczby dozwolonych przesunięć fazowych pozwala na zwiększenie szybkości transmisji. Algorytm 16-PSK zapewnia przesyłanie dwa razy większej liczby bitów na sekundę niż mechanizm 4-PSK. Jednak w praktyce szum i zniekształcenia sygnału ograniczają zdolność urządzeń do prawidłowego rozpoznawania niewielkich różnic w fazie.

*Choć istnieje wiele odmian kluczowania fazy, szумy i zniekształcenia sygnału ograniczają zdolność praktycznych systemów do rozróżniania małych zmian wartości fazy.*

## 10.11. Kwadraturowa modulacja amplitudy

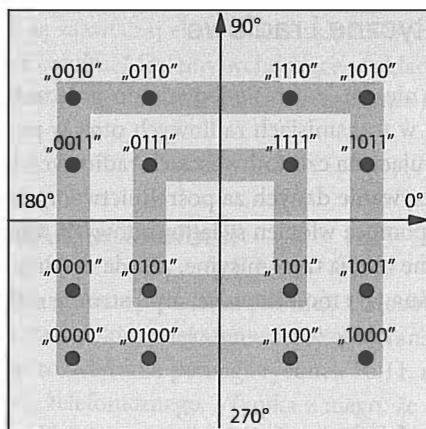
Skoro urządzenia nie są w stanie wykrywać dowolnie małych zmian fazy, to czy można jeszcze w jakikolwiek sposób zwiększyć przepływność bitową? Rozwiązaniem jest połączenie różnych technik modulacji i modyfikowanie kilku cech nośnej w tym samym czasie. Najbardziej wyrafinowanym w tym względzie mechanizmem jest kwadraturowa modulacja amplitudy (QAM — ang. *Quadrature Amplitude Modulation*)<sup>25</sup>. Wartości sygnału informacyjnego są w nim odwzorowywane za pomocą zmian fazy i amplitudy.

Analizując modulację QAM na diagramie konstelacji, należy przyjąć, że amplituda jest reprezentowana przez odległość punktu od początku układu współrzędnych. Na rysunku 10.8 pokazano jeden z wariantów modulacji QAM o nazwie 16QAM. Amplitudy zostały na nim oznaczone jako szare obszary.

## 10.12. Modem — urządzenie do modulacji i demodulacji

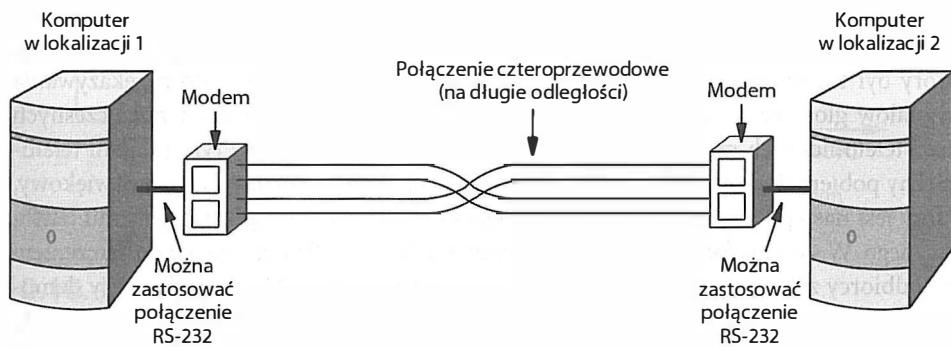
Urządzenia odbierające ciągi bitowe danych i modulujące fale nośne odpowiednio do przebiegu sygnału informacyjnego są nazywane **modulatorami**. Z kolei urządzenia odbierające zmodulowany sygnał nośny i odtwarzające pierwotną sekwencję bitową są nazywane **demodulatorami**. Transmisja danych wymaga więc zastosowania modulatora po stronie nadawczej i demodulatora po stronie odbiorczej. W praktyce większość systemów komunikacyjnych działa w trybie dupleksowym, co oznacza, że po obu stronach medium transmisyjnego muszą być zainstalowane zarówno modulatory (do wysyłania danych),

<sup>25</sup> Nazwa **kwadraturowa modulacja amplitudy** jest częściej stosowana w literaturze niż **kwadraturowe kluczowanie amplitudy**.



Rysunek 10.8. Diagram konstelacji dla modulacji 16QAM, na którym amplituda została oznaczona szarym kolorem

jak i demodulatory (do odbierania danych). W celu obniżenia kosztów, a także uproszczenia instalacji i utrzymania systemu producenci urządzeń implementują modulatory i demodulatory w pojedynczych modułach nazywanych **modemami** (od słów **modulator** i **demodulator**). Na rysunku 10.9 przedstawiono przykład połączenia dwóch modemów za pomocą czterech przewodów.



Rysunek 10.9. Połączenie dwóch modemów za pomocą czterech przewodów

Zgodnie z informacją zawartą na rysunku modem są projektowane do obsługi połączeń na dużych odległościach. Czteroprzewodowe połączenie dwóch modemów może być stosowane wewnętrz budynku, pomiędzy budynkami firmowymi lub pomiędzy miastami<sup>26</sup>.

<sup>26</sup> Połączenia przechodzące przez obszary publiczne muszą być dzierżawione od dostawców usług (zazwyczaj są to firmy telekomunikacyjne).

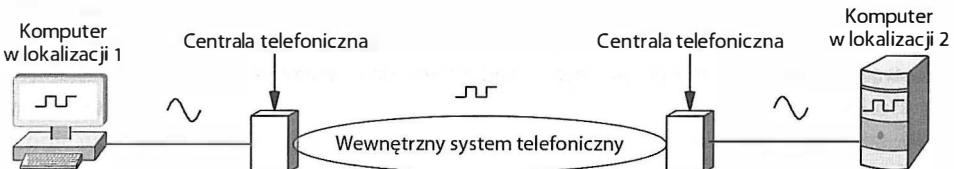
## 10.13. Modemy optyczne i radioowe

Zastosowanie modemów nie ogranicza się jedynie do połączeń przewodowych. Są one wykorzystywane również w transmisjach radiowych oraz w połączeniach światłowodowych. Dwa modemy pracujące na częstotliwościach radiowych (RF — ang. *Radio Frequency*) umożliwiają przekazywanie danych za pośrednictwem radia, a **modemy optyczne** przekazują informacje za pomocą włókien światłowodowych. Choć w ich działaniu wykorzystywane są zupełnie inne media transmisyjne, zasada działania pozostaje taka sama — po stronie nadawczej nośna jest modyfikowana, a po stronie odbiorczej są z niej odtwarzane dane.

## 10.14. Modemy telefoniczne

Kolejną interesującą aplikacją modemów jest wykorzystanie ich w systemach telefonii. Nośną nie jest wówczas sygnał elektryczny, lecz dźwięk. Podobnie jak w przypadku klasycznych modemów **modemy telefoniczne** odpowiadają za modulację nośnej po stronie nadawczej i demodulowanie jej po stronie odbiorczej. Zatem oprócz możliwości ustalania i odbierania połączeń różnica między modemami telefonicznymi i konwencjonalnymi wynika ze stosowania pasma o mniejszej szerokości, odpowiadającej częstotliwościom głosu.

W początkowym okresie rozwoju modemów telefonicznych opisane rozwiązanie wydawało się bardzo obiecujące. Modem przekształcał dane w zmodulowany sygnał analogowy, który był transportowany przez system telefoniczny przystosowany do przekazywania sygnałów głosowych. Jest trochę ironii w tym, że wewnętrzna budowa nowoczesnych sieci telefonicznych ma całkowicie cyfrowy charakter. W związku z tym modem telefoniczny pobiera dane cyfrowe i moduluje na ich podstawie analogowy sygnał dźwiękowy, który jest następnie przekształcany do postaci cyfrowej przez urządzenia systemu telefonicznego. W sieci telefonicznej jest przekazywany w formie cyfrowej, a przed dostarczeniem do odbiorcy znów podlega konwersji do postaci analogowej. Modem odbiorczy demoduluje analogową nośną i odtwarza na jej podstawie pierwotny cyfrowy ciąg danych. Przetwarzanie sygnału analogowego i cyfrowego w modemach telefonicznych zostało zilustrowane na rysunku 10.10.

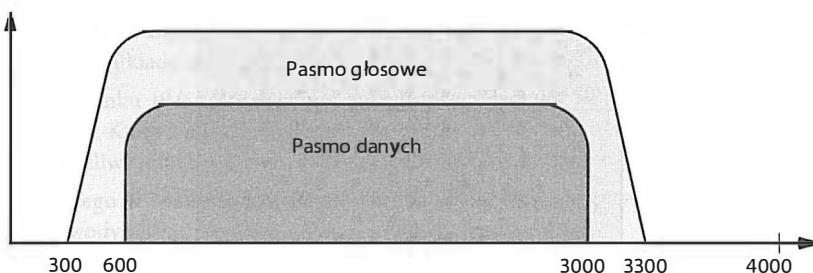


**Rysunek 10.10.** Przekazywanie sygnałów analogowych i cyfrowych (oznaczonych odpowiednio przebiegiem sinusoidalnymi i prostokątnymi) przez modemy odpowiedzialne za połączenie jednego komputera z drugim

Modemy telefoniczne są zazwyczaj elementami składowymi komputerów, co zostało pokazane na powyższym rysunku. Modemy wchodzące w skład komputerów nazywa się **modemami wewnętrznymi**, natomiast niezależne urządzenia tego typu określa się jako **modemy zewnętrzne**.

## 10.15. Modulacja QAM w telefonii

Kwadraturowa modulacja amplitudy znalazła również zastosowanie w modemach telefonicznych, gdyż pozwala na szybkie zwiększenie szybkości transmisyjnych. W zrozumieniu zasad działania takiego rozwiązania pomaga rysunek 10.11, na którym przedstawiono szerokość pasma połączenia telefonicznego. Wynika z niego, że większość systemów telefonicznych przenosi częstotliwości z przedziału od 300 Hz do 3300 Hz, choć niekiedy na granicach zakresu parametry łączą się nieco gorsze. Dlatego w celu zapewnienia wiernego odtworzenia danych i obniżenia poziomu szumu modemy telefoniczne korzystają z częstotliwości od 600 Hz do 3000 Hz. Szerokość pasma wynosi więc 2400 Hz. Zastosowanie modulacji QAM pozwala więc na istotne zwiększenie przepustowości.

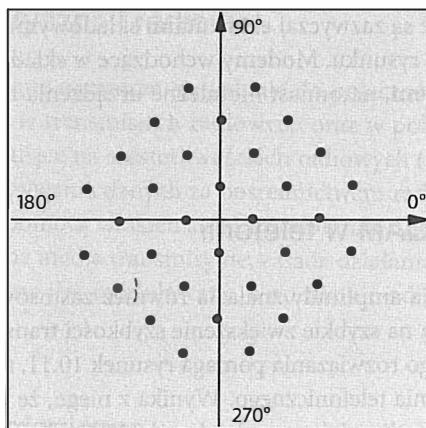


Rysunek 10.11. Pasmo głosowe i pasmo danych w połączeniu telefonicznym

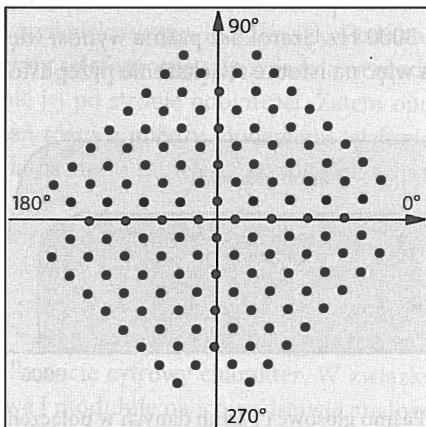
## 10.16. Modemy V.32 i V.32bis

Przykładami modemów telefonicznych wykorzystujących modulację QAM są urządzenia pracujące zgodnie ze standardem V.32 i V.32bis. Konstelacja QAM właściwa dla modemów V.32 została pokazana na rysunku 10.12. Zastosowanie 32 kombinacji wartości amplitudy i przesunięcia fazowego umożliwia przesyłanie danych z przepływnością 9600 b/s w każdym kierunku.

Modemy V.32bis wykorzystują 128 kombinacji wartości amplitudy i przesunięcia fazowego do przekazywania informacji z szybkością 14 400 b/s. Konstelacja odpowiadająca standardowi V.32bis została przedstawiona na rysunku 10.13. Wykrywanie tak niewielkich zmian między punktami konstelacji wymaga bardzo wyrafinowanych technik analizy sygnału.



Rysunek 10.12. Konstelacja modulacji QAM w modemie telefonicznym V.32



Rysunek 10.13. Konstelacja modulacji QAM w modemie telefonicznym V.32bis

## 10.17. Podsumowanie

Systemy przesyłania danych na dużych odległościach wykorzystują techniki modulowania fal nośnych do przenoszenia informacji. Modulacja nośnej polega na zmianie amplitudy, częstotliwości lub fazy przebiegu. W przypadku analogowych sygnałów informacyjnych najczęściej stosowane są modulacje amplitudy i częstotliwości.

Gdy sygnałem informacyjnym jest sygnał cyfrowy, proces modulacji nazywa się kluczowaniem. Podobnie jak w przypadku modulacji analogowej kluczowanie powoduje zmiany w przebiegu fali nośnej. Dozwolone są jednak tylko wybrane wartości zmian. Zestawienie możliwych stanów sygnału wyjściowego w modulacji fazy jest prezentowane za pomocą diagramu konstelacji. Jeśli liczba możliwych stanów sygnału zmodulowanego jest potęgą dwójki, do wyznaczenia odpowiedniego z tych stanów można użyć kilku bitów. Zwiększenie liczby stanów zapewnia między innymi kwadraturową modulację amplitudy, która łączy w sobie kluczowanie amplitudy i kluczowanie fazy.

Urządzeniem odpowiedzialnym za modulowanie i demodulowanie sygnału jest modem. Dwa modemy zapewniają dupleksową komunikację między jednostkami. Oprócz modemów telefonicznych dostępne są również modemy optyczne i radiowe. Z uwagi na ograniczenie pasma transmisyjnego w modemach telefonicznych wykorzystuje się kwadraturową modulację amplitudy. W modemach V.32 używane są 32 kombinacje fazy i amplitudy. Natomiast w modemach V.32bis liczba kombinacji wynosi 128.

## ZADANIA

- 10.1.** Wymień trzy główne rodzaje modulacji analogowych.
- 10.2.** Czy nośna o częstotliwości 1 Hz może być modulowana sygnałem o częstotliwości 2 Hz? Uzasadnij odpowiedź.
- 10.3.** Wykorzystując twierdzenie Shannona, wyjaśnij, dlaczego użytkowe systemy modulacji amplitudy utrzymują poziom nośnej w pobliżu maksimum.
- 10.4.** Jaka jest różnica między kluczowaniem a modulacją?
- 10.5.** Czy w modulacji fazy można stosować przesunięcia o wartościach  $90^\circ$ ,  $270^\circ$  lub  $360^\circ$ ? Narysuj przykładowy diagram, aby uzasadnić odpowiedź.
- 10.6.** Wyszukaj w internecie diagram konstelacji modulacji 32QAM. Ile punktów zawiera każda ćwiartka układu współrzędnych?
- 10.7.** Na rysunku 10.9 przedstawiono dupleksowe połączenie z wykorzystaniem czterech przewodów. Które z nich służą do przekazywania danych w każdym z kierunków? Udowodnij, że możliwe jest zrealizowanie tego samego zadania za pomocą trzech przewodów.
- 10.8.** Dlaczego w rozwiązaniu wspomnianym w poprzednim pytaniu preferowane są cztery przewody?
- 10.9.** Jaka jest maksymalna przepływność danych w systemie telefonicznym, którego pasmo przedstawiono na rysunku 10.11, jeśli stosunek sygnału do szumu wynosi 30 dB?

# Zawartość rozdziału

- 11.1. Wprowadzenie 205
- 11.2. Multipleksacja 205
- 11.3. Podstawowe rodzaje multipleksacji 206
- 11.4. Multipleksacja z podziałem częstotliwości (FDM) 206
- 11.5. Zakres częstotliwości w kanale komunikacyjnym 208
- 11.6. Hierarchia FDM 209
- 11.7. Multipleksacja z podziałem długości fali 210
- 11.8. Multipleksacja z podziałem czasu 211
- 11.9. Synchroniczne zwielokrotnienie TDM 211
- 11.10. Ramkowanie w telefonicznych systemach TDM 212
- 11.11. Hierarchia TDM 213
- 11.12. Wada synchronicznego systemu TDM
  - puste szczeliny czasowe 214
- 11.13. Statystyczny algorytm TDM 215
- 11.14. Odwrotna multipleksacja 216
- 11.15. Multipleksacja kodowa 216
- 11.16. Podsumowanie 218

# 11

## *Multipleksacja i demultipleksacja*

### 11.1. Wprowadzenie

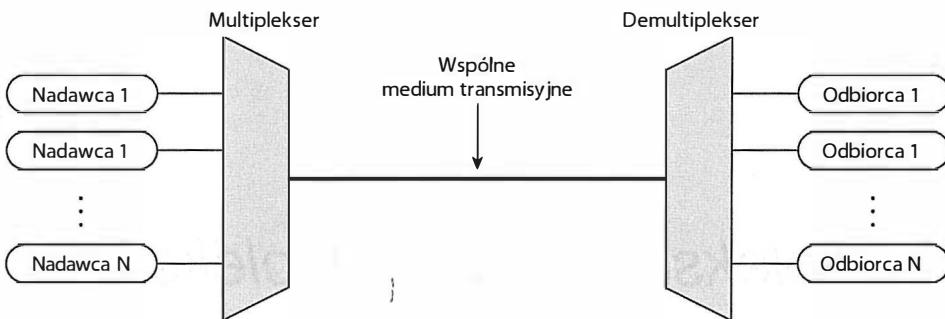
Rozdziały tej części książki zawierają opisy podstawowych komponentów systemu transmisji danych. W poprzednim rozdziale omówiono ideę modulacji i wyjaśniono zasady modulowania nośnej za pomocą analogowych i cyfrowych sygnałów informacyjnych.

W tym rozdziale temat transmisji danych został uzupełniony o zagadnienia związane z multipleksacją. Przedstawiono przyczyny jej stosowania. Opisano także cztery odmiany multipleksacji stosowane w sieciach komputerowych i internecie oraz zasady wykorzystywania zmodulowanych nośnych w wielu mechanizmach zwielokrotniania przepustowości.

### 11.2. Multipleksacja

Określenie **multipleksacja** (zwielokrotnienie) odnosi się do procesu łączenia strumieni danych pochodzących z różnych źródeł w pojedynczy strumień transmitowany we wspólnym medium. Moduł odpowiedzialny za realizację tego zadania nazywa się **multiplekserem**. Operacja odwrotna — wydzielanie pojedynczych strumieni danych ze wspólnego strumienia — jest określana jako **demultipleksacja**, a komponent wydzielający strumienie składowe to **demultiplexer**. Multipleksacja i demultipleksacja nie są związane jedynie z elementami sprzętowymi. Nie muszą również dotyczyć jedynie strumieni bitowych. Pomyśl łączenia i rozdzielania różnych strumieni danych stanowi podstawę funkcjonowania wielu komponentów sieci komputerowej. Sama idea została przedstawiona na rysunku 11.1.

Każdy z nadawców widocznych na rysunku komunikuje się z jednym wybranym odbiorcą. Mimo że każda para wymienia dane w sposób niezależny od innych par, wszyscy użytkownicy systemu korzystają z jednego medium transmisyjnego. Multiplekser łączy informacje dostarczane od różnych nadawców w taki sposób, aby demultiplexer mógł je dostarczyć do właściwych odbiorców.



**Rysunek 11.1.** Multipleksacja umożliwiająca parom nadajnik-odbiornik współdzielenie medium transmisyjnego

### 11.3. Podstawowe rodzaje multipleksacji

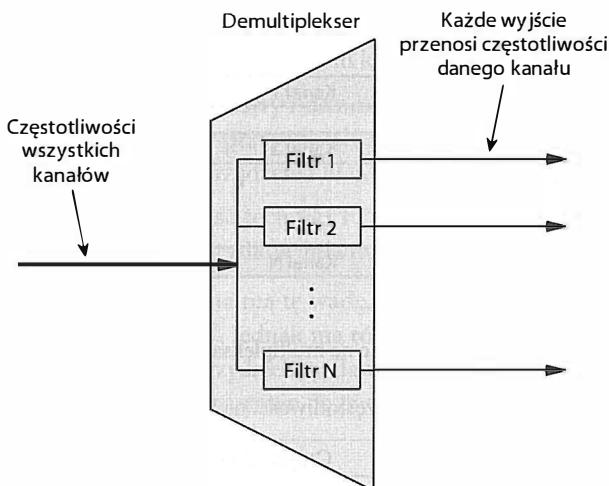
Istnieją cztery podstawowe mechanizmy multipleksacji, z których każdy ma pewne charakterystyczne właściwości i inne obszary zastosowań.

- multipleksacja z podziałem częstotliwości (zwielokrotnienie częstotliwościowe),
- multipleksacja z podziałem długości fali (zwielokrotnienie falowe),
- multipleksacja z podziałem czasu (zwielokrotnienie czasowe),
- multipleksacja kodowa (zwielokrotnienie kodowe).

Najczęściej wykorzystywany formami multipleksacji są multipleksacja z podziałem czasu i multipleksacja z podziałem częstotliwości. Podział długości fali jest pewną odmianą multipleksacji częstotliwościowej, która znajduje zastosowanie w transmisjach z wykorzystaniem włókien światłowodowych. Multipleksacja kodowa jest rozwiązaniem typowo matematycznym, używanym w telefonii komórkowej.

### 11.4. Multipleksacja z podziałem częstotliwości (FDM)

**Multipleksacja z podziałem częstotliwości** (FDM — ang. *Frequency Division Multiplexing*) jest wyjątkowo łatwa do zrozumienia, ponieważ jest ona podstawą działania wszystkich rozgłośni radiowych. Zasady działania mechanizmu wynikają z reguł fizycznych rządzących transmisją radiową. Grupa stacji radiowych może jednocześnie emitować fale elektromagnetyczne bez zakłócania swoich przekazów, jeśli każda ze stacji korzysta z oddzielnego **kanału** (tj. oddzielnej częstotliwości nośnej). Analogiczne rozwiązania są stosowane w transmisji danych. Systemy nadawcze generują wiele jednoczesnych fal nośnych, przesyłanych następnie w pojedynczym przewodzie, lub korzystają z fal o różnych długościach podczas przekazywania światła we włóknie światłowodowym. Po stronie odbiorczej demultiplexer uruchamia filtry, które przepuszczają pewne częstotliwości wokół częstotliwości nośnej. Budowa demultipleksera została przedstawiona na rysunku 11.2.



Rysunek 11.2. Zasada demultipleksacji FDM, w której zestaw filtrów przepuszcza częstotliwości pojedynczych kanałów i tłumia pozostałe

Zgodnie z koncepcją FDM filtry służą jedynie do weryfikowania częstotliwości. Jeśli nadawca i odbiorca pracują na tej samej częstotliwości nośnej, mechanizm FDM wydzieli ją spośród innych częstotliwości, nie zmieniając w jakikolwiek sposób charakterystyki sygnału. Dzięki temu można stosować dowolny rodzaj modulacji opisany w rozdziale 10.

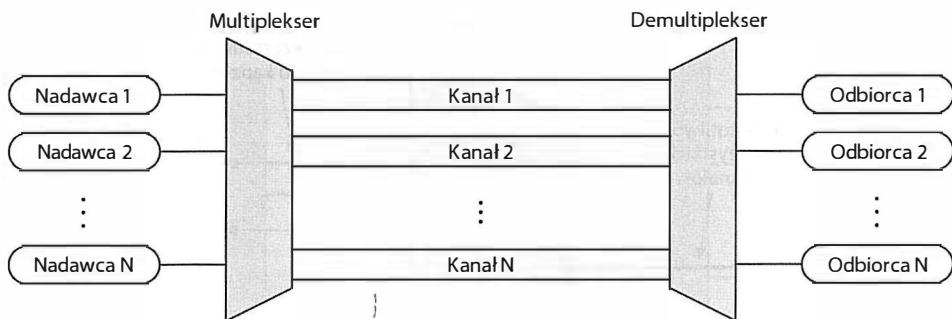
Najważniejsze jest to, że:

*Brak interferencji pomiędzy nośnymi o różnych częstotliwościach sprawia, że mechanizm multipleksacji z podziałem częstotliwości wydziela niezależny kanał komunikacyjny dla każdej pary nadawca-odbiorca, w którym można stosować dowolne techniki modulacji.*

Główną zaletą techniki FDM jest to, że wiele jednostek może w danym czasie korzystać ze wspólnego medium transmisyjnego. W pewnym sensie system FDM zapewnia osobny kanał komunikacyjny każdej parze urządzeń, a sama transmisja nie różni się niczym od przekazywania informacji w niezależnym fizycznym medium transmisyjnym. Rysunek 11.3 jest ilustracją do opisywanego zagadnienia.

Oczywiście, w każdym użytkowym systemie FDM istnieje pewna ograniczona liczba częstotliwości, które wyznaczają kanały komunikacyjne. Gdyby częstotliwości dwóch kanałów były zbyt zbliżone do siebie, mogłyby występować zakłócenia. Demultiplexery odbierające sygnały zbiorcze muszą być zdolne do rozdzielania sygnału na poszczególne nośne. Zapewnieniem odpowiednich odstępów częstotliwościowych między kanałami poszczególnych rozgłośni radiowych zajmuje się Urząd Komunikacji Elektronicznej. Projektanci systemów transmisji danych stosują te same techniki doboru częstotliwości oraz definiowania przerw (nazywanych **pasmami ochronnymi**) między kanałami.

Przykładowy schemat alokacji kanałów został zaprezentowany w tabeli 11.1. Wydzielono w nim 6 kanałów o szerokości 200 kHz każdy oraz pasma ochronne o szerokości 20 kHz.

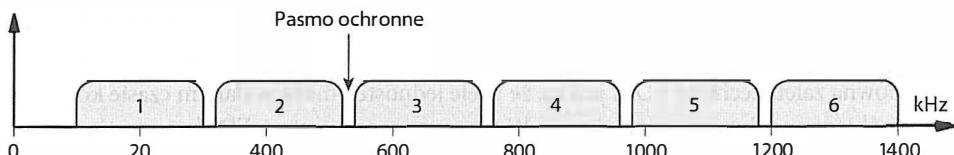


Rysunek 11.3. Ogólne spojrzenie na multipleksację z podziałem częstotliwości

Tabela 11.1. Przykład podziału częstotliwości na kanały z pasmami ochronnymi

Kanał	Częstotliwości
1	100 kHz – 300 kHz
2	320 kHz – 520 kHz
3	540 kHz – 740 kHz
4	760 kHz – 960 kHz
5	980 kHz – 1180 kHz
6	1200 kHz – 1400 kHz

Pasma ochronne są wyraźnie widoczne po zaprezentowaniu kanałów na wykresie częstotliwościowym. Graficzna reprezentacja podziału z tabeli 11.1 jest pokazana na rysunku 11.4.



Rysunek 11.4. Alokacja kanałów częstotliwościowych zdefiniowanych w tabeli 11.1 z uwzględnieniem pasm ochronnych

## 11.5. Zakres częstotliwości w kanale komunikacyjnym

Skoro fala nośna ma określoną częstotliwość, dlaczego w przedstawionym wcześniej przykładzie schemat alokacji definiował przedziały częstotliwości? Aby zrozumieć przyczyny takiego działania, trzeba najpierw zastanowić się nad właściwościami rozwiązania FDM:

- Długi okres dostępności. Technika FDM została opracowana przed dzisiejszymi systemami transmisji danych. Koncepcja podziału widma elektromagnetycznego na kanały pojawiła się w czasie pierwszych eksperymentów z rozgłośniami radiowymi.

- Popularność. Technika FDM jest stosowana w pracy stacji radiowych i telewizyjnych, a także w telewizjach kablowych i telefonii komórkowej AMPS.
- Analogowy charakter. Multipleksery i demultipleksery FDM odbierają i dostarczają sygnały analogowe, również w przypadkach, w których nośna została zmodulowana i niesie informacje cyfrowe. Urządzenia FDM traktują nośną jak sygnał analogowy.
- Uniwersalność. Uniwersalność techniki FDM wynika z filtrowania określonego zakresu częstotliwości bez weryfikowania innych właściwości sygnału.

Analogowy charakter rozwiązania ma tę wadę, że zwiększa podatność transmisji na zakłócenia i zniekształcenia sygnału<sup>27</sup>. Jednak ma również zaletę. Jest nią elastyczność systemu. Większość rozwiązań FDM przydziela nadawcy i odbiorcy pewien zakres częstotliwości i pozwala na dowolne ich wykorzystanie. Są dwie przyczyny takiego sposobu działania:

- zwiększenie przepustowości bitowej,
- zwiększenie odporności na zakłócenia.

Aby zwiększyć sumaryczną przepustowość, nadawca dzieli zakres częstotliwościowy kanału na  $K$  nośnych i wysyła  $1/K$  danych za pomocą określonej nośnej. W praktyce oznacza to multipleksację z podziałem częstotliwości, ale w ramach przydzielonego kanału. Taki dodatkowy podział jest czasami określany jako **allokacja subkanałów**.

Z kolei w celu zwiększenia odporności na zakłócenia nadawcy wykorzystują technikę **rozpraszania widma**. Choć istnieje wiele metod rozpraszania widma, podstawowa z nich zakłada, że zakres częstotliwości kanału jest dzielony na  $K$  nośnych. Urządzenia transmitemają wówczas te same dane na wielu nośnych, a odbiorniki wybierają tę kopię, która ma najmniej błędów. Opisane rozwiązanie doskonale sprawdza się w środowiskach, w których istnieje duże prawdopodobieństwo zakłócania sygnałów użytkowych w określonym czasie.

## 11.6. Hierarchia FDM

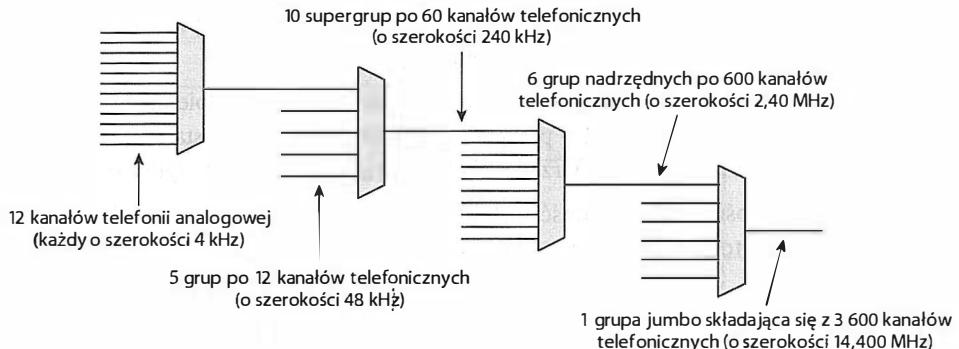
Elastyczność systemów FDM wynika z tego, że urządzenia mają możliwość przesuwania częstotliwości. Jeśli kilka odbieranych sygnałów zajmuje częstotliwości w przedziale od 0 Hz do 4 kHz, multiplekser może pozostawić pierwszy sygnał niezmieniony, ale przenieść drugi na częstotliwości od 4 kHz do 8 kHz, trzeci na częstotliwości od 8 kHz do 12 kHz itd. Technika ta stanowi podstawę hierarchicznego podziału częstotliwości, który umożliwia multiplekserom FDM odwzorowywanie wejść na większy, ciągły zakres częstotliwości. Koncepcja hierarchicznej alokacji częstotliwości w systemie FDM została przedstawiona na rysunku 11.5<sup>28</sup>.

Jak można wywnioskować z rysunku, podstawowy zbiór sygnałów wejściowych składa się z dwunastu sygnałów telefonicznych, z których każdy zajmuje częstotliwość z przedziału od 0 Hz do 4 kHz. Na pierwszym etapie sygnały są multipleksowane w taki sposób, aby

---

<sup>27</sup> Systemy transmisji danych wykorzystujące mechanizmy FDM wymagają często stosowania kabla współosiowego, który zwiększa odporność łączna na zakłócenia.

<sup>28</sup> Dodatkowe pasmo jest potrzebne do przenoszenia bitów ramkowania.



Rysunek 11.5. Hierarchia FDM wykorzystywana w telefonii

powstał jeden sygnał **grupowy** o częstotliwościach z przedziału od 0 Hz do 48 kHz. W kolejnym kroku pięć grup podlega dalszej multipleksacji, w której wyniku powstaje supergrupa o częstotliwościach od 0 do 240 kHz itd. Po zakończeniu całej operacji pojedynczy sygnał niesie informacje z 3 600 rozmów telefonicznych. Podsumowując:

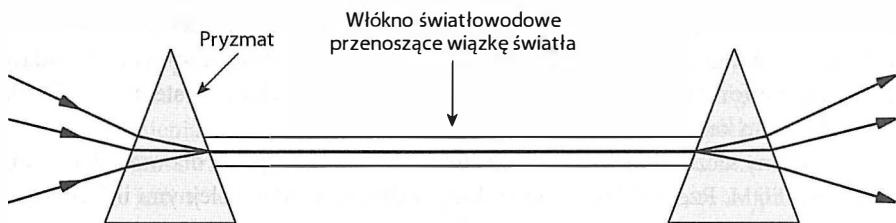
*Możliwe jest opracowanie hierarchicznego mechanizmu multipleksacji z podziałem częstotliwości, w którym na każdym etapie przetwarzania pobierane są sygnały będące sygnałami wynikowymi wcześniejszego etapu.*

## 11.7. Multipleksacja z podziałem długości fali

Określenie **multipleksacja z podziałem długości fali** (WDM — ang. *Wavelength Division Multiplexing*) odnosi się do mechanizmu FDM stosowanego w transmisjach z wykorzystaniem włókien światłowodowych<sup>29</sup>. Operacja WDM polega na zmianie długości fal, oznaczanych grecką literą  $\lambda$  i potocznie nazywanych **kolorami**. Analizę zagadnienia warto rozpocząć od przypomnienia sobie jednej z podstawowych zasad fizycznych dotyczących światła, zgodnie z którą przejście światła przez pryzmat powoduje rozszczepienie promienia na poszczególne kolory składowe. Oczywiście, pryzmat można zastosować również do wykonania odwrotnej operacji — oświetlenie pryzmatu promieniami o odpowiednich kolorach i padającymi pod określonym kątem powoduje utworzenie wiązki światła białego. Postrzegane przez ludzi kolory są tak naprawdę światłem o określonej długości fal.

Pryzmat jest podstawowym narzędziem multipleksacji i demultipleksacji optycznej. Multiplekser zbiera światło o różnych długościach fal i łączy je w pojedynczą wiązkę. Z kolei demultiplekser za pomocą pryzmatu wydziela światło o określonych długościach fal. Rysunek 11.6 stanowi ilustrację do opisanego mechanizmu.

<sup>29</sup> Często stosowane jest również określenie **gęstej multipleksacji z podziałem długości fali** (DWDM — ang. *Dense Wavelength Division Multiplexing*), odnoszące się do rozwiązań, w których wykorzystuje się wiele długości fal światła.



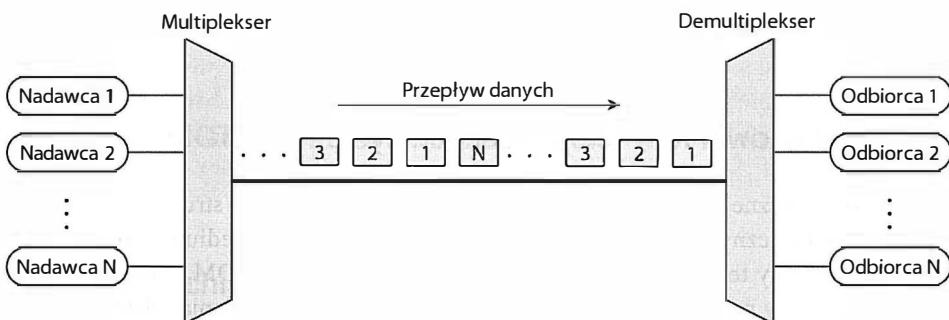
Rysunek 11.6. Łączenie i rozszczepianie światła o różnych długościach fal za pomocą pryzmatu

Najważniejsze jest to, że:

*Zastosowanie multipleksacji z podziałem częstotliwości w transmisjach światłowodowych polega na użyciu pryzmatów, które łączą lub wydzielają światło o określonych długościach fali. Działanie takie jest nazywane multipleksowaniem z podziałem długości fali.*

## 11.8. Multipleksacja z podziałem czasu

Główym rozwiązańem konkurencyjnym w stosunku do FDM jest **multipleksacja z podziałem czasu** (TDM — ang. *Time Division Multiplexing*). Działanie mechanizmu TDM nie odnosi się do szczególnych właściwości energii elektromagnetycznej, co jest charakterystyczne dla techniki FDM. Multipleksacja w dziedzinie czasu sprowadza się do naprzemiennego nadawania informacji z kolejnych źródeł. Rozwiązanie to zostało przedstawione na rysunku 11.7.



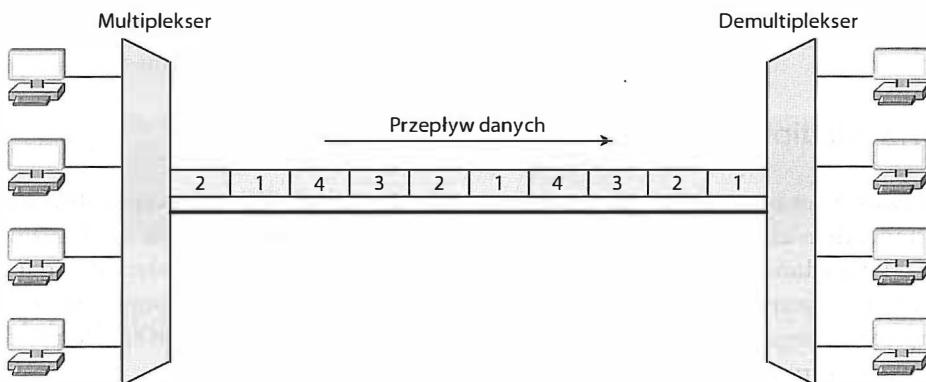
Rysunek 11.7. Multipleksacja z podziałem czasu, dzięki której informacje z różnych źródeł są przesyłane za pośrednictwem wspólnego medium

## 11.9. Synchroniczne zwielokrotnienie TDM

Multipleksacja z podziałem czasu jest powszechnie stosowanym rozwiązaniem internetowym i występuje w wielu odmianach. Rysunek 11.7 należy więc traktować jedynie jako ogólne spojrzenie na omawiane zagadnienia, pamiętając, że poszczególne implementacje

mogą się różnić wieloma szczegółami. Na przykład mechanizm przedstawiony na rysunku działa zgodnie z **algorytmem karuzelowym** (tj. informacje dostarczone przez nadawcę 1 poprzedzają informacje wysypane przez nadawcę 2 itd.). Niektóre systemy TDM wykorzystują algorytm karuzelowy, a inne nie.

Również inny szczegół widoczny na rysunku 11.7 nie jest typowy dla wszystkich rodzajów techniki TDM. Rzeczn dotyczy niewielkiego odstępu między kolejnymi informacjami. Z treści rozdziału 9. wiadomo, że w przypadku transmisji synchronicznej nie są stosowane żadne opóźnienia między transmitowanymi bitami. Zastosowanie w sieciach synchronicznych techniki TDM powoduje, że również między przekazywanymi informacjami nie ma żadnych przerw. Działający w ten sposób system jest nazywany **synchroniczną multipleksacją z podziałem czasu**. Synchroniczne systemy TDM bazują na algorytmie karuzelowym, który wyznacza transmitowane elementy. Na rysunku 11.8 przedstawiono przykład działania synchronicznego mechanizmu TDM w konfiguracji z czterema nadawcami.



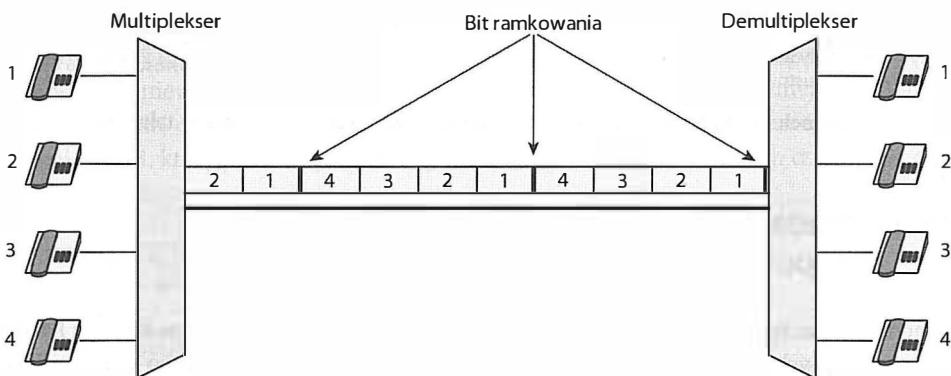
**Rysunek 11.8.** System synchronicznej multipleksacji z podziałem czasu obejmujący czterech nadawców

## 11.10. Ramkowanie w telefonicznych systemach TDM

Systemy telefoniczne korzystają z techniki TDM do zwielokrotniania strumieni cyfrowych rozmów telefonicznych, które są przekazywane przez wspólne medium transmisyjne. W praktyce firmy telekomunikacyjne posługują się akronimem TDM, odnosząc się do szczególnej formy mechanizmu, który jest stosowany do zwielokrotniania cyfrowych połączeń telefonicznych.

W rozwiązańach telefonicznych wykorzystuje się ciekawą technikę utrzymywania synchronizacji między multiplekserem i demultiplekserem. Potrzeba synchronizowania strumieni jest oczywista, jeśli uwzględni się fakt, że nadajniki TDM wysyłają slot za slotem bez dodatkowej informacji o tym, z którym wyjściem powiązane są poszczególne bloki. Ponieważ demultiplekser nie może określić początku szczeliny czasowej, nieznaczna różnica w działaniu zegarów obydwu urządzeń może spowodować błędную interpretację strumienia bitowego.

Aby wyeliminować ryzyko błędnej interpretacji bitów, mechanizmy TDM w systemach telefonicznych uwzględniają dodatkowy **kanał ramkowania**. Jednak zamiast przeznaczania na synchronizację oddzielnej szczeliny czasowej w każdej iteracji algorytmu karuzelowego do strumienia wyjściowego dodawany jest jeden bit. Demultiplekser wyodrębnia dane z kanału ramkowania i sprawdza, czy odpowiadają one naprzemiennie zmieniającym się bitom 0 i 1. Jeśli w wyniku błędu demultiplekser zgubi jeden bit, to najprawdopodobniej pomyłka zostanie zauważona w kanale ramkowania, co z kolei spowoduje ponowne przesłanie informacji. Sposób wykorzystania bitów ramkowania został przedstawiony na rysunku 11.9.



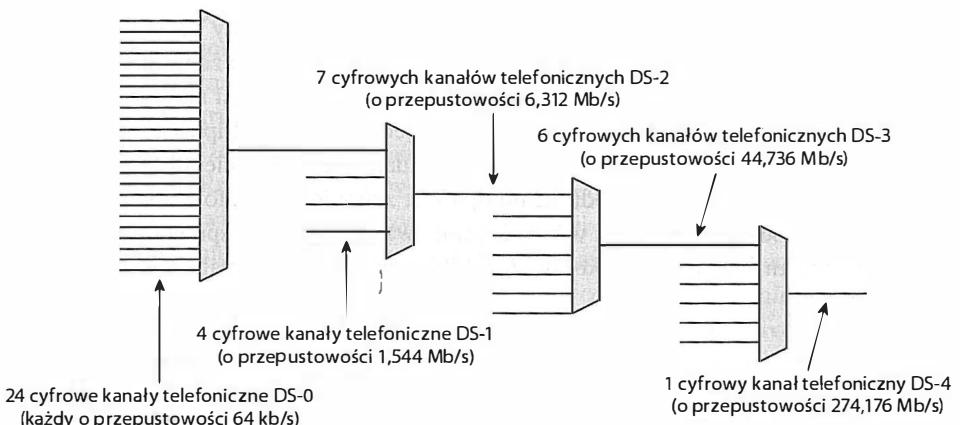
Rysunek 11.9. Synchroniczny system TDM stosowany w telefonii z bitami ramkowania otaczającymi każdą grupę szczelin czasowych

Podsumowując:

*Systemy TDM wykorzystywane do przenoszenia cyfrowych sygnałów rozmów telefonicznych zawierają dodatkowe bity ramkowania, które są wstawiane pomiędzy blokami bitów z jednej iteracji algorytmu karuzelowego. Naprzemienne nadawanie bitów 0 i 1 gwarantuje utrzymanie synchronizacji oraz wykrywanie ewentualnych błędów transmisyjnych.*

## 11.11. Hierarchia TDM

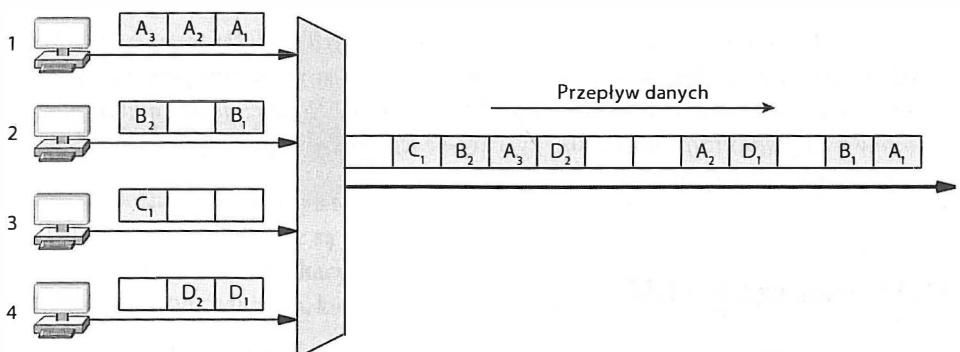
Podobnie jak FDM, również system TDM może mieć strukturę hierarchiczną. Różnica polega jedynie na tym, że zamiast zwielokrotniania częstotliwości, na każdym etapie zwielokrotniania zwiększa się N razy przepływność bitową. Dane są również uzupełniane o dodatkowe bity ramkowania, więc wynikowa przepływność jest nieznacznie większa niż suma przepływności wejściowych strumieni głosowych. Hierarchiczna struktura mechanizmu TDM została pokazana na rysunku 11.10. Warto ją porównać z przykładem multiplikacji FDM przedstawionym na rysunku 11.5.



Rysunek 11.10. Hierarchia TDM stosowana w północnoamerykańskiej telefonii

## 11.12. Wada synchronicznego systemu TDM — puste szczeliny czasowe

Synchroniczne systemy TDM doskonale sprawdzają się w konfiguracjach, w których każde ze źródeł danych generuje strumień o niezmiennej w czasie przepływności o wartości  $1/N$  pojemności wspólnego medium. W cyfrowej sieci telefonicznej rozmowa jest przekazywana ze stałą szybkością 64 kb/s. Z rozdziału 9. wiadomo jednak, że wiele źródeł informacji generuje dane w formie zbitek rozzielanych czasem bezczynności. Taki rodzaj ruchu nie pasuje do synchronicznych systemów TDM. Dowodem na to jest rysunek 11.11.



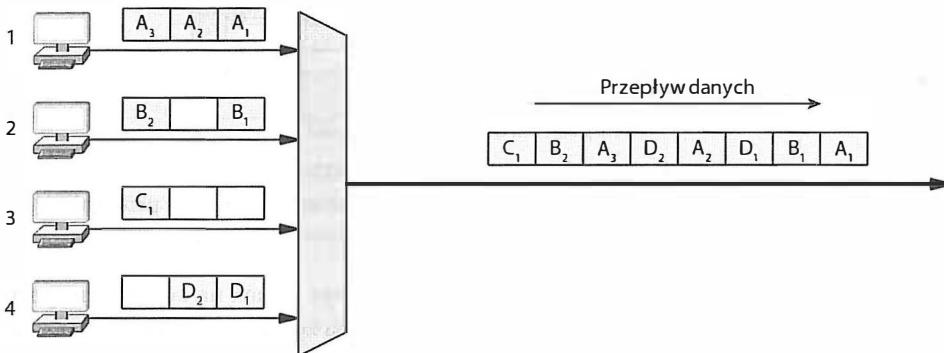
Rysunek 11.11. Pozostawianie niewypełnionych szczelin czasowych systemu TDM w przypadku braku danych do wysłania

Źródła informacji widoczne po lewej stronie rysunku generują dane w przypadkowych odstępach czasowych. Multiplekser synchronicznego systemu musi więc zostawać niewypełnione szczeliny czasowe, ponieważ w chwilach przeznaczonych dla poszczególnych nadajników nie dysponuje żadnymi danymi. W praktyce szczeliny nie mogą zostać puste, ponieważ system transmisyjny musi bezustannie przekazywać jakiekolwiek dane.

Wstawiane są więc w nie wartości zerowe, którym towarzyszy specjalny bit informujący o tym, że wartość jest nieistotna.

### 11.13. Statystyczny algorytm TDM

W jaki sposób można efektywniej wykorzystać współdzielone medium transmisyjne? Jedną z technik całosciowego zwiększenia przepływności bitowej jest **statystyczna multipleksacja z podziałem czasu (statyczne zwielokrotnienie)**<sup>30</sup>. Mimo niejasnego nazewnictwa technika wydaje się niezbyt skomplikowana. Dane do wysyłania są wybierane do wysłania zgodnie z algorytmem karuzelowym, jednak zamiast pozostawiania pustych szczelin czasowych w przypadku braku informacji, przekazywane są dane z kolejnego źródła. Wyeliminowanie niewykorzystywanych szczelin sprawia, że przesłanie tej samej ilości danych zajmuje mniej czasu. Na rysunku 11.12 przedstawiono system statystycznego zwielokrotnienia TDM, który przekazuje dane z rysunku 11.11 w ośmiu szczelinach czasowych, a nie w dwunastu.



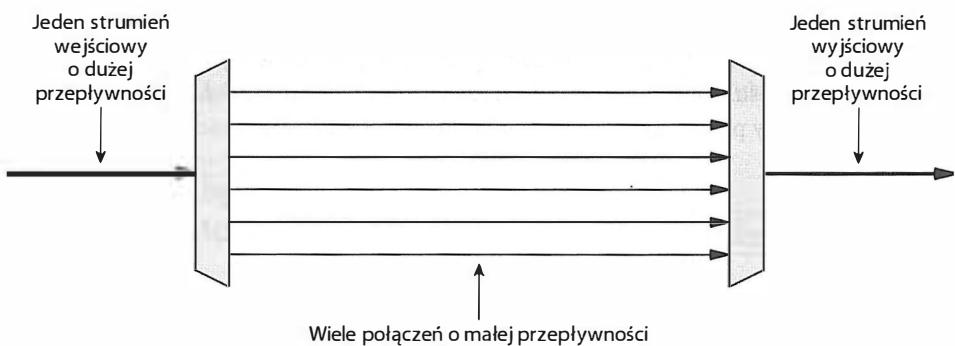
Rysunek 11.12. Statystyczna multipleksacja  
— brak wolnych szczelin czasowych, szybsze przekazywanie danych

Mimo że nie ma wolnych szczelin czasowych, statystyczne zwielokrotnianie wprowadza dodatkowy narzut transmisyjny. Przyczyną jest konieczność przeprowadzenia operacji demultipleksacji. W synchronicznym systemie TDM demultiplekser kojarzy każdą szczelinę czasową z odpowiednim odbiornikiem. W rozwiążaniu statystycznym dane z określonej szczeliny mogą być adresowane do dowolnego odbiorcy. Dlatego poza samymi danymi każdy blok danych musi zawierać identyfikator odbiornika, do którego należy przekazać informację. Techniki identyfikacji odbiorców w statystycznie multipleksowanych sieciach pakietowych i interenie zostały opisane w kolejnych rozdziałach książki.

<sup>30</sup> W niektórych opracowaniach używany jest termin **asynchroniczna multipleksacja z podziałem czasu**.

## 11.14. Odwrotna multipleksacja

Ciekawa technika multipleksacji jest stosowana w przypadkach, w których połączenie między dwoma punktami składa się z wielu mediów transmisyjnych, a żadne z tych mediów nie jest dostatecznie wydajne, aby przenieść strumień o określonej przepływności. Przykładem mogą tutaj być dostawcy usług internetowych, którzy korzystają z połączeń o przepływnościach większych niż dostępne przepustowości łączy. Rozwiązaniem problemu jest odwrócenie zasady multipleksacji — rozszczepienie strumienia o dużej szybkości transmisji na kilka strumieni o niższej przepływności, a następnie połączenie ich po stronie odbiorczej w jeden strumień, zgodnie z rysunkiem 11.13.



Rysunek 11.13. Odwrotna multipleksacja

- rozszczepienie strumienia o dużej przepływności na kilka strumieni o niższej przepływności
- i odtworzenie strumienia pierwotnego po stronie odbiorczej

Budowanie systemu odwrotnej multipleksacji nie polega jednak na odwrotnym połączeniu standardowych multiplekserów. Urządzenia muszą zostać zaprojektowane w taki sposób, aby po obydwu stronach łącza stosowany był ten sam mechanizm rozkładania bitów strumienia wejściowego na poszczególne połączenia składowe (o małej przepustowości). Ponadto, aby zapewnić dostarczenie bitów we właściwej kolejności, system musi poprawnie obsługiwać przypadki, w których opóźnienia transmisyjne poszczególnych łączów składowych są różne. Mimo dużej złożoności rozwiązanie to jest jednak często wykorzystywane w internecie.

## 11.15. Multipleksacja kodowa

Ostatnia technika multipleksacji — **multipleksacja kodowa** (CDM — ang. *Code Division Multiplexing*) — znajduje zastosowanie w telefonii komórkowej oraz niektórych łączach satelitarnych. Szczególną formą CDM jest wykorzystywany w telefonii komórkowej **wielodostęp kodowy** (CDMA — ang. *Code Division Multi-Access*).

W przeciwieństwie do FDM i TDM, rozwiązania CDM nie odnoszą się do żadnych parametrów fizycznych sygnału (częstotliwości lub czasu). Założenia CDM wywodzą się z interesujących rozważań matematycznych, zgodnie z którymi ortogonalne przestrzenie

wektorowe można łączyć i rozdzielać bez ryzyka wystąpienia wzajemnych zakłóceń. Zasada działania mechanizmu jest znacznie łatwiejsza do zrozumienia, jeśli analizuje się ją na przykładzie konkretnej techniki stosowanej w telefonii komórkowej. W rozwiązaniu tym każdemu nadawcy przypisywany jest niepowtarzalny kod binarny  $C_i$ , nazywany **sekwencją kodową**. Sekwencje kodowe są wybierane w taki sposób, aby stanowiły ortogonalne wektory (tzn. iloczyn skalarny dwóch dowolnych sekwencji wynosi 0). Za każdym razem, gdy nadawca musi przesłać jakąkolwiek wartość ( $V_i$ ), mnoży wartości  $C_i \times V_i$  i przesyła wynik tej operacji. Pozostali nadawcy przekazują swoje strumienie danych w tym samym czasie, co powoduje sumowanie się wartości. Aby zyskać wartość  $V_i$  po stronie odbiorcy, wystarczy przemnożyć tę sumę przez  $C_i$ .

Rozważmy inny przykład. Aby maksymalnie uprościć analizę, przyjmijmy, że sekwencja kodowa składa się jedynie z dwóch bitów, a wartości danych z czterech bitów. Sekwencję kodową należy postrzegać jako wektor. W tabeli 11.2 przedstawiono przykładową listę wartości.

**Tabela 11.2.** Przykładowe wartości wykorzystane w multipleksacji kodowej

Nadawca	Sekwencja kodowa	Wartość danych
A	1 0	1 0 1 0
B	1 1	0 1 1 0

Pierwszy krok polega na przekształceniu wartości binarnych w wektory, w których zero jest reprezentowane przez wartość -1:

$$C_1 = (1, -1) \quad V_1 = (1, -1, 1, -1) \quad C_2 = (1, 1) \quad V_2 = (-1, 1, 1, -1)$$

Wykonanie iloczynów  $C_1 \times V_1$  oraz  $C_2 \times V_2$  daje wyniki:

$$((1, -1), (-1, 1), (1, -1), (-1, 1)) \quad ((-1, -1), (1, 1), (1, 1), (-1, -1))$$

Wartości wynikowe są odwzorowywane za pomocą poziomów emitowanego sygnału. Wynikowy sygnał zbiorczy jest sumą dwóch sygnałów składowych:

$$\begin{array}{r} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ + & -1 & -1 & 1 & 1 & 1 & 1 & -1 \\ \hline 0 & -2 & 0 & 2 & 2 & 0 & -2 & 0 \end{array}$$

Odbiornik traktuje odebraną sekwencję jak wektor, oblicza iloczyn wektorowy odebranego ciągu i sekwencji kodowej, wynik interpretuje jako ciąg bitowy, w którym każda dodatnia wartość odpowiada logicznej jedynce, a każda wartość ujemna jest zamieniana na logiczne zero. Odbiornik 1 wykonuje więc działanie:

$$(1, -1) \cdot ((0, -2), (0, 2), (2, 0), (-2, 0))$$

którego wynikiem jest:

$$((0 + 2), (0 - 2), (2 + 0), (-2 + 0))$$

Zinterpretowanie wyniku jako sekwencji wartości:

$$\begin{array}{cccc} 2 & -2 & 2 & -2 \end{array}$$

prowadzi do wyznaczenia wartości binarnej:

$$\begin{array}{cccc} 1 & 0 & 1 & 0 \end{array}$$

Ciąg 1010 jest poprawną wartością  $V_1$ . W analogiczny sposób odbiornik 2 uzyska wartość  $V_2$  z tej samej transmisji.

Mogłoby się wydawać, że technika CDM nie jest tak użyteczna jak TDM. Rzeczywiście, konieczność stosowania długich sekwencji kodowych powoduje, że bywa ona nieefektywna w rozwiązaniach, w których pracuje niewiele urządzeń nadawczych. Wówczas statystyczny algorytm TDM okazuje się korzystniejszy.

Głównymi zaletami techniki CDM są łatwość skalowania i niewielkie opóźnienie transmisyjne w sieciach o dużym obciążeniu. Druga z wymienionych cech ujawnia się w porównaniu systemu CDM z systemem TDM. Gdy jedna ze stacji systemu TDM wysyła swoje dane, multiplekser musi umożliwić również transmisję  $N-1$  pozostałym stacjom przed każdą kolejną emisją danych z pierwszego urządzenia. Jeśli więc aktywność nadawców jest duża, opóźnienie dostarczania informacji z określonej jednostki może się okazać istotne. Natomiast w rozwiązaniach CDM wszyscy nadawcy wysyłają informacje w tym samym czasie, co przyczynia się do zmniejszenia opóźnienia. Techniki CDM znajdują zastosowanie przede wszystkim w telefonii, w której utrzymanie niewielkiego opóźnienia warunkuje wysoką jakość usługi. Podsumowując:

*Mechanizm CDM zapewnia mniejsze opóźnienia niż statystyczny algorytm TDM w sieciach o dużym obciążeniu.*

## 11.16. Podsumowanie

Multipleksacja jest jednym z najważniejszych rozwiązań w transmisji danych. Umożliwia parom nadajnik-odbiornik wymianę informacji za pośrednictwem współdzielonego medium. Wprowadzaniem danych z różnych nadajników do wspólnego medium zajmuje się multiplekser. Zadanie wydzielania pierwotnych strumieni i dostarczania ich do odbiorców należy do demultipleksera.

Powszechnie stosowane są cztery odmiany multipleksacji: z podziałem częstotliwości, z podziałem czasu, z podziałem długości fali oraz kodowa. Zwielokrotnianie częstotliwościowe (FDM) pozwala na jednoczesną komunikację wielu urządzeń z wykorzystaniem oddzielnego kanału, wyznaczanych przez niezależne częstotliwości fal elektromagnetycznych. Zwielokrotnianie falowe (WDM) jest pewną formą multipleksacji z podziałem częstotliwości stosowaną w transmisji światła we włóknach optycznych.

Zwielokrotnianie czasowe (TDM) jest techniką przesyłania pojedynczych bitów w danym czasie w ramach wspólnego medium. Synchroniczne systemy TDM przekazują informacje bez zbędnych okresów nieaktywności, wykorzystując do tego celu algorytm karuzelowy. Unikanie przerw polega na pomijaniu nadawców, którzy nie generują danych.

W technice CDM wykorzystuje się matematyczne operacje na kodach, które umożliwiają wielu nadawcom przesyłanie informacji w tym samym czasie bez ryzyka zakłóceń. Główną zaletą systemów CDM jest łatwość ich skalowania przy zagwarantowaniu małego opóźnienia.

## ZADANIA

- 11.1. Podaj przykłady multipleksacji w systemach komunikacji innych niż elektroniczne.
- 11.2. Wymień cztery rodzaje multipleksacji.
- 11.3. W jaki sposób wykorzystywane są fale elektromagnetyczne w systemie FDM?
- 11.4. Czym jest pasmo ochronne?
- 11.5. W rozwiązaniach FDM możliwe jest przydzielanie poszczególnym kanałom zakresu częstotliwości. Przy jakim rodzaju modulacji istotne jest stosowanie zakresów częstotliwości?
- 11.6. Opisz koncepcję wykorzystania zakresów częstotliwości do zwiększenia szybkości transmisji danych.
- 11.7. Wyjaśnij zasady podziału kanałów o wyższej przepustowości na kanały niższego rzędu w hierarchicznym systemie FDM.
- 11.8. Jakie zjawiska są wykorzystywane w łączeniu i rozdzieleniu światła o różnych długościach fal w rozwiązaniach WDM?
- 11.9. Czy w systemie TDM musi działać algorytm karuzelowy?
- 11.10. Dlaczego ramkowanie i synchronizacja są istotnymi elementami systemu TDM?
- 11.11. Jaka jest przepływność bitowa strumieni wyjściowych na określonym poziomie hierarchicznego systemu TDM (wyjaśnij zagadnienie, posługując się wartościami liczbowymi i przepływnościami bitowymi wejść)?
- 11.12. Założmy, że N użytkowników korzysta z systemu TDM. Warstwa transportowa może przekazywać K bitów na sekundę. Wyznacz minimalną i maksymalną przepływność bitową, której użytkownik może się spodziewać.
- 11.13. Założmy, że koszt łącza OC-12 stanowi 12 procent kosztu łącza OC-48. Jaki rodzaj multipleksacji może zostać wykorzystany przez dostawcę usług internetowych do obniżenia kosztu przesyłania danych z przepływnością OC-48? Uzasadnij odpowiedź.
- 11.14. Znайдź w internecie informację o długości sekwencji kodowej stosowanej w telefonicznych systemach CDMA.
- 11.15. Czy technika CDM zawsze jest najlepszą spośród czterech podstawowych rodzajów multipleksacji? Uzasadnij odpowiedź.

# Zawartość rozdziału

- 12.1. Wprowadzenie 221
- 12.2. Dostęp do internetu 221
- 12.3. Wąskopasmowe i szerokopasmowe technologie dostępowe 222
- 12.4. Łącze abonenckie i ISDN 223
- 12.5. Technologie cyfrowych linii abonenckich (DSL) 224
- 12.6. Charakterystyka łączego abonenckiego i mechanizmy adaptacyjne 225
- 12.7. Przepustowość łączego ADSL 226
- 12.8. Instalacja ADSL i filtry 227
- 12.9. Modemy kablowe 228
- 12.10. Przepustowość modemów kablowych 228
- 12.11. Instalacja modemu kablowego 229
- 12.12. Sieć HFC 229
- 12.13. Światłowodowe technologie dostępowe 230
- 12.14. Terminologia związana z modemami 231
- 12.15. Technologie dostępu bezprzewodowego 231
- 12.16. Wysokowydajne połączenia rdzenia internetowego 231
- 12.17. Zakończenie obwodu, moduły CSU/DSU i NIU 233
- 12.18. Standardy łączego cyfrowych 234
- 12.19. Standardy DS i ich przepustowości 235
- 12.20. Obwody o największej pojemności (standardy STS) 235
- 12.21. Standardy łączego optycznych 235
- 12.22. Sufiks C 236
- 12.23. Synchroniczna sieć optyczna (SONET) 236
- 12.24. Podsumowanie 238

# 12

## Technologie łączy dostępowych i rdzeniowych

### 12.1. Wprowadzenie

W poprzednim rozdziale przedstawiono jedną z najważniejszych koncepcji w transmisji danych — multipleksację strumieni danych. Opisano w nim mechanizmy podziału czasu i częstotliwości, które są wykorzystywane przez operatorów telekomunikacyjnych do przesyłania sygnałów telefonii cyfrowej.

Tematyka tego rozdziału również należy do grupy zagadnień związanych z transmisją danych, ponieważ dotyczy dwóch obszarów internetu. Pierwszym z nich są łącza dostępowe realizowane z wykorzystaniem modemów telefonicznych, DSL i kablowych, które mają na celu przyłączanie gospodarstw domowych i firm do sieci internetowej. Drugie zagadnienie to wysokoprzepustowe łącza cyfrowe stosowane w rdzeniu internetu. Omawiane wcześniej zagadnienie zwielokrotniania przepływności w systemach telefonicznych zostało tutaj rozszerzone o przykłady obwodów, które są oferowane przez operatorów telekomunikacyjnych odbiorcom biznesowym oraz dostawcom usług internetowych. Omówienie odnosi się przede wszystkim do tych elementów technologii, które są związane z wymianą danych, a szczególnie z multipleksacją i przepływnościami bitowymi.

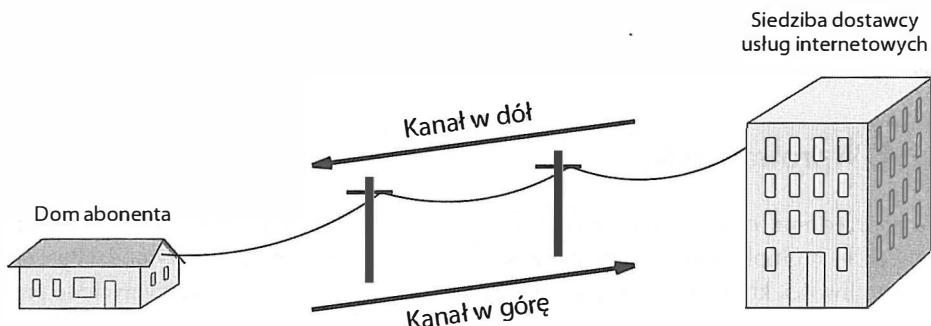
### 12.2. Dostęp do internetu

**Technologia dostępu do internetu** definiuje system wymiany danych, który łączy **abonenta** usługi internetowej (zazwyczaj osobę prywatną lub firmę) z **dostawcą usług internetowych** (ISP — ang. *Internet Service Provider*), czyli z firmą telekomunikacyjną lub operatorem telewizji kablowej. Analizując technologie dostępowe, trzeba pamiętać, że większość użytkowników korzysta z zasobów internetowych w sposób **asymetryczny**.

Zazwyczaj użytkownicy pobierają znacznie więcej danych z internetu, niż wysyłają. Na przykład aby wyświetlić stronę internetową, przeglądarka wysyła adres URL składający się z kilku bajtów. W odpowiedzi otrzymuje dokument, którego rozmiar jest liczony w tysiącach bajtów, a towarzyszące mu rysunki zajmują dziesiątki tysięcy bajtów. Firma utrzymująca stronę obserwuje ruch o odwrotnej charakterystyce.

*Ponieważ typowy użytkownik internetu odbiera więcej informacji, niż wysyła, technologie dostępu do sieci są projektowane w taki sposób, aby w jednym kierunku można było przekazywać więcej danych niż w przeciwnym.*

W przemyśle sieciowym stosuje się pojęcia **kanału w dół** (ang. *downstream*) i **kanału w górę** (ang. *upstream*), opisujące odpowiednio przesyłanie informacji od dostawcy usług internetowych do użytkownika oraz od użytkownika do dostawcy usług. Na rysunku 12.1 pokazano przykład zastosowania wspomnianych terminów.



Rysunek 12.1. Strumienie w góre i w dół w technologiach dostępowych

### 12.3. Wąskopasmowe i szerokopasmowe technologie dostępowe

Dostęp do internetu jest zapewniany z wykorzystaniem wielu różnych technologii. Ogólnie można je podzielić na dwie kategorie, zależnie od oferowanych przepustowości:

- wąskopasmowe,
- szerokopasmowe.

W rozdziale 6. wyjaśniona została różnica między szerokością pasma medium transmisyjnego i przepustowością. Terminologia stosowana w opisie łączyst dostępowych nie uwzględnia jednak takiego podziału. W przypadku rozwiązań sieciowych **szerokość pasma sieciowego** jest tożsama z szybkością transmisji danych. Terminy **wąskopasmowe** i **szerokopasmowe** są doskonałymi tego przykładami.

### 12.3.1. Technologie wąskopasmowe

Określenie **wąskopasmowe** odnosi się zazwyczaj do technologii, które umożliwiają dostarczanie danych z maksymalną przepływnością 128 kb/s. Maksymalna szybkość transmisji danych, którą można uzyskać w ramach połączenia modemowego przy niezaszumionej linii telefonicznej, wynosi 56 kb/s. Połączenia modemowe są więc klasyfikowane jako wąskopasmowe. Podobnie za wąskopasmowe uważa się obwody cyfrowe o niskiej przepustowości oraz niektóre usługi transmisji danych oferowane przez firmy telekomunikacyjne (na przykład ISDN). Zestawienie wąskopasmowych technologii dostępowych zostało przedstawione w tabeli 12.1.

Tabela 12.1. Najważniejsze technologie wąskopasmowe wykorzystywane w dostępie do internetu

Modemowe połączenia telefoniczne
Łącza dzierżawione z modemowymi zakończeniami
Łącza cyfrowe o przepustowości poniżej E1
ISDN i inne usługi transmisji danych oferowane przez operatorów telekomunikacyjnych

### 12.3.2. Technologie szerokopasmowe

Termin **szerokopasmowe** zazwyczaj odnosi się do technologii, które zapewniają wysoką przepustowość. Niemniej precyzyjne określenie granicy między łączami wąskopasmowymi i szerokopasmowymi byłoby bardzo trudne. Wielu ekspertów sugeruje, aby za technologie szerokopasmowe uznawać wszystkie rozwiązania, które gwarantują przepustowość powyżej 1 Mb/s. Z kolei firmy telekomunikacyjne używają terminu „szerokopasmowe” w odniesieniu do wszystkich łącz o przepustowości większej niż połączenie modemowe. Dlatego niekiedy można się spotkać z twierdzeniem, że usługa ISDN (gwarantująca przepływność 128 kb/s) jest usługą szerokopasmową. Najistotniejsze rozwiązania szerokopasmowe zostały wymienione w tabeli 12.2.

Tabela 12.2. Najważniejsze technologie szerokopasmowe wykorzystywane w dostępie do internetu

Technologie DSL
Technologie stosowane w sieciach kablowych
Technologie dostępu bezprzewodowego
Łącza cyfrowe o przepustowości E1 lub większej

## 12.4. Łącze abonenckie i ISDN

Termin **łączce abonenckie** odnosi się do fizycznego połączenia centrali telefonicznej operatora telekomunikacyjnego z siedzibą abonenta. Analizując przeznaczenie łączce abonenckiego, trzeba pamiętać, że jest ono niezależne od pozostałej części systemu telefonicznego.

Choć sam system telefoniczny jest projektowany w taki sposób, aby zapewnić przeniesienie pasma o szerokości 4 kHz (potrzebnego do przeprowadzenia rozmowy), łącze abonenckie, które jest zazwyczaj wykonane w formie skrętki, zapewnia transmisję sygnału o znacznie większej szerokości pasma. Często w niedużych odległościach od centrali telekomunikacyjnej możliwe jest przesyłanie sygnałów o częstotliwościach wyższych niż 1 MHz.

Wraz ze wzrostem popularności transmisji danych operatorzy telekomunikacyjni zaczęli opracowywać sposoby wykorzystania łącz abonenckiego do przekazywania strumieni danych o dużej przepływności. Jednym z pierwszych cyfrowych systemów zaproponowanych przez firmy telekomunikacyjne była sieć cyfrowa z integracją usług (ISDN — ang. *Integrated Services Digital Network*). Użytkownik takiej sieci otrzymuje trzy niezależne kanały cyfrowe oznaczone literami B, B i D (zapisywane często jako 2B+D). Dwa kanały B gwarantują przepustowość 64 kb/s i są przeznaczone do przenoszenia głosu (w formie cyfrowej), danych komputerowych lub skompresowanych sekwencji wizyjnych. Kanał D o przepustowości 16 kb/s jest przeznaczony do zarządzania połączeniem. Użytkownik korzysta z kanału D do wybierania usług realizowanych później w ramach kanałów B (na przykład rozmowy telefonicznej). Kanały B można ze sobą łączyć, tworząc w ten sposób pojedynczy kanał transmisyjny o przepustowości 128 kb/s. W czasie wdrażania technologii ISDN przepływność 128 kb/s wydawała się bardzo duża w porównaniu z wydajnością oferowaną przez modemy telefoniczne. Jednak najnowsze technologie łączy dostępowych zapewniają znacznie większą przepustowość kanałów komunikacyjnych, ograniczając zastosowania ISDN do kilku wybranych aplikacji.

## 12.5. Technologie cyfrowych linii abonenckich (DSL)

**Cyfrowa linia abonencka** (DSL — ang. *Digital Subscriber Line*) jest jedną z głównych technologii gwarantujących wysokowydajną transmisję danych w łączach dostępowych. Rozwiązanie to występuje w kilku wariantach, które zostały wymienione w tabeli 12.3. Z uwagi na różnicę jedynie w pierwszym słowie, cała rodzina technologii często jest ogólnie określana jako rozwiązania **xDSL**.

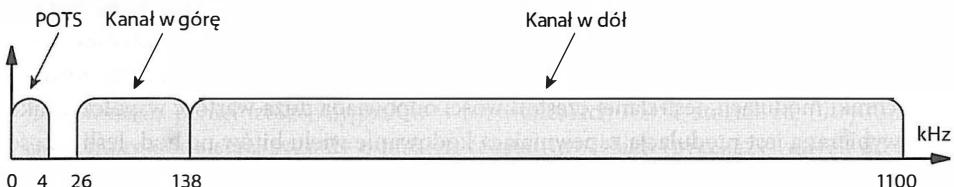
Tabela 12.3. Najważniejsze odmiany technologii DSL wchodzące w skład grupy xDSL

Nazwa	Znaczenie	Zastosowanie
ADSL	Asymetryczny DSL	Odbiorcy prywatni
ADSL2	Asymetryczny DSL wersja 2	Trzykrotnie większa szybkość transmisji niż w ADSL
SDSL	Symetryczny DSL	Firmy wysyłające dane
HDSL	DSL o dużej przepustowości	Firmy oddalone od centrali o mniej niż 2 km
VDSL	DSL o bardzo dużej przepustowości	Rozwiązanie przeznaczone do transmisji z szybkością 52 MB/s

Rozwiązywanie ADSL należy do najczęściej stosowanych form linii abonenckiej. Można je spotkać u większości odbiorców prywatnych. Zastosowana w nim technika podziału częstotliwości sprawia, że pasmo przenoszenia łącza dostępowego jest podzielone na trzy zakresy. Jeden z nich jest przeznaczony do świadczenia tradycyjnej usługi telefonicznej, nazywanej **podstawową usługą telefoniczną starego typu** (POTS — ang. *Plain Old Telephone Service*). Dwa pozostałe zakresy są wykorzystywane w transmisji danych. Najważniejsze jednak jest to, że:

*Z uwagi na zastosowanie mechanizmu podziału częstotliwości wymiana danych i tradycyjna usługa telefoniczna (POTS) mogą być realizowane jednocześnie z użyciem tych samych przewodów.*

Podział pasma w technologii ADSL został przedstawiony na rysunku 12.2.



Rysunek 12.2. Podział pasma w łączu dostępowym wykonany zgodnie z technologią ADSL

Skala na osi X nie jest liniowa. W przeciwnym przypadku przedział 4 kHz zarezerwowany na usługę POTS byłby niewidoczny, podobnie jak pasmo ochronne, ulokowane między obszarem POTS a kanałem w góre.

## 12.6. Charakterystyka łącza abonenckiego i mechanizmy adaptacyjne

Technologia ADSL jest niezwykle skomplikowana, ponieważ każde łącze abonenckie różni się od pozostałych pod względem parametrów elektrycznych. Jego zdolność do przenoszenia sygnałów zależy od odległości między urządzeniami, średnicy przewodów oraz poziomu zakłóceń elektromagnetycznych. Jako przykład można tutaj rozważyć dostarczanie usług do dwóch domów usytuowanych w różnych częściach miasta. Jeśli linia telefoniczna jednego odbiorcy jest ułożona w pobliżu stacji radiowej, emitowane przez tę stację fale radiowe będą przyczyną zakłóceń sygnału na pokrywających się częstotliwościami. Drugi odbiorca nie będzie obserwował skutków zakłócania sygnału na tych samych częstotliwościach (zakładając, że nie mieszka w pobliżu stacji radiowej). Może jednak doświadczać skutków interferencji w innym paśmie częstotliwościowym. Projektanci systemów ADSL nie mogą więc zdefiniować jednego zbioru częstotliwości nośnych lub technik modulacji, uznając, że sprawdzą się w każdych warunkach.

Łącza ADSL mają charakter **adaptacyjny**, co pozwala na uniezależnienie ich od różnic między poszczególnymi łączami. Oznacza to, że po uruchomieniu modemów ADSL testują

łączącą linię abonencką, określając jej charakterystykę transmisyjną. Następnie uzgadniają parametry komunikacji, wybierając ustawienia optymalne dla danego łącza. Urządzenia ADSL wykorzystują **cyfrową modulację wielotonową** (DMT — ang. *Discrete Multi Tone*), która łączy w sobie techniki zwielokrotniania częstotliwościowego oraz odwrotnej multipleksacji.

Zastosowana w rozwiązaniu DMT multipleksacja z podziałem częstotliwości polega na podziale pasma na 286 częstotliwości nazywanych **podkanałami**<sup>31</sup>. Spośród nich 255 jest przeznaczonych do przesyłania danych w dół, a 31 należy do kanału transmisji w góre. Dwa kanały w góre są zarezerwowane do przenoszenia informacji sterujących połączeniem. Teoretycznie urządzenie składa się z niezależnych modemów obsługujących poszczególne częstotliwości nośne. Same nośne są rozmieszczone na osi częstotliwości co 4,1325 kHz i nie zakłócają się wzajemnie. Ponadto, aby wyeliminować ryzyko zakłóceń pochodzących od analogowych sygnałów telefonicznych, nie wykorzystuje się częstotliwości poniżej 26 kHz. W chwili uruchomienia połączenia ADSL obydwa urządzenia końcowe analizują widmo częstotliwościowe, sprawdzając, które częstotliwości gwarantują poprawną pracę, a na których należy się spodziewać zakłóceń. Poza wyborem częstotliwości szacowana jest jakość sygnału w poszczególnych podkanałach i na tej podstawie dobierane są odpowiednie techniki modulacji. Jeśli danej częstotliwości odpowiada duża wartość współczynnika S/N, wybierana jest modulacja zapewniająca kodowanie wielu bitów na bod. Jeśli jakość transmisji na danej częstotliwości jest niska, stosowana jest modulacja o mniejszej liczbie bitów na bod. Podsumowując:

*Z uwagi na zmienną charakterystykę linii abonenckich, w rozwiązaniach ADSL stosuje się mechanizmy adaptacyjne, które umożliwiają modemom przeanalizowanie widma częstotliwościowego, a następnie wybranie odpowiednich częstotliwości nośnych oraz technik modulacji (najlepszych dla danej linii).*

## 12.7. Przepustowość łączy ADSL

Jaka jest maksymalna przepustowość łączy ADSL? W transmisji w dół możliwe jest uzyskanie przepływności 8,448 Mb/s, natomiast kanał w góre zapewnia przenoszenie danych z szybkością do 640 kb/s. Ponieważ jednak obowiązkowy kanał sterujący wymaga przepływności 64 kb/s, efektywna transmisja danych użytkownika nie przekracza 576 kb/s. W przypadku technologii ADSL2 maksymalna szybkość pobierania danych (w najlepszych warunkach) wynosi 20 Mb/s.

Z punktu widzenia użytkownika zastosowanie mechanizmów adaptacyjnych ma pewną interesującą właściwość. Technologia ADSL nie gwarantuje określonej przepustowości. Szybkość transmisji danych jest bowiem determinowana parametrami łącza. Odbiorcy mieszkający w większej odległości od central lub w pobliżu źródeł zakłóceń

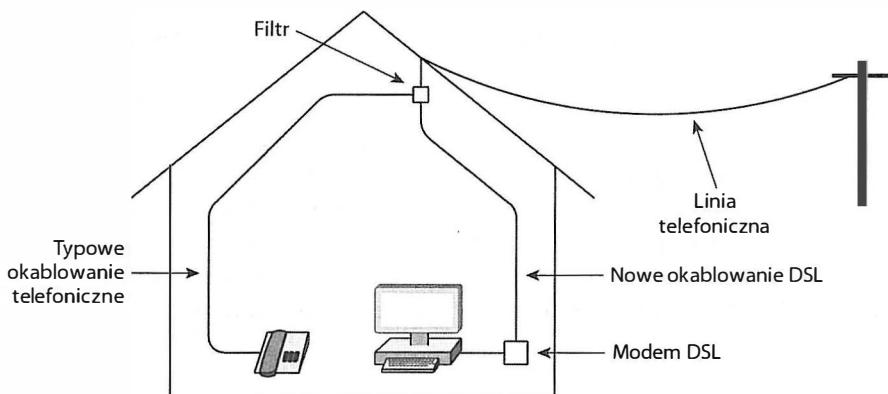
<sup>31</sup> Termin „podkanał” wynika z tego, że w niektórych wariantach DSL stosowany jest podział pasma na zakresy o szerokości 1,544 Mb/s nazywane „kanałami” i odpowiadające przepustowościom linii T1 (więcej informacji na ten temat znajduje się w dalszej części rozdziału).

mogą korzystać z połączeń o niższej przepustowości niż użytkownicy znajdujący się bliżej central oraz ci, których linie dostępowe nie są ułożone w pobliżu źródeł zakłóceń. Efektywna przepływność bitowa w kanale w dół zmienia się w zakresie od 32 kb/s do 8,448 Mb/s, a w kanale w górę od 32 kb/s do 640 kb/s.

Trzeba również pamiętać, że przepustowość łączego ADSL odnosi się jedynie do linii abonenckiej, czyli połączenia między odbiorcą a centralą telefoniczną. Na ostateczną szybkość pobierania danych z sieci wpływ ma wiele dodatkowych czynników. Na przykład jeśli użytkownik pobiera stronę internetową, efektywna przepływność może zostać zmniejszona w wyniku dużego obciążenia serwera, technologii zastosowanej do przyłączenia serwera do internetu oraz wydajności sieci przekazującej dane pomiędzy centralami operatora telekomunikacyjnego.

## 12.8. Instalacja ADSL i filtry

Choć tradycyjna telefonia analogowa zajmuje częstotliwości poniżej 4 kHz, podnoszenie słuchawki może powodować zakłócenia, które będą wpływaly na transmisję DSL. W celu zapewnienia pełnej izolacji sygnałów w rozwiązańach ADSL stosuje się filtry (ang. *splitter*), które oddzielają sygnały o niskich częstotliwościach (przekazywanych do jednego wyjścia) od sygnałów o wysokich częstotliwościach (kierowanych do innego wyjścia). Filtry są komponentami pasywnymi, co oznacza, że nie wymagają zasilania. Zazwyczaj instaluje się je u odbiorcy w obrębie budynku. Do jednego gniazda filtra należy przyłączyć kabel prowadzący do telefonu (lub kabel wewnętrznego systemu telefonicznego), a drugie gniazdo służy do podłączenia modemu ADSL. Stosowny schemat pokazano na rysunku 12.3.



Rysunek 12.3. Podłączenie filtra i instalacja okablowania ADSL

Coraz większą popularność zdobywa pewna odmiana technologii ADSL, nazywana **DSL lite**. Rozwiązanie to nie wymaga instalowania filtra na linii telefonicznej. Całe okablowanie budynkowe może wówczas przenosić sygnały ADSL. Konieczne jest jednak wstawienie filtra między aparatem telefonicznym a okablowaniem budynkowym. Zaletą opisanego rozwiązania jest znacznie łatwiejsze przyłączanie modemów DSL.

## 12.9. Modemy kablowe

Choć technologie ADSL gwarantują transmisję danych z szybkością znacznie większą, niż początkowo zakładano, linia abonencka ma pewne ograniczenia fizyczne. Największym problemem są parametry elektryczne skrętki. Brak ekranowania sprawia, że transmisja jest podatna na zakłóczenia, które w istotny sposób obniżają wydajność połączeń niektórych abonentów. Ponieważ jednocześnie rośnie zapotrzebowanie na łączą o wysokich przepustowościach, konieczne było opracowanie alternatywnych sposobów dostarczania sygnału. Powstało więc kilka przewodowych<sup>32</sup> i bezprzewodowych rozwiązań linii abonenckich.

Szczególnie użyteczne wydają się rozwiązania bazujące na istniejącym okablowaniu **telewizji kablowej**<sup>32</sup>. Medium transmisyjnym stosowanym w tego rodzaju transmisji jest kabel współosiowy, który charakteryzuje się większą szerokością pasma i mniejszą podatnością na zakłóczenia elektromagnetyczne niż skrętka. Systemy telewizji kablowej wykorzystują zwięzłość częstotliwościowe (FDM) do jednocięsnego przekazywania sygnałów wielu stacji.

Nietrudno się domyśleć, że przy dostępności tak wielu kanałów operator telewizji kablowej może wykorzystać pewne z nich do przekazywania cyfrowych informacji do każdego abonenta usługi. Wystarczy więc skonfigurować dwa **modemy kablowe** — jeden w stacji CATV, a drugi po stronie odbiorcy — taki, aby pracowały w danym kanale (tj. na danej częstotliwości), a następnie uwzględnić ten kanał w rozpowszechnianiu wraz z kanałami sygnałów telewizyjnych.

Mimo szerokiego pasma częstotliwościowego systemów CATV zakres przenoszonych częstotliwości nie pozwala na to, aby mechanizm zwięzłostruktury FDM wydzielał osobny kanał dla każdego odbiorcy. Przyczyna takiego ograniczenia jest oczywista. Wystarczy wyobrazić sobie gęsto zaludniony obszar miasta, w którym jeden operator telewizji kablowej ma miliony abonentów. Przeznaczenie osobnego kanału dla każdego odbiorcy nie jest możliwe ze względu na skalę przedsięwzięcia.

Aby rozwiązać ten problem, wykorzystuje się mechanizm będący połączeniem techniki FDM i statystycznej multipleksacji. Jego działanie polega na przydzielaniu kanału komunikacji cyfrowej grupie odbiorców (zazwyczaj kilku odbiorcom sąsiadującym ze sobą). Każdy abonent otrzymuje niepowtarzalny **adres**, który jest dołączany do każdego komunikatu przesyłanego w sieci. Modemy odbiorców prowadzą nasłuch na przypisanych im częstotliwościach, ale przed dostarczeniem komunikatu do użytkownika weryfikują adresy zawarte w odebranych informacjach, porównując je z adresem przypisanym abonentowi.

## 12.10. Przepustowość modemów kablowych

Jaka jest maksymalna szybkość transmisji w modemach kablowych? Teoretycznie systemy telewizji kablowych gwarantują przepustowość do 52 Mb/s w kanale w dół oraz 512 kb/s w kanale w górę. W praktyce te wartości są nieco mniejsze. Po pierwsze dlatego, że podana

<sup>32</sup> Sygnał telewizji kablowej (CATV — ang. *Community Antenna TeleVision*) jest dostarczany do odbiorców za pośrednictwem kabli współosiowych, przy wykorzystaniu techniki FDM.

przepływność odnosi się jedynie do połączenia między mieszkaniem abonenta a budynkiem operatora. Po drugie, pasmo jest dzielone z  $N$  innym odbiorcami (rozmiar wspomnianego wcześniej zbioru jest ustalany przez operatora sieci kablowej). Dla użytkownika sieci współdzielenie pasma z innymi użytkownikami jest wadą rozwiązań, ponieważ efektywna przepustowość łącza do danego odbiorcy zmienia się w czasie. W najgorszym przypadku wynosi  $1/N$  całkowitej przepustowości linii (przy założeniu, że  $N$  odbiorców korzysta z jednej częstotliwości).

## 12.11. Instalacja modemu kablowego

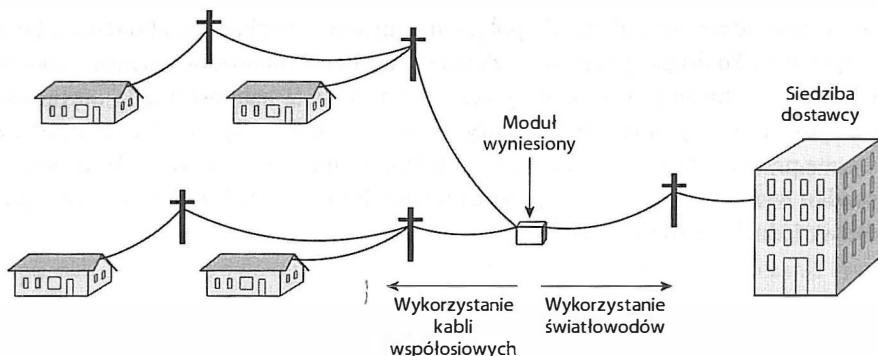
Dzięki zastosowaniu techniki FDM instalacja modemu kablowego nie nastręcza żadnych trudności. W przeciwieństwie do rozwiązań xDSL, które wymagają stosowania filtrów, modemy telewizyjne są przyłączane bezpośrednio do kabla. Komponenty FDM zawarte w gniazdach instalacji telewizyjnej oraz w modemach kablowych gwarantują separację kanałów danych i telewizji.

*Dzięki stosowaniu w systemach kablowych multipleksacji z podziałem częstotliwości modemy kablowe mogą być przyłączane bezpośrednio do istniejącego okablowania, bez konieczności stosowania dodatkowych filtrów.*

## 12.12. Sieć HFC

Jedną z najbardziej obiecujących koncepcji, która zapewnia dostarczanie danych z dużą przepustowością, jest idea sieci hybrydowej, bazującej na światłowodach i kablach współosiowych (HFC — ang. *Hybrid Fiber Coax*). Światłowody są wykorzystywane do łączenia stacji operatora telewizji kablowej. Natomiast kable współosiowe służą do dostarczania sygnałów do odbiorców końcowych. W założeniu system HFC ma strukturę hierarchiczną. W obszarach sieci wymagających dużej przepustowości stosowane są światłowody, a łącza o mniejszej szybkości transmisji są zrealizowane za pomocą kabli współosiowych. Aby wdrożyć opisywane rozwiązanie, operator musi zainstalować w pobliżu odbiorców urządzenie konwertujące sygnały przekazywane za pomocą światłowodów i kabli współosiowych. Urządzenia tego typu są z jednej strony przyłączone do światłowodu operatora telewizji kablowej, a z drugiej strony do kabla współosiowego, który dostarcza sygnał telewizyjny do grupy mieszkań abonentów. Rozwiązanie to zostało pokazane na rysunku 12.4.

Łącza o dużej przepustowości instalowane między siedzibą operatora telewizji kablowej a modułem obsługującym grupę odbiorców nazywa się połączeniami **magistralnymi**. Z kolei odgałęzienia instalacji prowadzące do poszczególnych odbiorców to połączenia **budynkowe**. Odcinki magistralne mogą mieć do 24 km długości. Natomiast długość połączeń budynkowych nie przekracza zazwyczaj 1 km.



Rysunek 12.4. System HFC

## 12.13. Światłowodowe technologie dostępowe

Firmy telekomunikacyjne mają do dyspozycji kilka hybrydowych technologii światłowodowych, a także rozwiązanie, w którym światłowód jest doprowadzany do domu odbiorcy. Zestawienie poszczególnych wersji systemu przedstawiono w tabeli 12.4.

Tabela 12.4. Nazwy technologii dostępowych bazujących na łączach światłowodowych

Skrót	Rozwinięcie
FTTC	ang. <i>Fiber To The Curb</i> — światłowód do krawężnika
FTTB	ang. <i>Fiber To The Building</i> — światłowód do budynku
FTTH	ang. <i>Fiber To The Home</i> — światłowód do mieszkania
FTTP	ang. <i>Fiber To The Premises</i> — światłowód do siedziby

**FTTC.** Rozwiązanie to jest zbliżone w działaniu do HFC, ponieważ wykorzystuje światłowód jako łącze magistralne o dużej przepustowości. Idea systemu polega na doprowadzeniu włókna światłowodowego w pobliże grupy użytkowników i zastosowaniu kabli miedzianych w końcowej części sieci. Łącze FTTC różni się od HFC tym, że odgałęzienie do konkretnego odbiorcy składa się z dwóch kabli, co umożliwia operatorowi świadczenie dodatkowej usługi (na przykład połączeń głosowych).

**FTTB.** Wybierając technologię łącza dostępowego, trzeba ustalić, jakie będzie zapotrzebowanie na pasmo ze strony odbiorców biznesowych i czy rozwiązania uwzględniające kable miedziane (w tym również kable współosiowe) spełnią stawiane im wymagania. Problem ten nie występuje w przypadku zastosowania technologii FTTB (bazującej na łączach światłowodowych), ponieważ zapewnia ona dużą przepustowość kanału w góre.

**FTTH.** Technologia FTTH (alternatywna wobec FTTB) zakłada wykorzystanie w łączu dostępowym włókna światłowodowego, który pozwoli na dostarczanie danych z dużą szybkością do abonentów prywatnych. W rozwiąaniu FTTH przepustowość kanału w góre jest znacznie większa niż w klasycznych instalacjach, ponieważ priorytetem jest tutaj możliwość przekazywania wielu strumieni audiowizualnych.

**FTTP.** Skrót FTTP opisuje ogólne rozwiązanie obejmujące technologie FTTB i FTTH.

## 12.14. Terminologia związana z modemami

Niezależnie od technologii linii abonenckiej każde łącze musi być zakończone modemami — jednym po stronie odbiorcy i drugim po stronie firmy telekomunikacyjnej. Modem umieszczony w centrali nazywa się **modemem czołowym** (ang. *head-end modem*). Natomiast urządzenie pracujące w domu abonenta jest nazywane **modemem końcowym** (ang. *tail-end modem*).

Modemy czołowe nie są niezależnymi urządzeniami. Są implementowane w formie modułów składających się z wielu modemów, które można wspólnie konfigurować i nadzorować. Zbiór modemów czołowych wykorzystywanych przez operatora telewizji kablowej jest skrótnie określany jako CMTS (ang. *Cable Modem Termination System*), czyli system zakończeń połączeń modemowych. Format przesyłanych danych oraz komunikaty żądań dostępu do usług (na przykład płatnych filmów) są opisane w specyfikacji interfejsu systemu przesyłania danych w sieciach kablowych (DOCSIS — ang. *Data Over Cable System Interface Specification*).

## 12.15. Technologie dostępu bezprzewodowego

Choć technologie ADSL i HFC umożliwiają dostarczanie usług cyfrowych do większości odbiorców, nie nadają się do zastosowania w każdych warunkach. Największa trudność z zastosowaniem wymienionych rozwiązań występuje na obszarach wiejskich. W przypadku gospodarstwa oddalonego o kilka kilometrów od najbliższego miasta doprowadzenie łącza ADSL nie jest możliwe ze względu na zbyt dużą odległość. Z kolei telewizja kablowa nie jest usługą szczególnie popularną na wsi.

Nawet w dzielnicach podmiejskich zastosowanie technologii ADSL bywa utrudnione ze względu na rodzaj wykorzystanego okablowania. Przesyłanie sygnału o wysokich częstotliwościach może się okazać niewykonalne w przypadku linii telefonicznych, które zawierają cewki Pupina, odczepy lub regeneratorы. Dlatego nawet na obszarach, na których dana technologia łącza abonenckiego sprawdza się u większości odbiorców, mogą występować problemy z podłączeniem niektórych domostw.

W wyjątkowo trudnych sytuacjach pomocne okazują się technologie dostępu bezprzewodowego. Kilka przykładowych rozwiązań tego typu wymieniono w tabeli 12.5. Z kolei szczegółowe ich omówienie znajduje się w rozdziale 16.

## 12.16. Wysokowydajne połączenia rdzenia internetowego

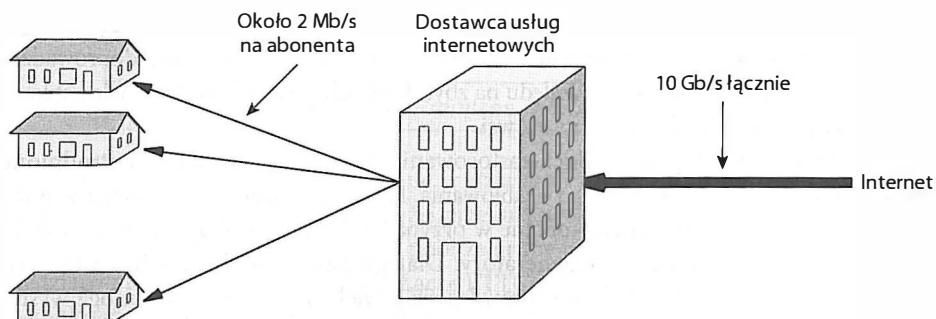
Osoby zajmujące się sieciami twierdzą, że technologie dostępowe rozwiązują **problem ostatniej mili**, czyli problem łącza doprowadzanego do domu abonenta lub siedziby niewielkiej firmy. Opisane technologie łączy dostępowych gwarantują wydajność, która jest

Tabela 12.5. Przykłady technologii dostępu bezprzewodowego

Technologia	Opis
Usługi 3G	Usługi transmisji danych w telefonii komórkowej trzeciej generacji (na przykład HSPA)
WIMAX	Technologia dostępu bezprzewodowego o przepustowości do 155 Mb/s wykorzystująca częstotliwości radiowe
Łącza satelitarne	Wielu dostawców komercyjnych oferuje transmisję danych realizowaną za pośrednictwem satelitów

zadowalającą dla większości odbiorców prywatnych i małych firm (do określania tego rodzaju odbiorców często wykorzystuje się skrót SOHO, pochodzący od angielskich słów *Small Office Home Office*, oznaczających małe biura i biura domowe). Łącza wielkich korporacji oraz połączenia między operatorami telekomunikacyjnymi muszą być jednak znacznie bardziej wydajne. Aby odróżnić tego rodzaju połączenia od przyłącznych charakterystycznych dla zewnętrznych obszarów internetu, stosuje się określenie **rdzeń**, a technologie gwarantujące dużą przepustowość danych nazywa się **technologiami rdzeniowymi**.

Aby wyobrazić sobie, jakie przepustowości obowiązują w rdzeniu, wystarczy rozważyć przykład dostawcy usług internetowych, który ma 5 000 abonentów. Można założyć, że każdy odbiorca dysponuje łączem dostępowym o przepustowości 2 Mb/s. Jakie byłoby zapotrzebowanie na pasmo, gdyby wszyscy użytkownicy chcieli pobierać dane w tym samym czasie? Przykład agregacji ruchu w łączu pomiędzy internetem a dostawcą przedstawiono na rysunku 12.5.



Rysunek 12.5. Agregacja ruchu internetowego przy założeniu, że do sieci przyłączonych jest 5000 odbiorców, z których każdy pobiera dane z przepływnością 2 Mb/s

Nasuwa się więc pytanie, z jakiej technologii może skorzystać dostawca usług internetowych, aby zrealizować długodystansową transmisję danych z przepustowością 10 Gb/s. Odpowiedzą są **cyfrowe obwody punkt-punkt** dzierżawione od operatorów telekomunikacyjnych. Choć wysokowydajne łączę cyfrowe były początkowo przeznaczone jedynie do wewnętrznego wykorzystania w ramach systemu telefonicznego, można je obecnie wydzierżawić do transmisji danych za comiesięczną opłatą. Ponieważ firmy telekomunikacyjne mają uprawnienia do układania okablowania pod ulicami miasta, obwody tego

typu można rozciągać między budynkami, między dwoma punktami w mieście lub nawet między miastami. Pobierana za taką usługę opłata zależy od przepustowości łącza oraz odległości między punktami końcowymi. Podsumowując:

*Cyfrowe obwody dzierżawione od operatorów telekomunikacyjnych stanowią podstawowe elementy, z których buduje się łącza transmisji danych na dużych odległościach. Koszt zależy od pojemności obwodu oraz jego długości.*

## 12.17. Zakończenie obwodu, moduły CSU/DSU i NIU

Aby skorzystać z dzierżawionego łącza cyfrowego, abonent musi się dostosować do reguł obowiązujących w systemie telefonicznym, w tym do standardów, które zostały zaprojektowane z myślą o przenoszeniu głosu w formie cyfrowej. Mogłoby się wydawać, że zachowanie standardów przesyłu informacji cyfrowych nie jest szczególnie trudnym zadaniem. W końcu komputery są urządzeniami cyfrowymi. Jednak branża komputerowa i telefoniczna rozwijały się niezależnie, co spowodowało, że standardy telefonicznych łączy cyfrowych różnią się od połączeń komputerowych. Niezbędne jest więc użycie dodatkowego urządzenia, które stanowi interfejs między komputerem a łączem cyfrowym dostarczonym przez firmę telekomunikacyjną. Urządzenie to składa się z dwóch komponentów — jednostki obsługi kanału i jednostki obsługi danych — umieszczonych w jednej obudowie. Od ich nazw pochodzi skrót CSU/DSU (ang. *Channel Service Unit/Data Service Unit*), którym opisywane jest całe urządzenie. Moduł CSU odpowiada za zakończenie i diagnostykę linii transmisyjnej. Zawiera na przykład mechanizm wykrywania przypadków przerwania połączenia. Realizuje również funkcje **pętli zwrotnej**, która umożliwia przekazywanie kopii wszystkich odbieranych danych z powrotem do nadawcy bez jakiegokolwiek przetwarzania.

Moduł CSU realizuje także funkcję, która informatykom często wydaje się niezrozumiała — zapobiega przesyłaniu długich serii jedynek bitowych. Wyeliminowanie długich ciągów jedynek wynika z charakterystyki sygnałów elektrycznych. Firmy telefoniczne początkowo projektowały łącza cyfrowe z założeniem, że będą one korzystały z kabli miedzianych. Inżynierowie obawiali się więc, że długie serie jedynek będą przyczyną utrzymania przepływu prądu w przewodniku przez zbyt długi czas. Aby uniknąć problemów, należało zastosować kodowanie gwarantujące równowagę napięciową (na przykład kodowanie różnicowe) lub technikę nazywaną **nadziewaniem bitami** (ang. *bit stuffing*), która eliminuje zbyt długie ciągi jedynek.

Komponent DSU przetwarza dane dostarczane do urządzenia CSU/DSU. Zajmuje się przekształcaniem cyfrowych informacji transmitowanych w łączu telekomunikacyjnym na format cyfrowy akceptowany przez komputer użytkownika. Standard interfejsu po stronie komputera zależy od przepustowości samego łącza. Jeśli jest ona mniejsza niż 56 kb/s, wystarczy port RS-232. W przypadku większych przepustowości komputer trzeba wyposażyć w wydajniejszy interfejs sprzętowy (na przykład w komponenty zgodne ze standardami RS-449 lub V.35).

Firma telekomunikacyjna instaluje dodatkowo **moduł interfejsu sieciowego** (NIU<sup>33</sup> — ang. *Network Interface Unit*), który wyznacza granicę między urządzeniami należącymi do operatora telekomunikacyjnego a sprzętem abonenta. Z tego względu jest on również nazywany **punktem demarkacyjnym**.

*Obwód cyfrowy musi być zakończony po obydwu stronach urządzeniami CSU/DSU. Moduł CSU/DSU odpowiada za przekształcanie formatów danych obowiązujących w sieci telekomunikacyjnej na formaty właściwe dla branży informatycznej i odwrotnie.*

## 12.18. Standardy łączy cyfrowych

Wydzierżawiony od firmy telekomunikacyjnej obwód cyfrowy musi spełniać te same standardy transmisji cyfrowej, które są stosowane przez operatora do obsługi rozmów telefonicznych. W Stanach Zjednoczonych standardy obwodów telefonicznych mają nazwy składające się z litery *T* oraz odpowiedniej wartości liczbowej. Inżynierowie nazywają je **standardami grupy T**. Jednym z najbardziej znanych rozwiązań tego typu jest obwód *T1*, który jest odpowiedzialny za przenoszenie danych w łączach dzierżawionych przez niewielkie przedsiębiorstwa.

Niestety, standardy z grupy *T* nie są uniwersalne. Pewna ich odmiana wykorzystywana jest w Japonii. Natomiast w Europie obowiązują inne rozwiązania. Nazwy europejskich standardów rozpoczynają się od litery *E*. Kilka wybranych łączy cyfrowych zostało przedstawionych w tabeli 12.6.

Tabela 12.6. Wybrane obwody cyfrowe i ich przepustowości

Nazwa	Przepustowość	Liczba obwodów głosowych	Region
przepustowość podstawowa	0,064 Mb/s	1	
T1	1,544 Mb/s	24	Ameryka Północna
T2	6,312 Mb/s	96	Ameryka Północna
T3	44,736 Mb/s	672	Ameryka Północna
E1	2,048 Mb/s	30	Europa
E2	8,448 Mb/s	120	Europa
E3	34,368 Mb/s	480	Europa

<sup>33</sup> Niekiedy do opisu modułu NIU wykorzystuje się nazwę Smartjack, która odnosi się do szczególnego rodzaju interfejsu NIU, produkowanego przez firmę Westell Corporation.

## 12.19. Standardy DS i ich przepustowości

Zgodnie z informacjami zamieszczonymi w rozdziale 11. firmy telekomunikacyjne stosują hierarchiczne zwielokrotnianie przepływności, które powoduje zagregowanie wielu kanałów rozmównych w pojedynczy obwód cyfrowy. Przepustowości standardów T i E zostały więc wybrane w taki sposób, aby pozwalały na transmisję wielu strumieni głosowych w tym samym czasie. Warto jednak zwrócić uwagę na to, że pojemność obwodów nie rośnie liniowo wraz ze wzrostem wartości liczbowych w nazwie standardu. Na przykład standard T3 definiuje obwód o pojemności znacznie większej niż trzykrotna pojemność obwodu T1. Trzeba również pamiętać, że firmy telekomunikacyjne oferują obwody o przepustowościach mniejszych niż wymienione w zestawieniu. Są to łącza o ułamkowych przepustowościach standardu T1.

Chcąc zachować techniczną poprawność, należałoby rozróżnić standardy T (charakterystyczne dla systemu przesyłowego) od rozwiązań opisujących zasady zwielokrotniania rozmów telefonicznych w pojedynczym połączeniu. Rozwiązania te są nazywane **standardami poziomów sygnału cyfrowego** (ang. *digital signal level standards*) lub po prostu **standardami DS**. Nazwy konkretnych specyfikacji rozpoczynają się od liter DS, po których zapisuje się liczbę (podobnie jak w standardach grupy T). Na przykład DS1 oznacza usługę, która umożliwia przenoszenie 24 rozmów telefonicznych w jednym łączu, a nazwa T1 wskazuje określony standard wykonania tego łączu. Ponieważ specyfikacja DS1 wyznacza szybkość transmisji, z technicznego punktu widzenia bardziej poprawne jest stwierdzenie, że „obwód pracuje z przepustością DS1”, niż że „obwód ma przepustowość T1”. W praktyce niewiele osób przejmuje się tym rozróżnieniem. Dlatego często można usłyszeć o „przepustowości T1”.

## 12.20. Obwody o największej pojemności (standardy STS)

W terminologii stosowanej przez firmy telekomunikacyjne obwody o dużej pojemności nazywa się **magistralami** lub łączami typu **trunk** (czytaj *trunk*). Działanie tego typu obwodów regulują specjalne standardy z grupy specyfikacji **synchronicznych sygnałów transportowych** (STS — ang. *Synchronous Transport Signal*). W tabeli 12.7 przedstawiono przepustowości definiowane przez poszczególne standardy STS. Wszystkie wartości wyrażono w Mb/s, co ułatwia ich porównywanie. Warto zwrócić uwagę na to, że przepustowości od STS-24 są większe niż 1 Gb/s.

## 12.21. Standardy łączy optycznych

Poza standardami STS firmy telekomunikacyjne operują analogicznymi specyfikacjami **łącz y optycznych** (OC — ang. *Optical Carrier*). Nazwy poszczególnych standardów optycznych zostały uwzględnione w tabeli 12.7. Różnica między specyfikacjami STS i OC polega jedynie na tym, że standardy STS odnoszą się do sygnałów elektrycznych przekazywanych przez interfejsy obwodów cyfrowych (tj. połączenia miedziane), a standardy OC opisują

Tabela 12.7. Przepustowości obwodów cyfrowych zgodnie z hierarchią STS

Nazwa w połączeniach kablowych	Nazwa w połączeniach światłowodowych	Przepustowość	Liczba obwodów głosowych
STS-1	OC-1	51,840 Mb/s	810
STS-3	OC-3	155,520 Mb/s	2 430
STS-12	OC-12	622,080 Mb/s	9 720
STS-24	OC-24	1 244,160 Mb/s	19 440
STS-48	OC-48	2 488,320 Mb/s	38 880
STS-192	OC-192	9 953,280 Mb/s	155 520

sygnały optyczne transmitowane we włóknach światłowodowych. Podobnie jak w przypadku innych terminów sieciowych inżynierowie nie przejmują się tym rozróżnieniem. Nazwa OC-3 oznacza więc często obwód cyfrowy o przepustowości 155 Mb/s niezależnie od tego, czy jest wykonany w formie kabla miedzianego, czy włókna optycznego.

## 12.22. Sufiks C

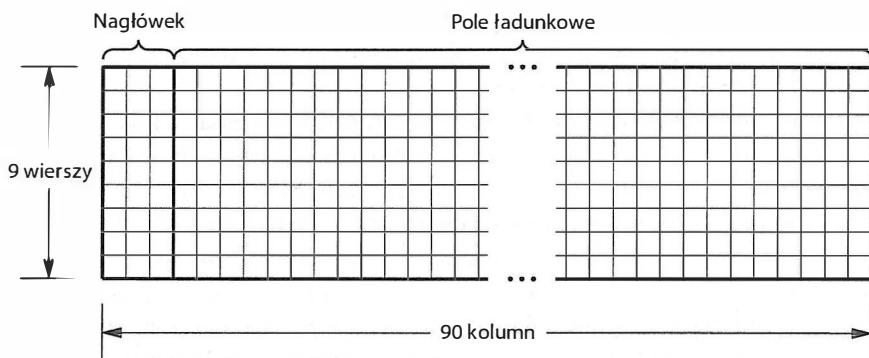
W nazewnictwie stosowanym do opisu synchronicznych sygnałów transportowych oraz łączys optycznych uwzględnia się jeszcze jedną cechę obwodów, która nie została wymieniona w tabeli 12.7. Nazwy obwodów mogą zawierać na końcu dodatkową literę C, która oznacza konkatenację przepustowości (ang. *concatenation*). Występowanie sufiksu informuje o tym, że łącze nie pozwala na odwrotną multipleksację. Obwód OC-3 może w praktyce składać się z trzech obwodów OC-1 o przepustowości 51,840 Mb/s. Może również występować jako pojedynczy obwód OC-3C (STS-3C) działający z szybkością 155,520 Mb/s.

Czy pojedyncze łącze pracujące z pełną przepustowością jest lepsze od kilku obwodów o mniejszych szybkościach transmisji danych? Odpowiedź zależy od sposobu wykorzystania obwodu. Pojedyncze łącze o pełnej przepustowości zapewnia większą elastyczność i eliminuje konieczność stosowania urządzeń odpowiedzialnych za odwrotną multipleksację. Poza tym transmisja danych różni się od transmisji głosu. W systemach telefonicznych obwody o wysokiej przepustowości są przeznaczone do agregowania strumieni o niższych przepływnościach. W sieciach danych operuje się natomiast pojedynczymi strumieniami informacyjnymi. Z tego względu większość projektantów sieci woli stosować obwody OC-3C niż OC-3.

## 12.23. Synchroniczna sieć optyczna (SONET)

Poza specyfikacjami przepustowości (STS i OC) firmy telekomunikacyjne opracowały także spory zbiór standardów transmisji cyfrowych. W Ameryce Północnej jest on znany jako standard **synchronicznych sieci optycznych** (SONET — ang. *Synchronous Optical Network*).

**NETwork**), natomiast w Europie opisuje **hierarchię synchronicznych systemów cyfrowych** (SDH — ang. *Synchronous Digital Hierarchy*). Rozwiązanie SONET definiuje szczegółowe parametry transmisji danych, w tym sposób ich ramkowania, zwielokrotniania (łączenia wielu strumieni o mniejszej przepływności w jeden strumień o dużej przepływności) oraz przekazywania informacji zegarowych wraz z danymi. Z uwagi na powszechnie stosowanie rozwiązań SONET wszelkie odniesienia do obwodów STS-1 należy traktować jak jednocześnie wskazanie standardu SONET jako mechanizmu kodowania w łączu przesyłowym. Na rysunku 12.6 został pokazany format ramki SONET wykorzystywany w obwodach STS-1.



Rysunek 12.6. Ramka SONET stosowana w obwodach STS-1

Każda ramka składa się z 810 oktetów. Zgodnie z założeniami standardu SONET są one podzielone na 9 „wierszy” i 90 „kolumn”. Rozmiar ramki zależy od przepustowości łączą, w którym jest przekazywana. W przypadku obwodu STS-3 rozmiar ten wynosi 2430 oktetów. Z czego wynika różnica? Telefonia cyfrowa bazuje na założeniu, że w każdej sekundzie pobieranych jest 8000 próbek PCM. Oznacza to, że kolejne próbki są pobierane w odstępach co 125 µs. Czas ten wyznacza rozmiar ramki SONET. Szybkość transmisji zgodna ze specyfikacją STS-1 wynosi 51,480 Mb/s. W czasie 125 µs jest więc transmitowanych 6480 bitów, co oznacza, że ramka składa się z 810 oktetów (wartości 8-bitowych). Analogicznie, przepustowość STS-3 narzuca transmisję 2430 oktetów w czasie 125 µs. Największą zaletą uzależnienia rozmiaru ramki od szybkości bitowej jest znaczne uproszczenie mechanizmów synchronicznego zwielokrotniania (zachowania synchronizacji podczas łączenia trzech strumieni STS-1 w jeden strumień STS-3).

Choć większość sieci transmisji danych bazuje na rozwiązaniach SONET w połączeniach punkt-punkt, sam standard umożliwia wdrażanie innych topologii. Szczególnie istotna jest topologia pierścieniowa, która zapewnia odporność na uszkodzenia jednego elementu sieci. Każda ze stacji należących do pierścienia zawiera urządzenie nazywane **multiplekserem add/drop**. Poza przekazywaniem danych w pierścieniu, multiplekser add/drop można skonfigurować w taki sposób, aby wprowadzał dane z lokalnych obwodów do ramek transmitowanych w sieci oraz wydzielał strumienie adresowane do lokalnych jednostek. W przypadku uszkodzenia pierścienia urządzenie wykrywa utratę ramek i wykorzystują drugi pierścień o odwrotnym kierunku transmisji do odtworzenia topologii. Podsumowując:

*Mimo że standard SONET określa sposób budowania wysokowydajnych sieci pierścieniowych przenoszących we włóknach światłowodowych wiele multipleksowanych obwodów danych, większość sieci transmisji danych wykorzystuje rozwiązania SONET jedynie do zdefiniowania mechanizmów ramkowania i kodowania w ramach łączy dzierżawionych.*

## 12.24. Podsumowanie

Technologie łączy dostępowych określają zasady przyłączania odbiorców prywatnych i mniejszych firm do internetu. Istnieje wiele rozwiązań tego typu, w tym połączenia telefoniczne, bezprzewodowe (z użyciem częstotliwości radiowych oraz połączeń satelitarnych) i przewodowe. Dwie technologie najczęściej wykorzystywane w czasie pisania książki to cyfrowe linie abonenckie (DSL) oraz łącza bazujące na modemach kablowych. W połączeniach DSL stosowana jest technika FDM, która umożliwia przekazywanie w kablu łączącym odbiorcę z centralą telefoniczną zarówno danych cyfrowych, jak i analogowego sygnału głosu. Modemy kablowe używają mechanizmu FDM do przesyłania strumieni danych w tym samym kablu współosiowym, w którym przekazywany jest sygnał telewizyjny. Informacje pochodzące od grupy odbiorców korzystających z modemów cyfrowych podlegają statystycznemu multipleksowaniu we wspólnie wykorzystywanym kanale komunikacyjnym.

W takich rozwiązańach jak HFC i FTTC wykorzystuje się włókna światłowodowe, które odpowiadają za dostarczanie danych do punktów znajdujących się w pobliżu grupy odbiorców, skąd są rozprowadzane do poszczególnych abonentów za pomocą kabli współosiowych. Opracowano również nowsze technologie, umożliwiające doprowadzenie światłowodów do domów i mieszkań użytkowników sieci.

Choć standardowe łącza dostępowe są wystarczające dla odbiorców indywidualnych i niewielkich firm, nie gwarantują dostatecznej przepustowości, aby znaleźć zastosowanie w rdzeniu internetu. Aby uzyskać dużą przepływność danych na dużych odległościach, dostawcy usług internetowych, a także większe przedsiębiorstwa dzierżawią od firm telekomunikacyjnych obwody punkt-punkt. Transmisja danych w łączach cyfrowych bazuje na zwielokrotnianiu czasowym (co opisują standardy T w Ameryce Północnej oraz standardy E w Europie). Definicje wysokowydajnych łączy cyfrowych są zawarte w specyfikacjach STS (obowiązujących w Ameryce Północnej) oraz SDH (obowiązujących w Europie). Odpowiadają im standardy z grupy OC, które odnoszą się do połączeń światłowodowych. Wielu inżynierów posługuje się nazwami standardów OC bez względu na rodzaj samego obwodu.

Zasady ramkowania danych w obwodach cyfrowych opisują specyfikacje SONET i SDH. Rozmiar ramki SONET zależy od przepustowości bitowej obwodu. Jedna ramka zawsze zajmuje czas 125 µs. Poza połączeniami punkt-punkt rozwiązania SONET umożliwiają także tworzenie pierścieni, które dzięki odpowiednim urządzeniom są odporne na uszkodzenia i mogą się automatycznie rekonfigurować po wykryciu awarii.

## ZADANIA

- 12.1. Czym jest **technologia łącza dostępowego**?
- 12.2. Dlaczego operatorzy telekomunikacyjni rozróżniają kanały w góre i w dół?
- 12.3. Podaj przykłady wąskopasmowych i szerokopasmowych technologii dostępowych.
- 12.4. Standard ISDN został w pewnym czasie wypromowany przez firmy telekomunikacyjne jako technologia dostępową o dużej przepustowości. Dlaczego liczba instalacji ISDN uległa zmniejszeniu?
- 12.5. Który wariant DSL będzie odpowiedni dla użytkownika, który więcej danych pobiera, niż wysyła?
- 12.6. Jaki rodzaj multipleksacji jest stosowany w połączeniach ADSL?
- 12.7. Dwóch odbiorców mieszka na tej samej ulicy. Obydwaj korzystają z usługi ADSL. Jednak pomiary przepływności strumieni bitowych wykazują, że jeden może pobierać dane z szybkością 1,5 Mb/s, a drugi z szybkością 2,0 Mb/s. Jaka może być tego przyczyna?
- 12.8. Dlaczego w połączeniach DSL stosowany jest filtr?
- 12.9. Który z modemów zapewnia większą przepustowość: modem DSL czy kablowy?
- 12.10. Dlaczego dostawcy usług internetowych wybierają rozwiązań HFC zamiast FTTP?
- 12.11. Gdzie jest umiejscowiony modem czołowy, a gdzie modem końcowy?
- 12.12. Jaka jest przewaga technologii WiMAX nad łączami satelitarnymi? W jakich obszarach łączą satelitarne są lepsze od WiMAX?
- 12.13. Jakie urządzenie jest zainstalowane między linią T1 a komputerem?
- 12.14. Użyj wyszukiarki internetowej, aby ustalić, jaki jest średni rozmiar filmu DVD. Ile czasu zajmie pobieranie filmu przez łącze T1? Ile przez łącze T3? Pomiń nagłówki.
- 12.15. Jeśli ktoś twierdzi, że ma obwód OC-12, wskazując kabel miedziany, jaki błąd popełnia? Podaj poprawne terminy.
- 12.16. Dlaczego projektanci sieci SDH posługują się nietypowymi wartościami przepustowości, zamiast wykorzystywać potęgi dziesiątki?
- 12.17. Wyjaśnij, w jaki sposób wyznaczany jest rozmiar ramki SONET.



# CZĘŚĆ III

## Przełączanie pakietów i technologie sieci komputerowych

**Przegląd technologii pakietowych  
oraz mechanizmów przełączania pakietów  
stosowanych w przewodowych  
i bezprzewodowych mediach transmisyjnych.**

### Rozdziały:

- |  |     |
|--|-----|
| Rozdział 13. Sieci lokalne — pakiety, ramki, topologie                                       | 243 |
| Rozdział 14. Podwarstwa MAC  | 261 |
| Rozdział 15. Przewodowe technologie LAN (Ethernet i 802.3)                                   | 275 |
| Rozdział 16. Technologie sieci bezprzewodowych   | 287 |
| Rozdział 17. Rozszerzenie sieci LAN — modemy optyczne, regeneratorы,<br>mosty i przełączniki | 313 |
| Rozdział 18. Technologie sieci WAN i routing dynamiczny                                      | 325 |
| Rozdział 19. Technologie sieciowe — przeszłość i teraźniejszość                              | 345 |

## Zawartość rozdziału

- 13.1. Wprowadzenie 243
- 13.2. Przełączanie obwodów 243
- 13.3. Przełączanie pakietów 245
- 13.4. Rozległe sieci pakietowe 246
- 13.5. Standardy formatów i identyfikatorów pakietów 247
- 13.6. Model i standardy IEEE 802 248
- 13.7. Sieci punkt-punkt i wielodostępne 250
- 13.8. Topologie sieci LAN 250
- 13.9. Identyfikacja pakietów, demultipleksacja i adresy MAC 252
- 13.10. Adresy w emisji pojedynczej, multiemisji  
i w rozgłoszeniach 253
- 13.11. Rozgłoszenia, multiemisja i efektywne dostarczanie danych  
do wielu jednostek 254
- 13.12. Ramki i proces ich formowania 255
- 13.13. Nadziewanie bajtami i bitami 256
- 13.14. Podsumowanie 257

# 13

## *Sieci lokalne — pakiety, ramki, topologie*

### 13.1. Wprowadzenie

W pierwszej części książki przedstawiono informacje na temat aplikacji internetowych i programowania sieciowego. W drugiej części zaprezentowano zagadnienia związane z transmisją danych (każdy z rozdziałów odnosił się do jednego z kluczowych aspektów funkcjonowania wszystkich sieci komputerowych).

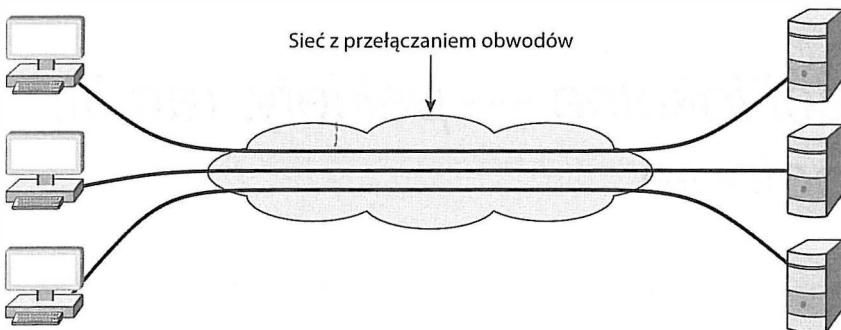
Ten rozdział rozpoczyna kolejną część książki, która jest poświęcona przełączaniu pakietów oraz technologiom budowy sieci komputerowych. Po krótkim wprowadzeniu wyjaśnione zostały w nim model standardów IEEE, koncepcja adresowania sprzętowego oraz identyfikacja ramek.

W kolejnych rozdziałach znajduje się omówienie sposobów wykorzystania pakietów w sieciach rozległych, a także zasad działania różnych przewodowych i bezprzewodowych rozwiązań sieciowych, które odpowiadają za przekazywanie pakietów.

### 13.2. Przełączanie obwodów

Termin **przełączanie obwodów** odnosi się do mechanizmów komunikacji, które wyznaczają trasy między nadawcą i odbiorcą i gwarantują odizolowanie wyznaczonego w ten sposób kanału komunikacyjnego od podobnych kanałów wykorzystywanych przez inne pary nadawca-odbiorca. Przełączenie obwodów jest najczęściej kojarzone z telefonią, gdyż podstawowe zadanie systemu telefonicznego polega właśnie na ustanawianiu dedykowanych połączeń między dwoma aparatami telefonicznymi. Nawet sama nazwa tej techniki

komunikacji pochodzi z wczesnych sieci telefonicznych, w których stosowano elektromechaniczne elementy przełączające odpowiedzialne za ustanawianie fizycznego obwodu elektrycznego. Zasada działania sieci z przełączaniem obwodów została zilustrowana na rysunku 13.1.



**Rysunek 13.1.** Sieć z przełączaniem obwodów, która zapewnia bezpośrednie połączenie między parą komunikujących się urządzeń

Obecnie sieci z przełączaniem obwodów funkcjonują dzięki urządzeniom elektronicznym, które odpowiadają za ustanawianie kanałów komunikacyjnych. Ponadto, wydzielanie niezależnych fizycznych połączeń między urządzeniami zastąpiono mechanizmami jednocześnie obsługującymi wiele obwodów, które są multipleksowane w ramach współdzielonego medium. Powstałe kanały komunikacyjne są nazywane **obwodami wirtualnymi**.

Odróżnianie sieci z przełączaniem obwodów od innych form komunikacji sieciowej nie wynika więc z istnienia niezależnych fizycznych tras przekazywania danych, ale z faktu, że spełniają one trzy cechy tego typu transmisji. Oto wspomniane cechy:

- ustanawianie połączeń punkt-punkt;
- wydzielenie faz tworzenia, używania i rozłączania obwodów;
- zapewnienie wydajności na poziomie właściwym dla niezależnych obwodów fizycznych.

Pierwsza własność oznacza, że obwód jest ustanawiany jedynie między dwoma urządzeniami końcowymi. Z kolei druga cecha odróżnia obwody **przełączane** (tj. ustanawiane na żądanie) od **trwałych** (tj. trwale pozostających w gotowości do wykorzystania). W zarządzaniu przełączanymi obwodami stosuje się trzyetapowe procedury, analogiczne do zestawiania połączeń telefonicznych. W pierwszej fazie połączenie jest ustanawiane. W drugiej urządzenia końcowe wykorzystują obwód do komunikowania się ze sobą. Trzecia faza sprawdza się do rozłączenia połączenia.

Ostatnia z wymienionych właściwości definiuje wyraźną granicę między sieciami z przełączaniem obwodów a innymi rodzajami systemów komunikacyjnych. Przełączanie obwodów oznacza bowiem, że na wymianę danych między dwoma jednostkami końcowymi nie może mieć wpływu komunikacja realizowana przez inne urządzenia sieciowe, nawet jeśli poszczególne strumienie danych są multipleksowane w ramach wspólnego medium. Sieć z przełączaniem obwodów musi więc działać tak, aby użytkownikom końco-

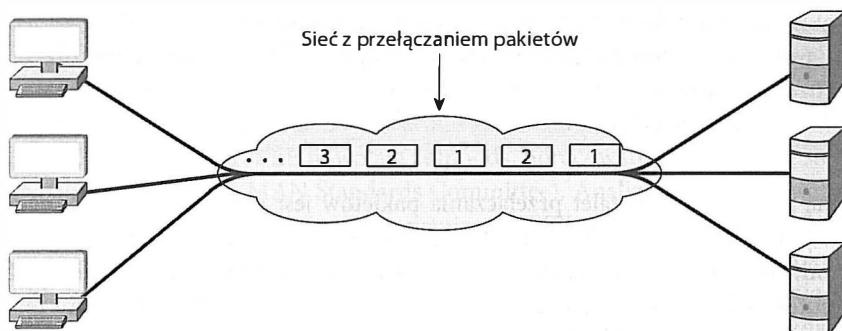
wym wydawało się, że korzystają z niezależnych połączeń fizycznych. Osiągnięcie tego celu jest możliwe tylko wtedy, gdy we współdzielonym medium są stosowane techniki zwielokrotniania częstotliwościowego lub synchronicznego zwielokrotniania czasowego.

Podsumowując:

*Technika przełączania obwodów tworzy iluzję korzystania z niezależnych fizycznych połączeń między komunikującymi się urządzeniami. Trasa taka jest tworzona na żądanie (gdy jest to potrzebne) i usuwana po zakończeniu transmisji.*

### 13.3. Przełączanie pakietów

Najważniejszą alternatywą dla przełączania obwodów jest **przełączanie pakietów**, które stanowi podstawę funkcjonowania internetu. Systemy przełączania pakietów bazują na statystycznej multipleksacji, w której dane z różnych źródeł rywalizują o dostęp do wspólnego medium transmisyjnego. Rozwiązanie to zostało przedstawione graficznie na rysunku 13.2.



Rysunek 13.2. Technika przełączania pakietów,  
która zakłada przesyłanie we wspólnym medium jednego pakietu w danym czasie

Najważniejsza różnica między przełączaniem pakietów a innymi formami multipleksowania statystycznego polega na tym, że nadajnik ma obowiązek podzielenia wejściowej wiadomości na kilka bloków danych nazywanych **pakietami**. Rozmiar pakietów jest różny w różnych implementacjach rozwiązania. W każdej technologii definiowane są jednak maksymalne rozmiary pakietów<sup>34</sup>.

Oto trzy podstawowe własności mechanizmu przełączania pakietów:

- Komunikacja ma charakter asynchroniczny.
- Przed rozpoczęciem wysyłania danych nie jest wymagane ustanowienie połączenia.
- Wydajność transmisji nie jest stała w związku ze stosowaniem statystycznej multipleksacji pakietów.

<sup>34</sup> Pakiety nie mają dużego rozmiaru. Zazwyczaj wynosi on 1500 bajtów.

Z pierwszej własności wynika, że technika przełączania pakietów umożliwia nadawcy komunikowanie się z jednym odbiorcą lub z większą grupą odbiorców. Ponadto wymiana danych może zachodzić w dowolnym czasie, a nadawca może wprowadzać dowolnie długie przerwy między kolejnymi pakietami. Druga cecha stanowi, że w przeciwnieństwie do przełączania obwodów system przełączania pakietów jest ciągle gotowy do przekazywania danych do dowolnych odbiorców. Nadawca nie musi ustanawiać połączenia przed przystąpieniem do generowania pakietów. Nie musi również powiadamiać niskopoziomowych mechanizmów transmisyjnych o zakończeniu komunikacji.

Trzecia własność wskazuje, że w multipleksowaniu poddawane są pakiety, a nie bity lub bajty. Oznacza to, że po uzyskaniu dostępu do medium transmisyjnego nadawca ma obowiązek przesłać cały pakiet. Dopiero po zakończeniu tej operacji inne urządzenia mogą rozpoczęć nadawanie pakietów. Jeśli żaden z nadawców nie jest gotowy do wysyłania danych, jedna stacja może generować pakiety przez dłuższy czas. Jeżeli jednak w kolejce do nadawania oczekuje  $N$  stacji, każda z nich będzie mogła wyemitować około  $1/N$  wszystkich pakietów.

Podsumowując:

*Przełączanie pakietów (stanowiące podstawę funkcjonowania internetu) jest formą statystycznej multipleksacji, w której dozwolona jest komunikacja typu jeden-do-wielu. Nadawca musi jednak podzielić wiadomość na pakiety, a po zakończeniu transmisji każdego z pakietów (przed rozpoczęciem przesyłania kolejnego) musi umożliwić nadawanie innym jednostkom.*

Jedną z największych zalet przełączania pakietów jest obniżenie kosztów działania systemu, wynikające ze współdzielenia łącza. Aby zapewnić komunikację między  $N$  komputerami, sieć z przełączaniem obwodów musi zagwarantować połączenie z każdym z komputerów i dodatkowo co najmniej  $N/2$  niezależnych tras. W przypadku techniki przełączania pakietów niezbędnym jest tyle połączeń, ile jest komputerów, oraz jedna wspólna trasa, wzduż której przekazywane są dane.

### 13.4. Rozległe sieci pakietowe

Kategoryzacja technologii przełączania pakietów wynika z odległości, na których działają poszczególne rozwiązania. Najtańsze spośród nich są sieci przeznaczone do wykorzystania na niewielkich obszarach (na przykład w budynkach). Najdroższe są, oczywiście, systemy obejmujące odległe lokalizacje (na przykład kilka miast). Zestawienie poszczególnych technologii widnieje w tabeli 13.1.

W praktyce wykorzystuje się niewiele technologii sieci MAN i nie stały się one sukcesem komercyjnym. Z tego powodu większość osób zajmujących się sieciami zalicza rozwiązania MAN do grupy WAN, ograniczając podział do dwóch kategorii — LAN i WAN.

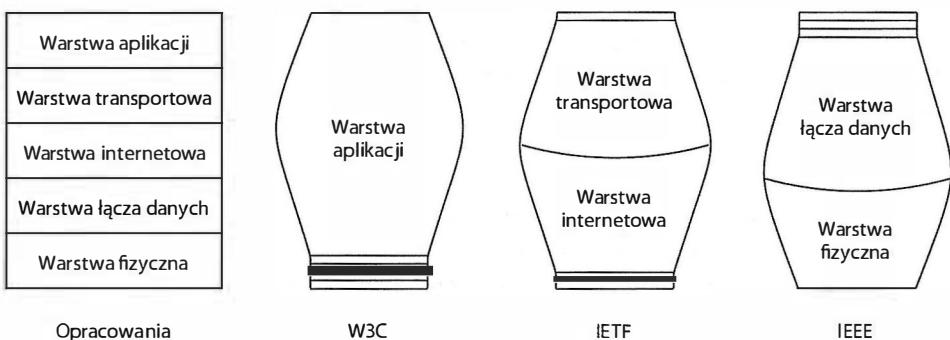
Tabela 13.1. Trzy kategorie sieci z przełączaniem pakietów

Nazwa	Znaczenie	Opis
LAN	Sieci lokalne (ang. <i>Local Area Network</i> )	Najtańsze. Obejmują pojedyncze pomieszczenia lub budynki.
MAN	Sieci metropolitarne (ang. <i>Metropolitan Area Network</i> )	Średni koszt. Rozciągają się na obszarze miast lub zgrupowań miast.
WAN	Sieci rozległe (ang. <i>Wide Area Network</i> )	Najbardziej kosztowne. Łączą wiele miast.

### 13.5. Standardy formatów i identyfikatorów pakietów

Jak wiadomo, systemy przełączania pakietów korzystają ze współdzielonego medium transmisyjnego. Muszą więc oznaczać każdy generowany pakiet tak, aby zawierał identyfikator stacji odbiorczej. Poza tym, aby uniknąć niejednoznaczności, wszyscy nadawcy muszą stosować jednakowy mechanizm identyfikowania odbiorców. Konieczne jest również zapisywanie identyfikatorów w określonych miejscach w pakiecie. Szczegółowe rozwiązania opisanych problemów są zdefiniowane w dokumentacji protokołów opracowanej przez organizacje normalizacyjne. Najpowszechniej stosowany zbiór standardów sieci LAN został przygotowany przez Institute for Electrical and Electronic Engineers (IEEE).

W 1980 r. organizacja IEEE powołała zespół ds. opracowania standardów sieci LAN i MAN (Project 802 LAN/MAN Standards Committee). Analizując przygotowane przez zespół opracowania, trzeba pamiętać, że organizacja IEEE skupia inżynierów zajmujących się przede wszystkim niższymi warstwami stosu protokołów. Dlatego czytając dokumentację IEEE, można odnieść wrażenie, że wszystkie inne aspekty funkcjonowania sieci są nieistotne. Na szczęście, istnieją inne instytucje normalizacyjne, z których każda kładzie szczególny nacisk na jedną z warstw. Na rysunku 13.3 został przedstawiony nieco humorystyczny sposób postrzegania stosu protokołów przez poszczególne organizacje standaryzacyjne.



Rysunek 13.3. Stos protokołów sieciowych w pracach różnych instytucji normalizacyjnych

Nie należy się więc spodziewać, że standardy opracowane przez jedną organizację będą kompletne lub że szczegółowość publikacji będzie proporcjonalna do ważności danej warstwy.

*Każda instytucja normalizacyjna skupia się na wybranych warstwach stosu protokołów. Organizacja IEEE zajmuje się przede wszystkim dwoma najniższymi warstwami stosu i technologii LAN.*

### 13.6. Model i standardy IEEE 802

Aby ułatwić zrozumienie standardów, organizacja IEEE podzieliła warstwę 2. stosu protokołów na dwie **podwarstwy**, zgodnie z tabelą 13.2.

Tabela 13.2. Podział warstwy 2. na podwarstwy zgodnie z modelem IEEE

Nazwa	Znaczenie	Przeznaczenie
LLC	Sterowanie połączeniem logicznym (ang. <i>Logical Link Control</i> )	Adresacja i demultiplesacja
MAC	Sterowanie dostępem do medium (ang. <i>Medium Access Control</i> )	Dostęp do wspólnego medium transmisyjnego

Podwarstwa **sterowania połączeniem logicznym** (LLC) definiuje zasady adresowania i wykorzystania adresów do demultiplesacji. Zagadnienie to zostało opisane w dalszej części rozdziału. Z kolei podwarstwa **sterowania dostępem do medium** (MAC) opisuje sposób korzystania ze wspólnego medium transmisyjnego przez większą liczbę komputerów.

Organizacja IEEE nie używa tekstowych nazw do identyfikacji grup osób pracujących nad standardami oraz tworzonych przez nie dokumentów. Przydziela natomiast identyfikatory o formacie XXX.YYY.ZZZ. Wartość liczbową XXX opisuje kategorię standardu, a część YYY odpowiada podkategorii. Jeżeli dana podkategoria obejmuje wiele specyfikacji, dodawany jest trzeci człon, wyróżniający określone standardy. Na przykład rozwiązania przeznaczone dla sieci LAN są zgrupowane w kategorii 802. Każda grupa opracowująca standard LAN posługuje się więc własnym identyfikatorem, takim jak 802.1, 802.2 itd. Oczywiście, ani wartość 802, ani poszczególne wyznaczniki podkategorii nie mają żadnego znaczenia technicznego — są jedynie wyróżnikami standardów. Przykładowa lista zagadnień objętych pracami organizacji IEEE została przedstawiona w tabeli 13.3.

Jak wynika z rysunku, organizacja IEEE utworzyła wiele grup roboczych, których celem jest opracowanie standardów różnych technologii sieciowych. Grupy składają się z przedstawicieli przemysłu oraz naukowców, którzy na regularnych spotkaniach wytyczają kierunki rozwoju technologii i przygotowują specyfikacje rozwiązań. Grupa pozostaje aktywna, dopóki jej prace postępują, a opracowywane przez nią technologie wydają się użyteczne. Jeśli członkowie grupy uznamą, że określone rozwiązanie nie znajdzie praktycznego zastosowania, prace mogą zostać wstrzymane. Przyczyną takiej decyzji może być na

**Tabela 13.3.** Przykłady identyfikatorów przypisanych przez organizację IEEE do różnych standardów sieci LAN

Identyfikator	Temat
802.1	Protokoły LAN wyższych warstw stosu protokołów
802.2	Sterowanie połączeniem logicznym
802.3	Ethernet
802.4	Token Bus (prace wstrzymane)
802.5	Token Ring
802.6	Sieci metropolitarne (prace wstrzymane)
802.7	Szerokopasmowe sieci LAN wykorzystujące kable współosiowe (prace wstrzymane)
802.9	Sieci LAN z integracją usług (prace wstrzymane)
802.10	Bezpieczeństwo sieci LAN (prace wstrzymane)
802.11	Bezprzewodowe sieci LAN (Wi-Fi)
802.12	Priorytety żądań
802.13	Kategoria 6 — sieci LAN 10Gb/s
802.14	Modemy kablowe (prace wstrzymane)
802.15	Sieci PAN 802.15.1 — Bluetooth 802.15.4 — ZigBee
802.16	Szerokopasmowy dostęp bezprzewodowy 802.16e — Mobilne szerokopasmowe sieci bezprzewodowe
802.17	
802.18	
802.19	
802.20	Mobilny szerokopasmowy dostęp bezprzewodowy
802.21	
802.22	Bezprzewodowe sieci regionalne

przykład pojawienie się korzystniejszej technologii lub opracowanie analogicznego standardu przez inną organizację normalizacyjną. Z tego powodu niektóre prace wymienione w tabeli 13.3 zostały oznaczone jako wstrzymane.

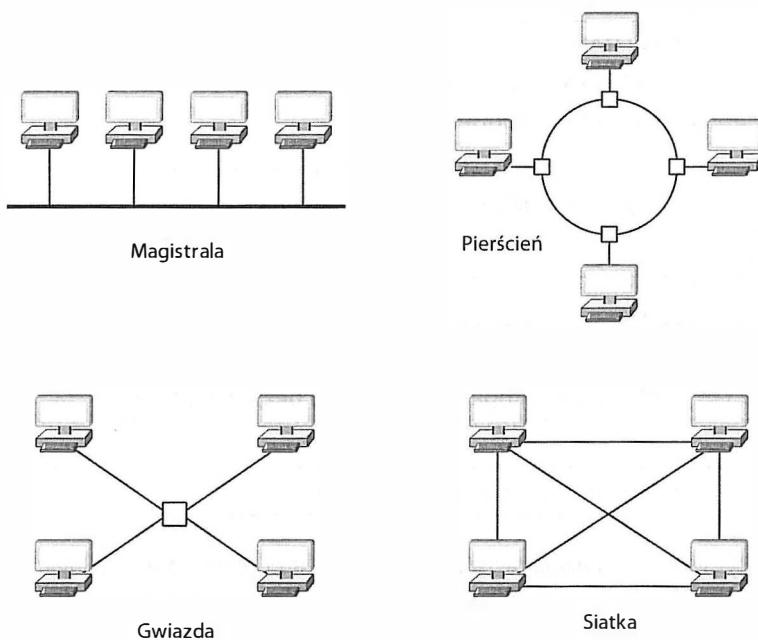
## 13.7. Sieci punkt-punkt i wielodostępne

Zgodnie z wcześniejszą definicją określenie **punkt-punkt** odnosi się do mechanizmów komunikacyjnych, które łączą ze sobą dokładnie dwa urządzenia. Rozwiązania z grupy LAN umożliwiają wielu komputerom współdzielenie medium transmisyjnego w taki sposób, aby każda stacja w sieci mogła wymieniać dane z inną jednostką. Tego rodzaju konfiguracje są określane jako **wielodostępne**, a sama sieć LAN jest wówczas nazywana **siecią wielodostępną**.

W ogólnym ujęciu technologie LAN zapewniają bezpośrednie połączenia między komunikującymi się jednostkami. Inżynierowie często twierdzą, że sieci LAN łączą **komputery**, ze świadomością, że urządzenie takie jak drukarka również może zostać przyłączone do wielodostępnej sieci LAN.

## 13.8. Topologie sieci LAN

Z uwagi na mnogość technologii LAN niezwykle ważna jest umiejętność określania podobieństw i różnic między poszczególnymi z nich. Jednym ze sposobów podziału sieci o podobnych cechach jest klasyfikowanie ich ze względu na **topografię**, czyli ogólny kształt sieci. W tym podrozdziale (na rysunku 13.4) pokazane zostały cztery podstawowe topologie sieci LAN. Ich szczegółowy opis znajduje się w kolejnych punktach.



Rysunek 13.4. Cztery topologie sieci LAN

### 13.8.1. Topologia magistrali

Sieć zbudowana zgodnie z **topologią magistrali** zazwyczaj składa się z pojedynczego kabla, do którego przyłączane są komputery<sup>35</sup>. Każdy przyłączony do magistrali komputer wysyła sygnały, które są odbierane przez wszystkie pozostałe komputery dołączone do tego samego kabla. Dzięki przyłączeniu wszystkich jednostek do wspólnego kabla każde urządzenie może przekazywać dane do dowolnie wybranej stacji docelowej. Oczywiście, niezbędna jest koordynacja takich działań, aby tylko jeden komputer przesyłał informacje w danej chwili.

### 13.8.2. Topologia pierścienia

Sieć bazująca na **topologii pierścienia** składa się z komputerów połączonych w formie zamkniętej pętli — za pomocą kabli pierwszy komputer jest łączony z drugim, drugi z trzecim itd. Ostatnia jednostka jest natomiast łączona z pierwszą. W niektórych rozwiązańach pierścieniowych przyłączenie komputera do sieci wymaga zastosowania oddzielnego urządzenia, które jest elementem wspomnianego pierścienia. Dzięki temu sieć może pracować również wtedy, gdy niektóre z jej komputerów są odłączone. Nazwa topologii (**pierścień**) wynika z tego, w jaki sposób wiele osób wyobraża sobie łączenie elementów sieci ze sobą. Również z rysunku 13.4 wynika, że komputery i łączące je kable tworzą pierścień. W praktyce jednak fizyczne połączenia między jednostkami nie mają nic wspólnego z pierścieniem. Są natomiast układane wzdłuż halli lub prowadzone pionowymi kanałami między poszczególnymi kondygnacjami budynku.

### 13.8.3. Topologia siatki

Sieć zgodna z **topologią siatki** zapewnia bezpośrednie połączenie każdego komputera z każdym innym komputerem sieci. Największą wadą tego rozwiązania jest jego koszt. Sieć siatkowa łącząca  $n$  jednostek wymaga wielu połączeń ( $p$ ). Ich liczbę określa wzór:

$$p = \frac{n!}{(n-2)! \cdot 2!} = \frac{n^2 - n}{2} \quad (13.1)$$

Z powyższego wzoru wynika, że liczba połączeń między komputerami w sieci siatkowej rośnie szybciej niż liczba samych jednostek sieciowych. Z uwagi na wysoki koszt połączeń topologie siatkowe są stosowane w niewielu praktycznych sieciach LAN.

### 13.8.4. Topologia gwiazdy

Sieć ma **topologię gwiazdy**, gdy wszystkie komputery są połączone z pewnym elementem centralnym. Elementem centralnym jest urządzenie elektroniczne (nazywane **koncentratorem**), które odbiera dane od komputerów i dostarcza je do odpowiednich jednostek docelowych.

---

<sup>35</sup> W praktycznych rozwiązaniach na końcach kabla magistralnego przyłączane są terminatory, które zapobiegają odbijaniu się sygnału.

W praktyce sieci gwiazdowe zazwyczaj nie mają symetrycznego kształtu z koncentratorem równo oddalonym od wszystkich stacji końcowych. Często koncentrator jest umieszczony z dala od komputerów, z którymi współpracuje. Komputery są bowiem instalowane w biurach, a koncentratory znajdują się w miejscach łatwo dostępnych dla pracowników obsługi sieci.

### 13.8.5. Przyczyny stosowania wielu technologii

Każde rozwiązanie ma pewne wady i zalety. Topologia pierścienia znacznie ułatwia koordynację dostępu komputerów do sieci i monitorowanie poprawności pracy sieci. Jednak odłączenie lub przecięcie jednego kabla powoduje przerwę w działaniu sieci. Topologia gwiazdy jest odporna na pojedyncze uszkodzenia łączy, gdyż jeden kabel odpowiada tylko za komunikację z jednym komputerem. W topologii magistrali potrzebnych jest mniej kabli niż w rozwiązaniu gwiazdowym, ale podobnie jak w przypadku pierścienia, przerwanie jednego połączenia powoduje wstrzymanie pracy całej sieci. Pozostałe różnice między rozwiązaniami zostały opisane w rozdziałach, które szczegółowo prezentują poszczególne technologie. Na tym etapie rozważań należy zapamiętać, że:

*Sieci podlegają ogólnej kategoryzacji na podstawie sposobu przyłączania urządzeń sieciowych. Choć istnieje możliwość wykonania sieci w topologii siatki, do powszechnie stosowanych topologii zaliczają się jedynie topologie gwiazdy, pierścienia i magistrali. Każda z nich ma pewne wady i zalety.*

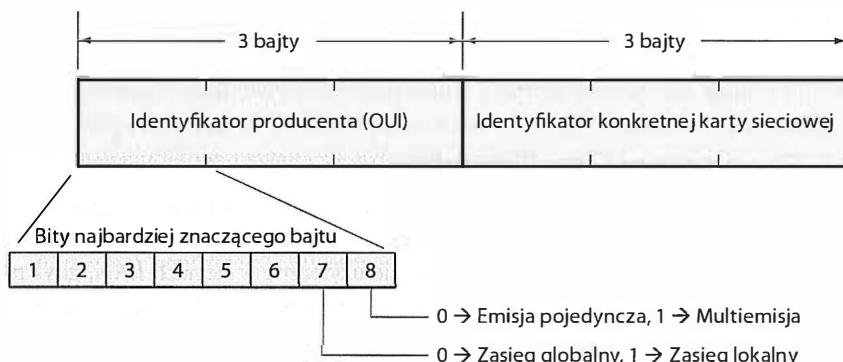
## 13.9. Identyfikacja pakietów, demultipleksacja i adresy MAC

Poza specyfikacjami opisującymi technologie budowy sieci LAN organizacja IEEE opracowała także standard **adresowania**. Analizując techniki adresowania, trzeba pamiętać, że pakiety danych są przekazywane we wspólnym medium, zgodnie z rysunkiem 13.2<sup>36</sup>. W najprostszym przypadku każdy pakiet przesyłany we współdzielonym medium jest przeznaczony dla określonego odbiorcy i tylko jeden z odbiorców przetwarza dany pakiet. W systemie przełączania pakietów demultipleksacja odbywa się na podstawie identyfikatora nazywanego **adresem**. Adres jest przypisany każdemu komputerowi w sieci. Jest również zawarty w każdym pakiecie i wyznacza odbiorcę danych.

W schemacie adresowania zaproponowanym przez organizację IEEE adres składa się z 48 bitów i jest nazywany **adresem mechanizmu sterowania dostępu do medium** (MAC — ang. *Media Access Control*). Ponieważ wywodzi się z technologii Ethernet, często jest nazywany przez inżynierów **adresem ethernetowym**. Aby zagwarantować niepowtarzalność adresów, organizacja IEEE wymusiła przydział innego identyfikatora każdej karcie sieciowej. Każda sprzedawana karta sieciowa (NIC — ang. *Network Interface Card*) dysponuje więc własnym niepowtarzalnym adresem MAC, przypisanym jej przez producenta.

<sup>36</sup> Rysunek 13.2 znajduje się na stronie 245.

Organizacja IEEE nie zajmuje się jednak przydzielaniem pojedynczych adresów. Wyznacza natomiast bloki adresów i udostępnia je pojedynczym producentom. Producenci mogą wówczas we własnym zakresie dbać o niepowtarzalność identyfikatorów wśród oferowanych produktów. Składający się z 48 bitów adres jest podzielony na dwie części po 3 bajty. Pierwsza część odpowiada **niepowtarzanemu identyfikatorowi organizacyjnemu** (OUI — ang. *Organizationally Unique ID*), określającemu producenta urządzenia. Natomiast druga część opisuje konkretną kartę sieciową (NIC). Podział ten został przedstawiony na rysunku 13.5.



Rysunek 13.5. Podział 48-bitowego adresu MAC

Dwa najmniej znaczące bity najstarszego bajtu identyfikatora OUI mają specjalne znaczenie. Najmniej znaczący bit najstarszego bajtu wskazuje, czy adres jest wykorzystywany w emisji pojedynczej (ang. *unicast*) (wówczas ma wartość 0), czy w multiemisji (ang. *multicast*) (wówczas ma wartość 1). Kolejny bit określa, czy wartość OUI jest niepowtarzalna globalnie (1), czy lokalnie (0). Multiemisja została opisana w kolejnym podrozdziale. Adresy o zasięgu globalnym są przydzielane przez organizację IEEE. Adresy lokalne są natomiast przeznaczone do zadań testowych oraz dla instytucji, które muszą operować adresami z niezależnie utworzonej przestrzeni adresowej.

## 13.10. Adresy w emisji pojedynczej, multiemisji i w rozgłoszeniach

Schemat adresowania IEEE definiuje trzy rodzaje adresów, odpowiadające trzem sposobom dostarczania pakietów. Zestawienie rodzajów adresów znajduje się w tabeli 13.4.

Może się wydawać nieco dziwne, że adresy IEEE uwzględniają specjalny bit, na którego podstawie rozróżniane są adresy emisji pojedynczej i multiemisji, a nie umożliwiają oznaczenia adresu rozgłoszeniowego. Standard stanowi jednak, że **adres rozgłoszeniowy** składa się z 48 bitów o wartości 1. Bit multiemisji jest więc również ustawiony na 1. Rozgłoszenia można bowiem interpretować jako szczególną formę multiemisji. Adres multiemisji wyznacza grupę jednostek. Natomiast adres rozgłoszeniowy odnosi się do wszystkich komputerów pracujących w sieci.

Tabela 13.4. Trzy rodzaje adresów MAC i ich przeznaczenie

Rodzaj adresu	Znaczenie w dostarczaniu pakietów
Adres emisji pojedynczej	Jednoznacznie identyfikuje komputer odbiorczy i informuje, że tylko wskazana jednostka powinna otrzymać kopię pakietu.
Adres rozgłoszeniowy	Odpowiada wszystkim komputerom w sieci i informuje, że wszystkie jednostki powinny otrzymać kopię pakietu.
Adres multiemisji	Identyfikuje grupę komputerów w danej sieci i informuje, że każdy komputer z danej grupy powinien otrzymać kopię pakietu.

### 13.11. Rozgłoszenia, multiemisja i efektywne dostarczanie danych do wielu jednostek

Adresy multiemisji i rozgłoszeniowe są szczególnie użyteczne w sieciach LAN, gdyż pozwalały na efektywne dostarczanie danych do wielu komputerów. Ich szczególne znaczenie staje się bardziej oczywiste, gdy uwzględnimy fakt, że transmisja pakietów w sieci LAN odbywa się we współdzielonym medium. W typowej sytuacji każdy komputer przyłączony do sieci monitoruje wspólne medium, wyodrębnia poszczególne pakiety i sprawdza zawarte w nich adresy. Na tej podstawie podejmuje decyzję o dalszym przetwarzaniu pakietu lub jego odrzuceniu. Mechanizm ten został opisany w algorytmie 13.1.

#### Algorytm 13.1. Przetwarzanie pakietów przez jednostki sieci LAN

Cel:

Obsługa pakietu odebranego z sieci LAN

Rozwiążanie:

```

Wyodrębnienie z pakietu adresu docelowego (D).
if (D odpowiada „własнемu adresowi”) {
    Zaakceptowanie pakietu i przekazanie do dalszego
    przetwarzania.
} else if (D odpowiada adresowi rozgłoszeniowemu) {
    Zaakceptowanie pakietu i przekazanie do dalszego
    przetwarzania.
} else if (D odpowiada jednemu z adresów multiemisji,
    związanemu z grupą multiemisji, do której należy dany
    komputer) {
    Zaakceptowanie pakietu i przekazanie do dalszego
    przetwarzania.
} else {
    Odrzucenie pakietu.
}

```

Z przedstawionego algorytmu wprost wynika efektywność transmisji rozgłoszeniowej i multiemisji. Pojedyncza kopia pakietu jest w takich przypadkach przesyłana we wspólnym medium do wszystkich zainteresowanych nią komputerów jednocześnie. Rozważmy

jako przykład transmisję rozgłoszeniową. Zamiast wykonywać  $N$  emisji tego samego pakietu do poszczególnych jednostek, nadawca może wysłać jeden pakiet opatrzony adresem rozgłoszeniowym, który gwarantuje dostarczenie danych do wszystkich działających jednostek.

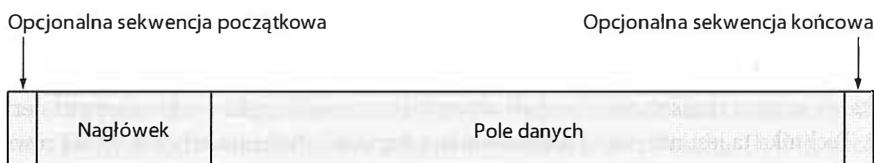
## 13.12. Ramki i proces ich formowania

W rozdziale 9. wprowadzono pojęcie ramkowania, które odnosiło się do synchronicznych systemów komunikacyjnych i opisywało mechanizm umożliwiający odbiornikowi wyznaczenie początku i końca wiadomości. W ogólnym ujęciu termin **ramkowanie** odpowiada operacji formowania wiadomości zgodnie z pewną wstępnie ustaloną strukturą, w której zachowanie właściwej kolejności bitów i bajtów gwarantuje nadawcy i odbiorcy poprawne interpretowanie zawartości komunikatu. W sieciach pakietowych **ramka** odpowiada jednemu pakietowi i składa się z dwóch części:

- nagłówka przechowującego metadane (na przykład adres);
- pola danych zawierającego zasadnicze informacje.

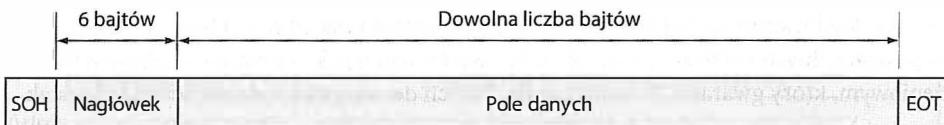
**Nagłówek** ramki obejmuje dane niezbędne do właściwego przetwarzania ramki. Zazwyczaj jest w nim zapisany adres jednostki, do której ramka powinna zostać dostarczona. Obszar **pola danych** zawiera treść przekazywanej wiadomości. Jego rozmiar jest zazwyczaj znacznie większy od rozmiaru nagłówka. W większości technologii sieciowych treść komunikatu nie ma znaczenia dla operacji dostarczania danych. Analizowana jest jedynie zawartość nagłówka. Pole danych może więc zawierać dowolną sekwencję bajtów, która jest istotna jedynie dla odbiorcy i nadawcy.

Ramki są z reguły formowane w taki sposób, aby nagłówek poprzedzał pole danych. Dzięki temu odbiornik może rozpoczęć przetwarzanie ramki przed odczytaniem wszystkich jej bitów. W niektórych rozwiązaniach kolejne ramki są od siebie oddzielane krótkimi sekwencjami startowymi i końcowymi. Stosowny przykład został przedstawiony na rysunku 13.6.



Rysunek 13.6. Struktura typowej ramki w sieci pakietowej

Działanie mechanizmów ramkowania przeanalizujmy na przykładzie transmisji bajtowej. Założymy, że mechanizm dostarczania danych może przesyłać 8-bitowe bajty o dowolnej wartości ze stacji nadawczej do odbiorczej. Przyjmijmy, że pakiet składa się z 6 bajtów nagłówkowych oraz dowolnej liczby bajtów danych. Pojedynczy bajt posłuży również do oznaczenia początku ramki oraz jej końca. W zestawie ASCII występują specjalne znaki początku nagłówka (SOH — ang. *Start Of Header*) oraz zakończenia transmisji (EOT — ang. *End Of Transmission*). Format ramki wynikowej został pokazany na rysunku 13.7.



Rysunek 13.7. Przykład ramki, w której zastosowano znaki SOH i EOT do wyznaczenia początku i końca tejże ramki

Taki format ramki wprowadza pewien niepotrzebny narzut transmisyjny. Uwidacznia się on w przypadku transmitowania dwóch ramek bezpośrednio po sobie. Pierwsza z nich kończy się znakiem EOT, po którym od razu następuje znak SOH wyznaczający początek kolejnej ramki. W praktyce do odseparowania dwóch bloków danych wystarczyłby jeden znak. Mechanizm ramkowania, który uwzględnia znaczniki początku i końca sekwencji bajtowej, generuje zbędne znaki separujące komunikaty.

Generowanie znaków zakończenia ramki staje się uzasadnione w przypadku asynchronicznej transmisji pakietowej w sieci, w której mogą występować błędy. W komunikacji asynchronicznej zastosowanie znaku EOT umożliwia odbiornikowi rozpoczęcie przetwarzania ramki bez konieczności oczekiwania na rozpoczęcie kolejnej ramki. Z kolei w razie wystąpienia błędu znaczniki SOH i EOT ułatwiają synchronizację urządzeń i powrót do normalnej pracy (jeśli nadawca przerwie wysyłanie ramki, odbiorca będzie mógł stwierdzić, że otrzymał jedyne część komunikatu).

### 13.13. Nadziewanie bajtami i bitami

Zgodnie ze standardem ASCII znakowi SOH odpowiada wartość szesnastkowa 0x01, a znakowi EOT wartość 0x04. Co się stanie, jeśli w polu danych wystąpi jeden bajt lub kilka bajtów o wartościach 0x01 lub 0x04? O właściwą obsługę tego typu zdarzeń dba mechanizm **nadziewania bajtami**, który umożliwia przekazywanie dowolnych treści bez obaw o wystąpienie niejednoznaczności w ich interpretacji.

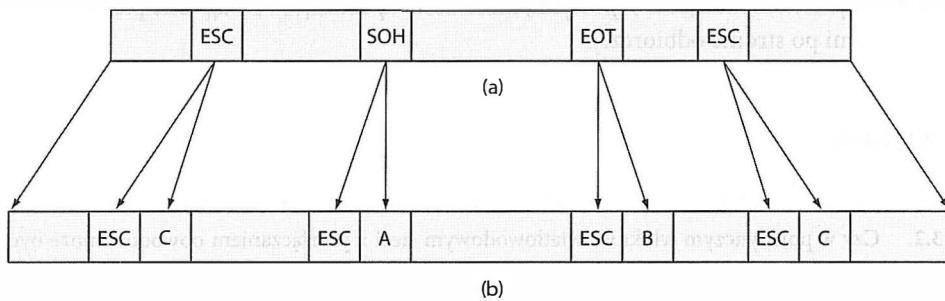
Odróżnienie bajtów danych od informacji sterujących jest możliwe dzięki temu, że nadajnik zastępuje każdy bajt sterujący, który mógłby wystąpić w polu danych, odpowiednią sekwencją zastępczą. Z kolei odbiornik zamienia sekwencje zastępcze właściwymi bajtami danych. W rezultacie w ramce przenoszone są informacje o dowolnej treści bez ryzyka, że system transmisyjny pomyli występujące w nich znaki z informacjami sterującymi. Technika ta jest nazywana **nadziewaniem bajtami**, choć niekiedy stosuje się również określenie **nadziewania znakami**. Analogiczne rozwiązanie w systemach strumieniowania bitowego nosi nazwę **nadziewania bitami**.

Przeanalizujmy działanie mechanizmu nadziewania bajtami na przykładzie ramki przedstawionej na rysunku 13.7. Wykorzystanie znaków SOH i EOT do wyznaczania granic ramki oznacza, że znaki te nie mogą wystąpić w polu danych. Rozwiążanie polega więc na zarezerwowaniu trzeciego znaku, za pomocą którego oznaczone zostaną wszystkie niedozwolone wartości pola danych. Założymy, że wspomnianym trzecim znakiem jest znak ESC o kodzie szesnastkowym 0x1B. Jeśli w polu danych wystąpi którykolwiek z trzech zarezerwowanych znaków, nadajnik zastąpi go sekwencją dwóch znaków. Przykładowe odwzorowanie wartości zostało przedstawione w tabeli 13.5.

**Tabela 13.5.** Przykład nadziewania bajtami, w którym znaki specjalne są zastępowane dwuznakowymi sekwencjami bajtowymi

Bajt w polu danych	Wysyłana sekwencja
SOH	ESC A
EOT	ESC B
ESC	ESC C

Z zestawienia wynika, że nadawca zastępuje każdy znak SOH dwoma znakami: ESC i A. Każde wystąpienie wartość EOT jest reprezentowane przez znaki ESC i B, a każdy znak ESC jest przesyłany jako zbitka znaków ESC i C. Stacja odbiorcza wykonuje operację odwrotną. Jeśli natopka znak ESC, po którym następuje znak A, B lub C, wstawia pojedynczy znak odpowiadający danej kombinacji. Na rysunku 13.8 przedstawiono przykładowe pole danych przed wykonaniem operacji nadziewania bajtami i po jej realizacji. Warto zwrócić uwagę na to, że ramka wynikowa nie zawiera wartości SOH i EOT w polu danych.



**Rysunek 13.8.** Nadziewanie bajtami — pierwotne dane (a) i wynik operacji (b)

## 13.14. Podsumowanie

Sieci transmisji danych dzielą się na sieci z przełączaniem obwodów i sieci z przełączaniem pakietów. Przełączanie pakietów, stanowiące podstawę działania internetu, jest formą statystycznego multipleksowania, w którym nadawca dzieli wiadomość na mniejsze pakiety. Technologie sieci pakietowych są klasyfikowane jako sieci lokalne (LAN), sieci rozległe (WAN) oraz sieci metropolitarne (MAN). Rozwiązania LAN i WAN należą do najczęściej stosowanych w praktyce.

Opracowywaniem standardów transmisji danych zajmuje się organizacja o nazwie IEEE. W odniesieniu do sieci LAN jej zadanie koncentruje się głównie na definiowaniu dwóch najwyższych warstw stosu protokołów.

Sieci LAN są projektowane zgodnie z czterema topologiami: magistrali, gwiazdy, pierścienia i siatkową. Topologie siatkowe rzadko znajdują praktyczne zastosowanie z uwagi na wysoki koszt ich implementacji.

Każdy pakiet przesyłany za pośrednictwem sieci LAN zawiera adres MAC, który identyfikuje odbiorcę danych. Zgodnie ze standardem opracowanym przez organizację IEEE, adres MAC składa się z 48 bitów podzielonych na dwa pola. Pierwsze z nich opisuje dostawcę urządzeń, a drugie jest niepowtarzalną wartością identyfikującą określony komponent. Adres uwzględnia również informację o rodzaju transmisji. Informuje o emisji pojedynczej (czyli przesyłaniu danych do wybranego komputera), rozgłoszeniu (czyli wysyłaniu informacji do wszystkich komputerów w danej sieci LAN) lub multiemisji (dostarczaniu danych do grupy komputerów w sieci).

Termin **ramka** jest wykorzystywany do definiowania formatu pakietu właściwego dla danej sieci. Ramka składa się z dwóch elementów — nagłówka zawierającego metainformacje oraz pola danych obejmującego zasadniczą treść komunikatu. W sieciach przekazujących informacje w formie znaków ramka może być wyznaczana przez specjalne znaki, które wskazują jej początek i koniec.

Technika nadziewania bitami (lub bajtami) umożliwia zarezerwowanie pewnych bajtów (lub sekwencji bitowych) do wyznaczania początku i końca ramki. W celu zagwarantowania, że w polu danych nie wystąpią zarezerwowane wartości, nadawca musi zastąpić je specjalnymi kombinacjami bajtowymi, które zostaną usunięte i zastąpione pierwotnymi wartościami po stronie odbiorczej.

## ZADANIA

- 13.1. Na czym polega przełączanie obwodów i jakie są najważniejsze cechy przełączania obwodów?
- 13.2. Czy w pojedynczym włóknie światłowodowym sieci z przełączaniem obwodów może być ustanowionych wiele obwodów? Uzasadnij odpowiedź.
- 13.3. W jaki sposób przesyłane są pliki o dużych rozmiarach w sieciach pakietowych?
- 13.4. Który rodzaj systemu (z przełączaniem obwodów czy z przełączaniem pakietów) będzie odpowiedni do przekazywania prezentacji audiowizualnej? Dlaczego?
- 13.5. Wymień cechy charakterystyczne sieci LAN, MAN i WAN.
- 13.6. Wymień nazwy dwóch podwarstw drugiej warstwy stosu protokołów zdefiniowane przez organizację IEEE. Opisz przeznaczenie każdej z nich.
- 13.7. Czym charakteryzuje się połączenie punkt-punkt?
- 13.8. Wymień cztery podstawowe topologie sieci LAN.
- 13.9. Czy okablowanie sieci pierścieniowej może być ułożone w linii prostej (na przykład wzdłuż korytarza)? Uzasadnij odpowiedź.
- 13.10. Ile połączeń jest potrzebnych do zrealizowania sieci siatkowej składającej się z 20 komputerów?
- 13.11. Skąd wiadomo, że dany adres MAC jest adresem emisji pojedynczej?
- 13.12. Opisz adresy emisji pojedynczej, multiemisji i rozgłoszeniowe. Jakie jest ich przeznaczenie?
- 13.13. Jak przebiega proces podejmowania decyzji o zaakceptowaniu pakietu w komputerze przyłączonym do sieci LAN?
- 13.14. Jaki termin jest wykorzystywany do opisu metadanych towarzyszących pakietowi?
- 13.15. Podaj definicję pojęcia **ramka**.

- 13.16.** Do czego służy nadziewanie bajtami?
- 13.17.** Napisz dwa programy komputerowe. Pierwszy z nich powinien pobierać plik wejściowy i generować dane wypełnione dodatkowymi bajtami zgodnie ze schematem przedstawionym w tabeli 13.5. Drugi powinien usuwać nadmiarowe bajty. Zademonstruj współdziałanie programów z programami przygotowanymi przez inne osoby.

# *Zawartość rozdziału*

14.1. Wprowadzenie	261
14.2. Podział mechanizmów regulujących dostęp do medium	261
14.3. Statyczna i dynamiczna alokacja kanałów	262
14.4. Protokoły alokacji kanałów	263
14.5. Protokoły sterowania dostępem	264
14.6. Protokoły dostępu swobodnego	266
14.7. Podsumowanie	272

# 14

## Podwarstwa MAC

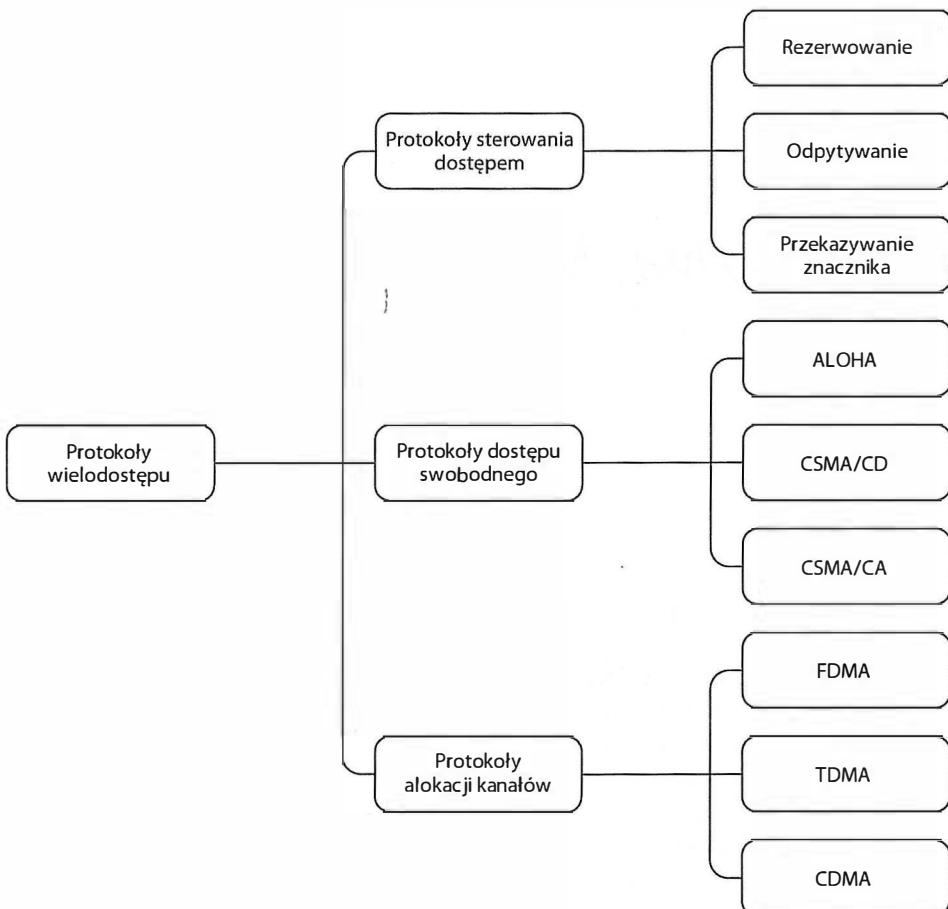
### 14.1. Wprowadzenie

W tej części książki przedstawione zostały zagadnienia związane z wymianą danych w sieciach pakietowych. Tematem poprzedniego rozdziału była idea przełączania pakietów oraz budowa dwóch podstawowych rodzajów sieci pakietowych — sieci WAN i LAN. Zaprezentowano w nim również model standardów IEEE i wyjaśniono przyczynę podziału warstwy 2. stosu protokołów na dwie podwarstwy.

Ten rozdział stanowi kontynuację wcześniejszych rozważań i prezentuje szczegółowe informacje na temat podwarstwy MAC. Wyjaśnione zostały w nim również protokoły systemów wielodostępnych oraz zagadnienia związane ze statyczną i dynamiczną alokacją kanału. W kolejnych rozdziałach opisane są technologie sieciowe, które bazują na prezentowanych tutaj mechanizmach.

### 14.2. Podział mechanizmów regulujących dostęp do medium

W jaki sposób można skoordynować działania wielu niezależnych komputerów, które stają się o dostęp do wspólnego medium transmisyjnego? Istnieją trzy rozwiązania: zastosowanie zmodyfikowanych technik multipleksacji, zaimplementowanie rozproszonego algorytmu sterowania dostępem do medium oraz wdrożenie strategii losowego zajmowania medium. Na rysunku 14.1 przedstawiono podział różnych praktycznych rozwiązań z tych zakresów.



Rysunek 14.1. Podział protokołów sterujących dostępem do wspólnego medium

### 14.3. Statyczna i dynamiczna alokacja kanałów

Pojęcie **alokacja kanałów** służy do opisu procesu kojarzenia określonej wymiany danych z kanałem rezerwowanym przez niskopoziomowy system transmisyjny. Mechanizm jest zbliżony w działaniu do multipleksacji opisanej w rozdziale 11. Jako przykład można przeanalizować technikę zwielokrotniania częstotliwościowego (FDM). Większość systemów FDM przypisuje komunikującym się jednostkom pewną określoną częstotliwość nośną. Wydzielany jest w ten sposób jeden kanał. Jeśli powiązanie między urządzeniami końcowymi i częstotliwością nośną nie zmienia się w czasie, system alokacji kanałów określa się jako **statyczny** (o powiązaniu **jeden-do-jednego**).

Statyczna alokacja kanałów sprawdza się doskonale w sytuacjach, w których liczba komunikujących się ze sobą jednostek jest wstępnie ustalona, a wymiana danych zawsze zachodzi między tymi samymi jednostkami. Jednak w wielu praktycznych konfiguracjach sieciowych liczba stacji zmienia się nieustannie. Przykładem może być sieć telefonii komór-

kowej dużego miasta. Użytkownicy przemieszczają się. Mogą włączać i wyłączać telefony w dowolnym momencie. Zbiór aparatów telefonicznych pozostających w zasięgu stacji bazowych zmienia się w czasie. Niezbędne jest więc zastosowanie mechanizmu **dynamicznej alokacji kanałów**. Gwarantuje on utworzenie nowego powiązania jednostki (na przykład telefonu komórkowego) z kanałem w chwili pojawienia się tej jednostki oraz usunięcie powiązania, gdy dana jednostka przestanie korzystać z sieci.

Podsumowując:

*Statyczna alokacja kanałów jest użyteczna wtedy, gdy liczba komunikujących się urządzeń jest wstępnie znana i nie zmienia się w czasie. Większość sieci wymaga jednak implementacji mechanizmu dynamicznego przydzielu kanałów.*

## 14.4. Protokoły alokacji kanałów

Protokoły alokacji kanałów stanowią rozszerzenie technik multipleksowania strumieni danych, które zostały opisane w rozdziale 11. Lista najważniejszych rozwiązań z tej grupy widnieje w tabeli 14.1.

Tabela 14.1. Trzy główne rodzaje protokołów alokacji kanałów

Protokół	Opis
FDMA	Wielodostęp z podziałem częstotliwości (ang. <i>Frequency Division Multiple Access</i> )
TDMA	Wielodostęp z podziałem czasu (ang. <i>Time Division Multiple Access</i> )
CDMA	Wielodostęp kodowy (ang. <i>Code Division Multiple Access</i> )

### 14.4.1. FDMA

Z powyższego rysunku wynika, że techniki alokacji kanałów są powiązane ze zwielokrotnianiem częstotliwościowym, czasowym i kodowym. Na przykład technika FDMA uzupełnia multipleksację z podziałem częstotliwości. Rozszerzenie to polega na udostępnieniu mechanizmów, które umożliwiają niezależnym stacjom wybranie takich częstotliwości, które nie będą zakłócały wymiany danych realizowanej przez inne stacje na innych częstotliwościach. W jaki sposób system FDMA przydziela nośne? W niektórych rozwiązaniach wykorzystywany jest do tego celu centralny sterownik zajmujący się dynamiczną alokacją częstotliwości. Każda nowo przyłączająca się stacja wykorzystuje zarezerwowany kanał sterujący do nawiązania połączenia ze sterownikiem. Następnie przesyła do niego żądanie przydzielu częstotliwości. Sterownik odszukuje wolną nośną i stację o jej dostępności. Po wstępnej wymianie informacji stacja korzysta z przydzielonej jej nośnej (tj. przydzielonego kanału) w dalszej komunikacji.

### 14.4.2. TDMA

Rozszerzenie zwielokrotniania czasowego, opisywane jako TDMA, jest rozwiązaniem analogicznym do mechanizmu alokacji częstotliwości. W najprostszym przypadku każda jednostka uczestnicząca w wymianie danych otrzymuje liczbę z przedziału od 1 do N. Poszczególne stacje wysyłają swoje dane w kolejności 1, 2, 3, ... N. Podobnie jak w przypadku FDMA system TDMA zapewnia dynamiczną alokację kanału przez przypisanie odpowiedniej szczeliny czasowej urządzeniu włączającemu się do sieci.

{}

### 14.4.3. CDMA

Systemy zwielokrotnienia kodowego umożliwiają jednoczesne przesyłanie danych przez wiele stacji. Jest to możliwe dzięki zastosowaniu specjalnych matematycznych technik kodowania kanałowego (zagadnienie to zostało opisane w rozdziale 11.). Mechanizm CDMA stanowi implementację techniki zwielokrotniania kodowego.

## 14.5. Protokoły sterowania dostępem

Protokoły sterowania dostępem do medium są w pewnym sensie rozproszonymi wersjami multipleksacji statystycznej. Trzy najważniejsze rozwiązania z tej grupy przedstawiono w tabeli 14.2.

Tabela 14.2. Trzy najważniejsze rodzaje protokołów sterowania dostępem do medium

Rodzaj	Opis
Odpytywanie	Centralny sterownik okresowo odpytuje stacje o chęć przesłania danych i umożliwia wprowadzenie do sieci jednego pakietu.
Rezerwacja	Stacje dostarczają żądania udostępnienia medium w kolejnej turze transmisji danych.
Przekazywanie znacznika	Stacje przekazują między sobą znacznik. Za każdym razem, gdy stacja otrzyma znacznik, może przesyłać jeden pakiet.

### 14.5.1. Odpytywanie

W sieciach z zaimplementowanym mechanizmem **odpytywania** wykorzystywany jest centralny sterownik, którego zadanie polega na okresowym odwoływaniu się do kolejnych stacji i umożliwianiu im przesłania pakietu. Funkcjonowanie sterownika jest zgodne z algorytmem 14.1. Najważniejszym elementem wspomnianego algorytmu jest wybór jednostki, która jako następna uzyska dostęp do medium. Powszechnie stosuje się dwa mechanizmy doboru stacji:

- karuzelowy (ang. *round robin*),
- priorytetowy.

**Algorytm 14.1.** Sterowanie dostępem do medium zgodnie z mechanizmem odpytywania**Cel:**

Sterowanie przesyłaniem pakietów zgodnie z mechanizmem odpytywania

**Rozwiązanie:**

```
Sterownik w nieskończoność ponawia operację {  
    Wybranie stacji S i przesłanie do niej zapytania o chęć  
    skorzystania z medium.  
    Oczekiwanie na odpowiedź ze stacji S (na przesłanie  
    pakietu lub informacji o nieaktywności).  
}
```

Zastosowanie algorytmu karuzelowego oznacza, że każda stacja ma jednakową szansę na przesłanie pakietu. Z kolei w rozwiążaniu priorytetowym wybrane jednostki mają większe szanse na dostęp do medium niż inne. Technika ta pozwala na przykład na przypisanie większego priorytetu telefonowi IP niż komputerowi osobistemu.

**14.5.2. Rezerwacja**

System **rezerwacji** często znajduje zastosowanie w łączności satelitarnej. Podstawą jego działania jest podział transmisji na dwa etapy, dzięki czemu każda emisja pakietu jest zaplanowana w czasie poprzedniej rundy transmisji. Zazwyczaj dostęp do medium nadzoruje centralny sterownik, działający zgodnie z algorytmem 14.2.

**Algorytm 14.2.** Sterowanie dostępem do medium zgodnie z mechanizmem rezerwacji**Cel:**

Sterowanie przesyłaniem pakietów zgodnie z mechanizmem rezerwacji

**Rozwiązanie:**

```
Sterownik w nieskończoność ponawia operację {  
    Przygotowanie listy stacji gotowych do przesłania  
    pakietu.  
    Umożliwienie transmisji stacjom z listy.  
}
```

W pierwszym kroku każda stacja zgłasza, czy ma pakiet do przesłania w następnej rundzie transmisji, czy nie. Sterownik emituje natomiast listę stacji, które otrzymały pozwolenie na nadawanie. W drugiej fazie stacje wymienione na liście przystępują do transmisji danych. Działanie sterownika jest rozwiązywane na kilka sposobów. Zazwyczaj do zberania informacji o rezerwacjach przeznaczony jest oddzielnny kanał, aby w tym samym czasie mogła być realizowana wymiana danych w kanale podstawowym.

### 14.5.3. Przekazywanie znacznika

Mechanizm **przekazywania znacznika** jest wykorzystywany w wielu rozwiązaniach praktycznych, związanych głównie z topologiami pierścieniowymi<sup>37</sup>. W zrozumieniu zasady działania algorytmu pomocne może być wyobrażenie sobie kilku połączonych w okrąg komputerów, z których tylko jeden przetwarza w danej chwili specjalny komunikat nazywany **znacznikiem**. Sterowanie dostępem do medium jest regulowane przez algorytm 14.3.

**Algorytm 14.3.** Sterowanie dostępem do medium zgodnie z mechanizmem przekazywania znacznika

Cel:

Sterowanie przesyłaniem pakietów zgodnie z mechanizmem przekazywania znacznika

Rozwiązanie:

Każdy komputer w sieci ponawia następujące czynności {  
 Oczekивание на доставление сообщения.  
 Надание пакета, если кто-либо из других ожидает передачи.  
 Прекращение передачи сообщения в следующую станцию.  
 }

Jeśli żadna ze stacji nie ma nic do nadania, znacznik krąży nieustannie pomiędzy kolejnymi jednostkami sieci. Kierunek przekazywania znacznika jest wyznaczany przez urządzenia zarządzające pracą pierścienia. Jeśli obieg znacznika jest zgodny z ruchem wskazówek zegara, wzmiarkowana w algorytmie **następna stacja** jest kolejną stacją przyłączoną do sieci wyznaczaną zgodnie z ruchem wskazówek zegara. W przypadku zastosowania mechanizmu przekazywania znacznika w innych topologiach (na przykład w topologii magistrali) każdej jednostce sieciowej odpowiada pewna pozycja w logicznej sekwencji stacji. Przekazywanie znacznika odbywa się zgodnie z porządkiem tej sekwencji.

## 14.6. Protokoły dostępu swobodnego

W wielu sieciach (szczególnie w sieciach LAN) nie obowiązują żadne mechanizmy sterowania dostępem do medium transmisyjnego. Przyłączone do sieci komputery rywalizują o dostęp bez żadnego nadzoru ze strony urządzeń zewnętrznych. W przypadku dostępu **swobodnego** stacja korzysta z medium jedynie wtedy, gdy ma pakiet do przesłania. Ponadto, w celu wyeliminowania problemu jednoczesnego korzystania z medium przez większą liczbę komputerów w tym samym czasie stosuje się technikę losowego wstrzymywania nadawania. Losowość poszczególnych algorytmów została szczegółowo opisana w kolejnych punktach podrozdziału. Z kolei w tabeli 14.3 zostały wymienione trzy spośród protokołów swobodnego dostępu.

---

<sup>37</sup> Przekazywanie znaczników było niegdyś podstawowym sposobem działania pierścieniowych sieci LAN. Jednak z czasem popularność tego rozwiązania znacznie zmalała. Obecnie przekazywanie znaczników rzadko jest stosowane w sieciach lokalnych.

Tabela 14.3. Trzy protokoły dostępu swobodnego

Rodzaj	Opis
ALOHA	Pierwszy protokół tego typu, który został wykorzystany w sieci radiowej na Hawajach. Często opisywany w podręcznikach akademickich z uwagi na łatwość analizy, ale niewykorzystywany praktycznie.
CSMA/CD	Wielodostęp z wykrywaniem nośnej i detekcją kolizji (ang. <i>Carrier Sense Multiple Access with Collision Detection</i> ). Podstawa funkcjonowania Ethernetu i najczęściej implementowany protokół dostępu swobodnego.
CSMA/CA	Wielodostęp z wykrywaniem nośnej i unikaniem kolizji (ang. <i>Carrier Sense Multiple Access with Collision Avoidance</i> ). Podstawa działania sieci bezprzewodowych (Wi-Fi).

#### 14.6.1. Mechanizm ALOHA

Dostęp swobodny po raz pierwszy zastosowano w sieci ALOHAnet zbudowanej na Hawajach. Choć nie jest ona obecnie wykorzystywana, wdrożone w niej rozwiązania stały się podstawą działania innych podobnych systemów. Sama sieć składała się z pojedynczego nadajnika o wysokiej mocy, umieszczonego centralnie względem stacji połączonych z komputerami. Każda ze stacji zawierała nadajnik o mocy wystarczającej do komunikacji z jednostką centralną (ale zbyt małej do wymiany informacji z innymi stacjami). Sieć ALOHAnet pracowała na dwóch częstotliwościach — 413,475 MHz, przeznaczonej dla ruchu **wychodzącego** z centralnego nadajnika do jednostek zależnych, oraz 407,305 MHz, obsługującej ruch **przychodzący** do stacji centralnej z jednostek zdalnych. Budowę systemu ALOHAnet pokazano na rysunku 14.2.



Rysunek 14.2. Częstotliwości ruchu wychodzącego i przychodzącego w sieci ALOHAnet

Działanie protokołu ALOHA nie było szczególnie skomplikowane. Gdy którakolwiek ze stacji miała pakiet do nadania, rozpoczęła emisję na częstotliwości ruchu przychodzącego. Centralny nadajnik powielał otrzymane informacje na częstotliwości ruchu wychodzącego (przekazywanego do wszystkich pozostałych stacji). Aby zweryfikować poprawność dostarczanych danych, stacja nadawcza odbierała jednocześnie sygnał w kanale ruchu wychodzącego. Gdy odebrany pakiet dokładnie odpowiadał wysyłanemu, komputer rozpoczął nadawanie kolejnej porcji danych. W przypadku braku transmisji zwrotnej stacja nadawcza zaprzestała emisji na pewien czas, po czym powtarzała nadawanie.

W jakich sytuacjach pakiet nie docierał w poprawnej formie? Gdy występowały interferencje. Równoczesne rozpoczęcie nadawania przez dwie stacje powodowało wzajemne zakłócanie się sygnałów. Do opisu takiego zdarzenia wykorzystano termin **kolizja**. W protokole komunikacyjnym uwzględniono mechanizm obsługi kolizji, którego działanie sprowadzało się do wymuszenia **retransmisji** pakietu. Rozwiążanie to jest powszechnie stosowane w wielu dzisiejszych sieciach.

Czas przerwy w nadawaniu (przed retransmisją) musiał być odpowiednio dobrany. W przeciwnym przypadku dwie stacje mogłyby rozpocząć emisję w tym samym momencie. To z kolei doprowadziłoby do ponownego wystąpienia kolizji. Aby rozwiązać problem, wprowadzono element losowości (każda stacja w sposób losowy dobierała czas przerwy). W ten sposób prawdopodobieństwo interferencji sygnałów zostało istotnie zmniejszone. Przeprowadzone analizy dowiodły, że sieć ALOHAnet była bardzo obciążona i występowało w niej wiele kolizji. Nawet po zastosowaniu mechanizmu losowego dobierania przerw procentowa wartość udanych transmisji danych wynosiła około 18% pojemności kanału (oznacza to wykorzystanie kanału na poziomie 18%).

#### 14.6.2. Mechanizm CSMA/CD

W 1973 roku inżynierowie z firmy Xerox PARC opracowali niezwykle użyteczną technologię sieciową bazującą na protokole dostępu swobodnego. W 1978 roku firmy Digital Equipment Corporation, Intel i Xerox przygotowały rozwiązanie (nieformalnie nazywane **standardem DIX**), które stało się podstawą działania **Ethernetu**. Pierwotnie technologia ethernetowa zakładała wykorzystanie pojedynczego kabla, do którego przyłączane były komputery<sup>38</sup>. Kabel ten pełnił funkcję współdzielonego medium transmisyjnego (zastępując przestrzeń, w której były przesyłane fale elektromagnetyczne). Przekazywanie danych polegało na transmisji sygnału w ramach wspólnego kabla. Niepotrzebne okazały się dwie częstotliwości nośne oraz centralny nadajnik. Mimo pewnych różnic między rozwiązaniami, systemy Ethernet i ALOHAnet łączyły jeden problem. Jak rozwiązać problem kolizji, które występują w chwili, gdy dwie jednostki jednocześnie rozpoczęły nadawanie sygnałów?

W standardzie Ethernet zaproponowano kilka innowacyjnych sposobów obsługi kolizji:

- wykrywanie nośnej,
- detekcję kolizji,
- binarne wykładnicze wyznaczanie czasu wstrzymania transmisji.

**Wykrywanie nośnej.** Zamiast umożliwić stacjom nadawanie danych za każdym razem, gdy pakiet zostanie przygotowany do wysłania, przyjęto założenie, że każda jednostka będzie monitorowała stan wspólnego medium i sprawdzała, czy w danym czasie nie jest realizowana inna transmisja danych. Mechanizm znany jako **wykrywanie nośnej** eliminuje najbardziej oczywiste problemy i znacznie zwiększa poziom wykorzystania sieci.

**Detekcja kolizji.** Mimo zastosowania techniki wykrywania nośnej problem kolizji nie został całkowicie usunięty. Może ona powstać w sytuacji, w której dwie jednostki oczekują

<sup>38</sup> W kolejnym rozdziale zostały opisane nowoczesne warianty okablowania ethernetowego.

na zakończenie transmisji realizowanej między innymi komputerami. W chwili gdy obydwie ustalały, że łącze zostało zwolnione, jednocześnie rozpoczęła nadawanie własnych informacji. Częściowo przyczyną takiego stanu jest skończona wartość czasu potrzebnego na przeniesienie impulsów elektrycznych w kablu. Stacja przyłączona do jednego końca kabla nie może wykryć nośnej w przewodzie, jeśli w tym samym czasie nadawanie rozpoczęcie jednostka znajdująca się na drugim końcu magistrali.

Dekodacja kolizji polega na monitorowaniu stanu medium transmisyjnego również w czasie wysyłania informacji. Jeśli odbierany z przewodu sygnał odbiega od emitowanego, wiadomo, że powstała kolizja. Taka technika postępowania jest nazywana **detekcją kolizji**. Wykrycie kolizji powoduje wstrzymanie emisji sygnału.

Transmisję ethernetową komplikują wiele czynników. Na przykład po wykryciu kolizji konieczne jest podtrzymanie nadawania sygnału przez czas wystarczający do tego, żeby wszystkie stacje zarejestrowały tę kolizję. Ponadto, po zakończeniu nadawania łącze musi pozostać w stanie nieaktywnym przez czas określany jako **przerwa międzyramkowa** (wynoszący 9,6 µs w Etherencie o przepustowości 10 Mb/s), aby zagwarantować wszystkim stacjom możliwość nadawania po wykryciu nieaktywności sieci. Uwzględnienie takich parametrów w specyfikacji daje obraz tego, z jaką starannością przygotowywano założenia technologii.

**Binarne wykładnicze wyznaczanie czasu wstrzymania transmisji** (ang. *Binary Exponential Backoff*). Mechanizmy ethernetowe nie tylko wykrywają kolizje, ale również przywracają poprawne funkcjonowanie sieci. Po ustaleniu, że wystąpiła kolizja, komputer, zanim przystąpi do nadawania, musi się upewnić, że medium jest wolne. Podobnie jak w systemie ALOHAnet, w sieciach Ethernet obowiązuje losowanie wartości czasu nieaktywności, które zapewnia, że dwie jednostki nie rozpoczęły nadawania informacji w tej samej chwili. W standardzie zapisano maksymalną wartość przerwy ( $d$ ) i narzucono obowiązek losowania czasu z przedziału od zera do  $d$ . W większości przypadków po wystąpieniu kolizji jedna ze stacji wstrzymuje swoje działanie na czas krótszy niż druga jednostka i szybciej przystępuje do ponownego wysłania pakietu, dzięki czemu przywracana jest normalna praca sieci.

W sytuacji, w której dwa komputery (lub większa ich liczba) wylosują zbliżone wartości czasu nieaktywności, wznowienie transmisji nastąpi niemal w tym samym momencie. W rezultacie powstanie następna kolizja. Aby uniknąć serii kolizji, każdy komputer przyłączony do sieci Ethernet musi podwoić zakres wartości, z którego losuje czas przerwy, jeśli operacja ta jest wykonywana po raz kolejny. Po pierwszej kolizji wybierana jest wartość z przedziału od 0 do  $d$ , po drugiej z przedziału od 0 do  $2d$ , po trzeciej z przedziału od 0 do  $4d$  itd. Po kilku kolizjach zakres dopuszczalnych wartości jest dostatecznie duży, żeby mieć pewność, że tylko jeden z komputerów wylosuje najkrótszy czas przerwy i rozpoczęcie nadawanie bez dalszych przeszkód.

Podwajanie zakresu po każdej kolizji (od którego pochodzi nazwa mechanizmu) oznacza, że sieć może szybko powrócić do pełnej sprawności, ponieważ komputery przez dłuższy czas oczekują na zwolnienie wspólnego medium. Nawet w mało prawdopodobnym przypadku wylosowania niemal identycznych wartości wykładniczy mechanizm wyznaczania czasu nieaktywności gwarantuje, że rywalizacja o dostęp zakończy się po kilku kolizjach.

Połączenie wszystkich opisanych technik jest znane jako algorytm **wielodostępu z wykrywaniem nośnej i detekcją kolizji** (CSMA/CD). Działanie mechanizmu zostało opisane w algorytmie 14.4.

#### Algorytm 14.4. Transmisja pakietu zgodnie z algorytmem CSMA/CD

Cel:

Przesłanie pakietu zgodnie z algorytmem CSMA/CD

Realizacja:

Oczekiwanie na gotowość pakietu do wysłania.

Oczekiwanie na zwolnienie medium transmisyjnego  
(wykrywanie nośnej).

Wstrzymanie działania na czas przerwy międzyramkowej.

Przypisanie zmiennej  $x$  standardowego zakresu losowania ( $d$ ).

Próba przesłania pakietu (detekcja kolizji).

while (we wcześniejszej transmisji wystąpiła kolizja) {

    Wylosowanie wartości opóźnienia  $q$  z przedziału od 0 do  $x$ .

    Wstrzymanie działania na czas  $q$  mikrosekund.

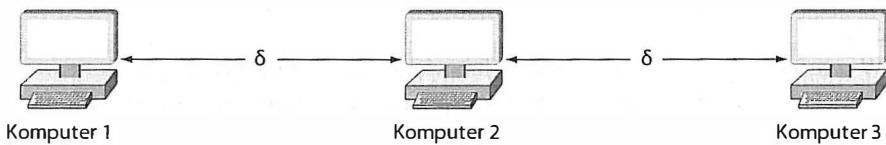
    Podwojenie wartości  $x$  na wypadek kolejnej iteracji.

    Próba retransmitowania pakietu (detekcja kolizji).

}

#### 14.6.3. Mechanizm CSMA/CA

Mimo że algorytm CSMA/CD doskonale sprawdza się w sieciach przewodowych, nie jest stosowany w rozwiązańach bezprzewodowych. Przyczyną jest ograniczony zasięg ( $\delta$ ) nadajnika karty. Jeśli odległość odbiornika od nadajnika jest większa niż  $\delta$ , odbiornik nie będzie odbierał sygnału, a tym samym nie będzie mógł wykryć nośnej. Problem ten można zaobserwować w sieci złożonej z trzech jednostek o konfiguracji odpowiadającej rysunkowi 14.3.

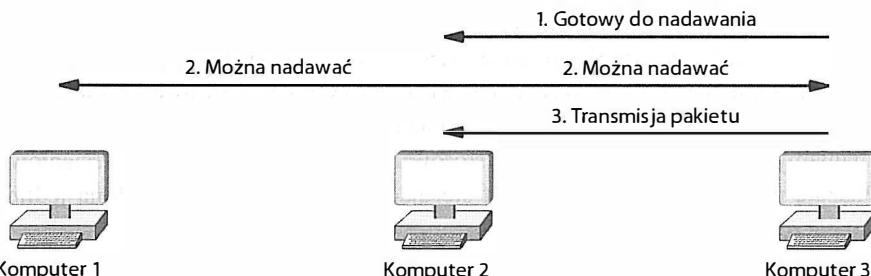


Rysunek 14.3. Trzy komputery wyposażone w bezprzewodowe karty LAN umiejscowione w maksymalnej odległości od siebie

Komputer 1 może się komunikować z komputerem 2, ale nie odbiera żadnych danych od komputera 3. Zatem gdy komputer 3 przesyła informacje do komputera 2, mechanizmy wykrywania nośnej w komputerze 1 nie rejestrują żadnej emisji. Analogicznie, w przypadku jednoczesnego wysyłania sygnałów przez komputery 1 i 3 tylko komputer 2 wykryje kolizję. Opisany problem jest często nazywany **problemem ukrytej stacji**. Część jednostek nie jest bezpośrednio „widoczna” dla innych urządzeń.

W celu zagwarantowania poprawnej pracy wszystkich stacji współdzielących medium transmisyjne w bezprzewodowych sieciach LAN wprowadzono zmodyfikowaną wersję wcześniejszego protokołu. Nowy mechanizm jest znany pod nazwą **wielodostępu z wykry-**

**waniem nośnej i unikaniem kolizji** (CSMA/CA — ang. *Carrier Sense Multiple Access with Collision Avoidance*). Zamiast uzależniać działanie systemu od odebrania informacji przez wszystkie komputery, mechanizm CSMA/CA (stosowany w bezprzewodowych sieciach LAN) przed dostarczeniem pakietu do stacji odbiorczej wymusza w niej chwilową emisję sygnału. Jeśli zarówno nadawca, jak i odbiorca wyemitują krótką wiadomość, wszystkie komputery w ich zasięgu będą wiedziały, że rozpoczęyna się nadawanie pakietu. Przebieg tej procedury przedstawiono na rysunku 14.4.



Rysunek 14.4. Wymiana komunikatów w czasie transmisji pakietu z komputera 3 do komputera 2

Zgodnie z przedstawionym schematem działania komputer 3 przesyła do jednostki 2 krótką informację o tym, że jest gotowy do przesłania pakietu. W odpowiedzi komputer 2 wysyła wiadomość potwierdzającą możliwość dostarczenia pakietu. Wszystkie komputery będące w zasięgu stacji 3 odbierają wstępne ogłoszenie, a wszystkie jednostki w pobliżu komputera 2 otrzymują odpowiedź. W rezultacie, mimo że komputer 1 nie może odebrać zasadniczego sygnału lub wykryć nośnej, zostaje poinformowany o realizowanej transmisji.

Stosowanie algorytmu CSMA/CA nie eliminuje kolizji komunikatów sterujących, ale pozwala na łatwe obsługiwanie tego typu problemów. Na przykład jeśli komputery 1 i 3 spróbują wysłać w tym samym czasie pakiet do komputera 2, wyemitowane przez nie komunikaty sterujące spowodują kolizję. Komputer 2 wykryje taką kolizję i nie odeśle odpowiedzi na pierwotne zgłoszenie. Po wystąpieniu kolizji, a przed ponowieniem komunikatu sterującego stacje nadawcze uruchomią mechanizm losowego wstrzymywania transmisji. Ponieważ informacje sterujące mają znacznie mniejszy rozmiar od samego pakietu, prawdopodobieństwo wystąpienia drugiej kolizji jest znikome. Ostatecznie więc jeden z komunikatów dotrze do komputera 2 w poprawnej formie i komputer 2 będzie mógł odesłać właściwą odpowiedź.

Podsumowując:

*Komputery w bezprzewodowych sieciach LAN bywają oddalone od siebie na odległość większą, niż może pokonać sygnał radiowy. Z tego powodu w sieciach tych stosuje się mechanizm CSMA/CA, który wymusza na nadawcy i odbiorcy wysłanie krótkiego komunikatu sterującego przed rozpoczęciem transmisji zasadniczego pakietu.*

## 14.7. Podsumowanie

Zdefiniowana przez organizację IEEE podwarstwa MAC operuje protokołami, które zarządzają dostępem do współdzielonego medium transmisyjnego. Techniki alokacji kanałów komunikacyjnych są uzupełnieniami dla mechanizmów zwielokrotniania czasowego, częstotliwościowego oraz kodowego i są nazywane protokołami wielodostępu czasowego, częstotliwościowego i kodowego. Sama alokacja ma dwie formy: statyczną i dynamiczną.

Protokoły sterowania dostępem do medium umożliwiają poszczególnym stacjom uczestniczenie w statystycznej multipleksacji. Mechanizmy odpytywania bazują na centralnym sterowniku, który okresowo sprawdza, czy stacje sieciowe mają jakiekolwiek pakiety do przesłania. Systemy rezerwacji (wykorzystywane w transmisjach satelitarnych) wymagają od stacji zgłoszenia gotowości do nadawania w kolejnej turze transmisji. Z kolei mechanizmy przekazywania znacznika (stosowane zazwyczaj w sieciach pierścieniowych) przekazują między stacjami komunikat sterujący, który uprawnia określona jednostkę do rozpoczęcia transmisji danych.

Protokoły dostępu swobodnego działają w oparciu o założenie, że stacje powinny rywalizować o dostęp do medium. W pierwszym rozwiążaniu tego typu (w protokole ALOHA) wykorzystano dwie częstotliwości nośne, po jednej dla ruchu przychodzącego i wychodzącego. Jeśli stacja nadawcza nie odebrała własnego pakietu, rozpoczynała retransmisję. Dostęp do wspólnego medium w sieciach Ethernet reguluje protokół CSMA/CD. Jego zadanie polega na uniemożliwieniu nadawania, jeśli w danej chwili realizowana jest inna transmisja, a także na przywracaniu poprawnego funkcjonowania sieci po wystąpieniu kolizji (do czego służy mechanizm binarnego, wykładniczego wyznaczania czasu przerwy).

Ponieważ w bezprzewodowych sieciach LAN część stacji pozostaje ukryta przed innymi jednostkami sieciowymi, obowiązek zarządzania ruchem spoczywa na protokole CSMA/CA. Zgodnie z jego założeniami, przed wyemitowaniem pakietu obydwa komputery uczestniczące w komunikacji muszą wysłać krótką wiadomość sterującą. Służy ona do poinformowania wszystkich pozostałych jednostek sieciowych o rozpoczęjącej się transmisji.

## ZADANIA

- 14.1. Omów trzy podstawowe sposoby zarządzania dostępem do wspólnego medium.
- 14.2. Podaj przykład sieci, w której stosowana jest dynamiczna alokacja kanałów.
- 14.3. Wymień i opisz trzy podstawowe techniki alokacji kanałów.
- 14.4. Omów algorytm odpytywania oraz dwie ogólne polityki odpytywania.
- 14.5. W jaki sposób sterownik systemu rezerwacyjnego buduje listę stacji, które rozpoczną nadawanie w kolejnej turze?
- 14.6. Czym jest znacznik? W jaki sposób znaczniki zarządzają dostępem do medium?
- 14.7. Co się stanie w przypadku jednoczesnego rozpoczęcia nadawania przez dwie stacje w protokole ALOHA? W jaki sposób problem zostanie rozwiązany?
- 14.8. Wyjaśnij znaczenie słów, od których pochodzi akronim CSMA/CD.

- 14.9.** Wyjaśnij działanie mechanizmu binarnego wykładniczego wyznaczania przerwy w transmisji.
- 14.10.** Dlaczego w algorytmie CSMA/CD wykorzystuje się przerwę o losowej długości (podpowiedź: pomysł o sieci złożonej z wielu identycznych komputerów)?
- 14.11.** Dlaczego w sieciach bezprzewodowych niezbędny jest mechanizm CSMA/CA?

# Zawartość rozdziału

- 15.1. Wprowadzenie 275
- 15.2. Ethernet 275
- 15.3. Format ramki ethernetowej 276
- 15.4. Pole typu i demultipleksacja 276
- 15.5. Ethernet w wersji IEEE (802.3) 277
- 15.6. Połączenia sieci LAN i karty sieciowe 278
- 15.7. Rozwój Ethernetu — gruby Ethernet 278
- 15.8. Cienki Ethernet 279
- 15.9. Skrętka i koncentratory ethernetowe 280
- 15.10. Fizyczna i logiczna topologia Ethernetu 281
- 15.11. Okablowanie budynkowe 281
- 15.12. Odmiany okablowania i przepustowości 281
- 15.13. Złącza kabli ethernetowych 283
- 15.14. Podsumowanie 284

# 15

## *Przewodowe technologie LAN (Ethernet i 802.3)*

### **15.1. Wprowadzenie**

Ta część książki jest poświęcona technologiom sieci pakietowych. W rozdziale 13. przedstawiony został model IEEE 802 (wykorzystywany w sieciach LAN) oraz podwarstwy łączna logicznego (LLC) i dostępu do medium (MAC) (wchodzące w skład warstwy 2. stosu protokołów). Omówiono również 48-bitowy adres, który stanowi istotny element podwarstwy LLC. W rozdziale 14. omówiona została podwarstwa MAC oraz protokoły sterowania dostępem do medium, w tym CSMA/CD.

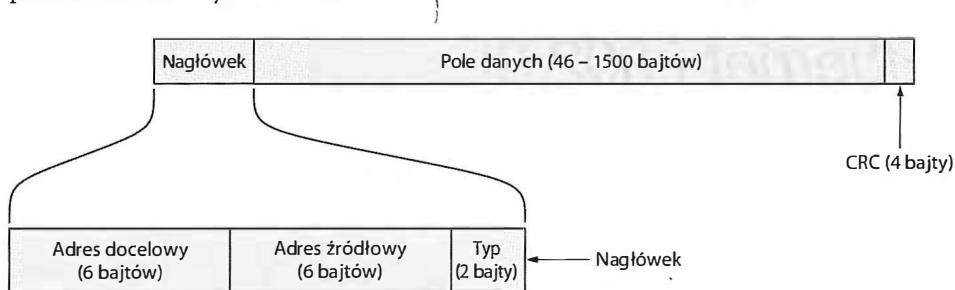
W tym rozdziale kontynuowany jest temat sieci lokalnych ze szczególnym uwzględnieniem przewodowych technologii LAN. Przedstawiono w nim sposoby wykorzystania opisanych wcześniej mechanizmów w budowie systemów ethernetowych, które zdominowały inne technologie przewodowych sieci lokalnych.

### **15.2. Ethernet**

Zgodnie z informacjami zamieszczonymi w rozdziale 14. Ethernet jest technologią sieci LAN, która została opracowana przez firmę Xerox PARC, a następnie ustandaryzowana przez konsorcjum Digital Equipment Corporation, Intel i Xerox. Istnieje na rynku od trzydziestu lat i choć urządzenia, okablowanie i media transmisyjne zmieniły się diametralnie, wiele z założeń funkcjonalnych rozwiązania nadal obowiązuje. Jednym z najciekawszych aspektów rozwoju sieci Ethernet jest zachowywanie zgodności z wcześniejszymi wersjami standardu. Urządzenia pracujące zgodnie z nowymi założeniami mają zdolność do wykrywania starszych jednostek i automatycznie przystosowują się do pracy z nimi.

### 15.3. Format ramki ethernetowej

Określenie **format ramki** odnosi się do sposobu budowania pakietu, czyli wyznaczania takich jego właściwości jak rozmiar oraz znaczenie poszczególnych pól. Zachowanie zgodności nowszych wersji standardu Ethernet z wcześniejszymi rozwiązaniami jest możliwe głównie dzięki temu, że od czasu opracowania specyfikacji DIX w 1970 roku format ramki nie uległ zmianie. Budowa ramki oraz znaczenie poszczególnych pól nagłówka zostały przedstawione na rysunku 15.1.



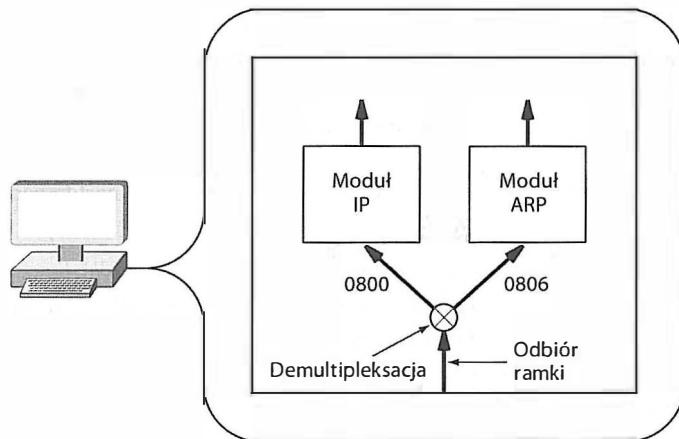
Rysunek 15.1. Format ramki ethernetowej

Z analizy rysunku wynika, że ramka Ethernet składa się z nagłówka o stałym rozmiarze, pola danych o zmiennej długości oraz dwubajtowej wartości algorytmu CRC<sup>39</sup>. Nagłówek zawiera trzy pola: 48-bitowy **adres docelowy** wskazujący odbiorcę, 48-bitowy **adres źródłowy** identyfikujący komputer nadawczy oraz 16-bitowe pole **typu** ramki.

### 15.4. Pole typu i demultiplexacja

Występujące w ramce Ethernet pole typu odpowiada za multipleksację i demultipleksację strumieni danych i umożliwia komputerowi korzystanie z wielu protokołów komunikacyjnych w tym samym czasie. W kolejnych rozdziałach zostały opisane różne protokoły używane w internecie i przenoszone w ramkach Ethernet. Przykładami mogą tutaj być datagramy IP oraz komunikaty ARP. Zarówno datagramom IP, jak i komunikatom ARP odpowiadają pewne niepowtarzalne wartości (w zapisie szesnastkowym są to odpowiednio wartości 0800 i 0806) identyfikujące daną formę komunikacji. Przekazując w ramce datagram, nadawca ustawia pole typu na 0800. Dzięki temu odbiorca może na podstawie pola typu przekazać zawartość pola danych do modułu odpowiedzialnego za przetwarzanie ramek zawierających datagramy. Operacja demultiplesacji została zilustrowana na rysunku 15.2.

<sup>39</sup> Podczas przesyłania ramki w sieci jest ona kodowana za pomocą algorytmu Manchester (opisanego w rozdziale 6.). Może więc być poprzedzona 64-bitową preambułą naprzemiennych jedynek i zer logicznych.

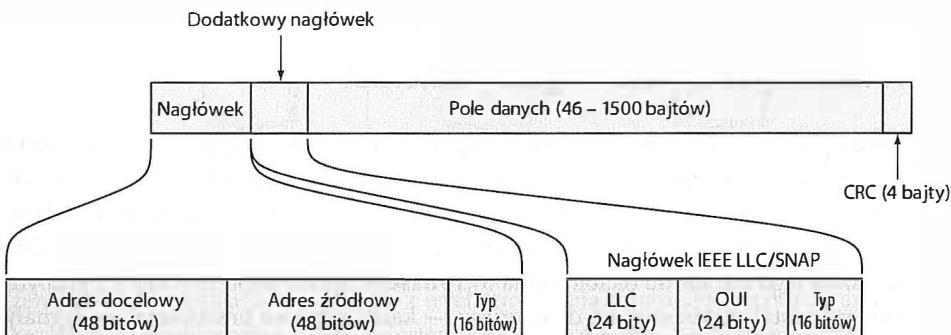


Rysunek 15.2. Wykorzystanie pola typu w procesie demultiplesacji

## 15.5. Ethernet w wersji IEEE (802.3)

W 1983 roku organizacja IEEE opracowała standard przeznaczony do stosowania w sieciach Ethernet i zaproponowała zmianę formatu ramki ethernetowej<sup>40</sup>. Grupa robocza pracująca nad wspomnianym standardem została oznaczona jako 802.3. Dlatego w celu odróżnienia rozwiązania IEEE od innych wiele osób posługuje się określeniem **Ethernet 802.3**.

Główna różnica między tradycyjnym Ethernetyem a Ethernetyem 802.3 wynika ze sposobu interpretacji pola typu. W standardzie 802.3 pole to służy do przechowywania informacji o **długości pakietu**, a definicja typu pakietu jest zapisywana w dodatkowym 8-bajtowym nagłówku. Dodatkowy nagłówek jest opisywany jako **nagłówek LLC/SNAP** (lub krócej **SNAP**) od angielskich słów *Logical Link Control/Sub-Network Access Protocol*, oznaczających sterowanie łączem logicznym i definiowanie protokołu dostępowego w podsieci. Format nagłówka pokazano na rysunku 15.3.



Rysunek 15.3. Format ramki IEEE 802.3 z nagłówkiem LLC/SNAP

<sup>40</sup> Działania IEEE nie zakończyły się sukcesem. Większość implementacji bazuje na oryginalnym formacie ramki.

Jak nietrudno zauważyc na rysunku, całkowity rozmiar ramki ethernetowej w standardzie 802.3 pozostaje taki sam, jak oryginalnej ramki Ethernet (1514 bajtów). Zalecenie IEEE spowodowało więc zmniejszenie dopuszczalnego rozmiaru pola danych z 1500 bajtów do 1492. Dodatkowe osiem bajtów nagłówka zajmuje bowiem przestrzeń zarezerwowaną wcześniej na dane.

Aby zachować zgodność dwóch wersji rozwiązań, zastosowano następującą konwencję:

*Jeśli trzynasty i czternasty bajt ramki Ethernet zawiera wartość liczbową mniejszą niż 1500, pole należy interpretować jako pole długości pakietu. Ramka podlega wówczas przetwarzaniu opisanemu w zaleceniu 802.3. Jeśli wartość jest większa od 1500, pole przechowuje informacje o typie pakietu, a ramkę należy przetwarzać zgodnie z pierwotnym standardem Ethernet.*

## 15.6. Połączenia sieci LAN i karty sieciowe

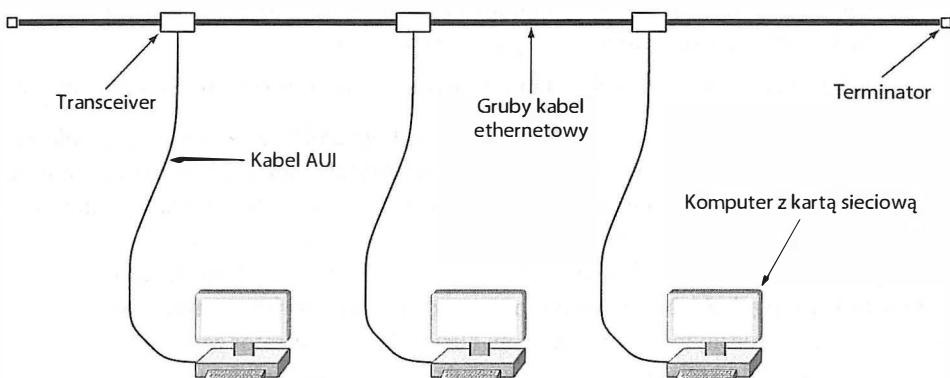
Z punktu widzenia architektury komputera sieć LAN wydaje się być urządzeniem wejścia-wyjścia, które jest przyłączane w ten sam sposób, jak dysk lub karta graficzna. Oczywiście, wymaga to przyłączenia do magistrali komputera **karty sieciowej** (NIC — ang. *Network Interface Card*). Do zadań karty sieciowej należą: weryfikacja adresu, wyliczanie wartości CRC i analiza ramek (karta sieciowa sprawdza, czy adres docelowy zapisany w ramce zgadza się z adresem komputera, i odrzuca ramki, które nie są przeznaczone dla danej jednostki). Ponadto odpowiada za przyłączenie komputera do sieci i obsługę mechanizmów transmisji danych (tj. wysyłanie i odbieranie ramek). Sama karta jest wykonywana w formie obwodu drukowanego ze złączem odpowiadającym wewnętrznej magistrali komputera oraz z zewnętrznym gniazdem, do którego można przyłączyć kabel z wtyczką właściwą dla określonej sieci LAN. Większość komputerów jest standardowo wyposażana w karty sieciowe. Niemniej karta sieciowa jest elementem niezależnym, który można wymienić w dowolnym momencie.

## 15.7. Rozwój Ethernetu — gruby Ethernet

Od czasu opracowania pierwszej wersji standardu Ethernet w roku 1970 technologia ta była wielokrotnie modyfikowana. Największe zmiany zawsze dotyczyły mediów transmisyjnych i okablowania. Pierwotne rozwiązanie było często nazywane **grubym Ethernetem**, ponieważ jako medium transmisyjne stosowano w nim gruby kabel współosiowy (formalna nazwa tego standardu okablowania to **10Base5**). Sprzęt współpracujący z grubym Ethernetem został podzielony na dwie grupy — karty sieciowe przetwarzające sygnały cyfrowe oraz oddzielne urządzenia elektroniczne (nazywane **transceiverami**), których zadaniem było wykrywanie nośnej w medium transmisyjnym, przekształcanie bitów na impulsy elektryczne oraz zamiana odbieranych sygnałów na strumienie bitowe.

Transceiver był połączony z kartą sieciową komputera za pomocą specjalnego kabla **interfejsu przyłączeniowego** (AUI — ang. *Attachment Unit Interface*). Sam transceiver

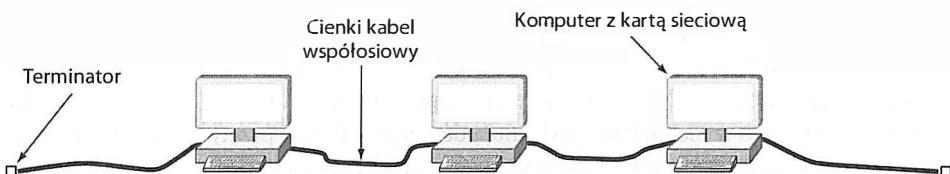
był zazwyczaj oddalony od komputera. Na przykład w sieci budynkowej mógł być przyłączony do magistrali ethernetowej ułożonej w podwieszonym suficie korytarza. Na rysunku 15.4 przedstawiono okablowanie charakterystyczne dla grubego Ethernetu z kablem AUI łączącym komputer z transceiverem.



Rysunek 15.4. Okablowanie grubego Ethernetu

## 15.8. Cienki Ethernet

Druga zaproponowana generacja okablowania zakładała wykorzystanie cieńskiego kabla współosiowego, który był zdecydowanie bardziej giętki od kabla grubego Ethernetu. Formalnie rozwiązanie to nazwano **10Base2**, a potocznie stosowano określenie **cienki Ethernet**. Zmiany nie ograniczały się jedynie do samego kabla magistralnego. Usunięto również połączenia AUI między komputerem i transceiverm, przenosząc moduł nadawczo-odbiorczy do wyposażenia karty sieciowej. W związku z tym kabel współosiowy musiał być ułożony między komputerami wchodzący w skład sieci. Rozwiązanie to zostało przedstawione na rysunku 15.5.



Rysunek 15.5. Druga generacja okablowania ethernetowego

Cienki Ethernet miał swoje wady i zalety. Największą zaletą było uproszczenie i obniżenie kosztów instalacji, wynikające z wyeliminowania transceiverów oraz z możliwości rozkładania okablowania w łatwo dostępnych miejscach (na przykład na biurku między komputerami, pod podłogą lub w duktach kablowych). Do wad rozwiązania z pewnością trzeba zaliczyć podatność na awarie. Jeśli którykolwiek z użytkowników rozłączył segment sieci (na przykład w celu przełożenia kabla lub podłączenia innego komputera), cała sieć przestała działać.

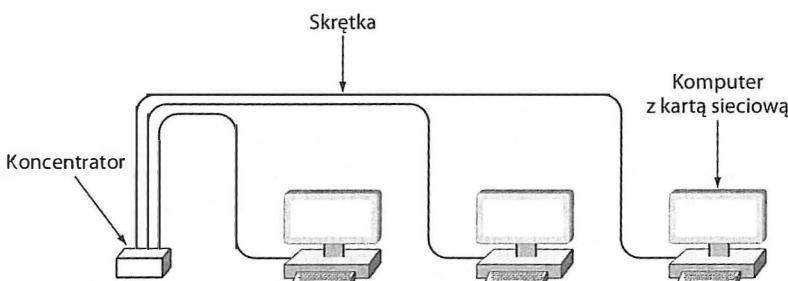
## 15.9. Skrótka i koncentratory ethernetowe

Trzecia generacja okablowania wprowadziła istotne zmiany do funkcjonowania sieci Ethernet.

- Kabel współosiowy został zastąpiony centralnym urządzeniem elektronicznym, niezależnym od komputerów przyłączonych do sieci.
- Zamiast niewygodnych kabli ekranowanych zastosowano skrótki niekranowane.

W związku z zastąpieniem kabli współosiowych skrótkami trzecia generacja okablowania ethernetowego często jest nazywana **ethernetem skrótkowym**. A ponieważ poprzednie wersje rozwiązań nie są obecnie wykorzystywane, wszelkie odniesienia do Ethernetu należy rozumieć jako odwołania do sieci Ethernet bazującej na skrótkach.

W pierwotnej wersji skrótkowego Ethernetu centralnym urządzeniem sieciowym był **koncentrator** (ang. *hub*). Na rynku oferowano różne odmiany koncentratorów, różniące się rozmiarami i zależnymi od nich cenami. Niewielki koncentrator zawierał od czterech do ośmiu **portów**, do których można było przyłączyć komputery lub inne urządzenia sieciowe (na przykład drukarki). Większe koncentratory obsługiwały setki przyłączów. Przykładowy schemat okablowania pokazano na rysunku 15.6.



Rysunek 15.6. Trzecia generacja Ethernetu, bazująca na skrótkach

Zawarte w koncentratorze moduły elektroniczne symulują przyłączenie do wspólnego kabla magistralnego, dzięki czemu cały system działa zgodnie z pierwotnymi założeniami standardu Ethernet. Komputery przyłączone do koncentratora uzyskują dostęp do sieci zgodnie z algorytmem CSMA/CD, odbierają kopie wszystkich przesyłanych ramek i na podstawie zawartego w nich adresu podejmują decyzję o dalszym przetwarzaniu lub odrzuceniu dostarczonych informacji. Zachowano również wcześniejszy format ramki. W praktyce oprogramowanie zainstalowane w komputerze nie może więc wykryć różnic między poszczególnymi wariantami wykonania sieci. Zainstalowana w komputerze karta sieciowa ukrywa szczegóły implementacyjne przed oprogramowaniem, które z niej korzysta.

Najważniejsze do zapamiętania jest to, że:

*W Ethernecie z okablowaniem skrótkowym zamiast wspólnego kabla wykorzystywane jest urządzenie elektroniczne nazywane koncentratorem.*

## 15.10. Fizyczna i logiczna topologia Ethernetu

Jak wiadomo, sieci LAN podlegają kategoryzacji na podstawie topologii (tj. kształtu sieci). Lista najważniejszych topologii sieciowych została przedstawiona na rysunku 13.4<sup>41</sup>. Można by więc zadać pytanie, jaka jest topologia Ethernetu. Odpowiedź jest jednak dość skomplikowana.

Pierwotną wersję rozwiązania bezsprzecznie można zaliczyć do topologii magistrali. Oryginalny Ethernet jest zresztą podawany jako sztandarowy przykład topologii magistralnych. Zastosowanie koncentratora spowodowało, że sieć przyjęła formę gwiazdy. Nawet termin **koncentrator** ma na celu podkreślenie, że jest to element centralny sieci. Jednak sposób działania urządzenia (emulowanie fizycznego kabla) sprawia, że funkcjonowanie sieci niczym nie różni się od konfiguracji, w której komputery przyłączano do wspólnego kabla. Inżynierowie sieciowi często żartowali sobie, że koncentrator jest tak naprawdę:

„magistralą w pudełku”

Zatem aby określić topologię Ethernetu, trzeba wprowadzić rozróżnienie na topologie **logiczne i fizyczne**. Logicznie Ethernet skrętkowy pracuje w topologii magistrali. Fizycznie odpowiada jednak topologii gwiazdy.

*Wprowadzenie rozróżnienia na topologie fizyczne i logiczne pozwala na stwierdzenie, że w warstwie fizycznej Ethernet skrętkowy odpowiada topologii gwiazdy. Natomiast logicznie jego działanie jest zgodne z topologią magistrali.*

## 15.11. Okablowanie budynkowe

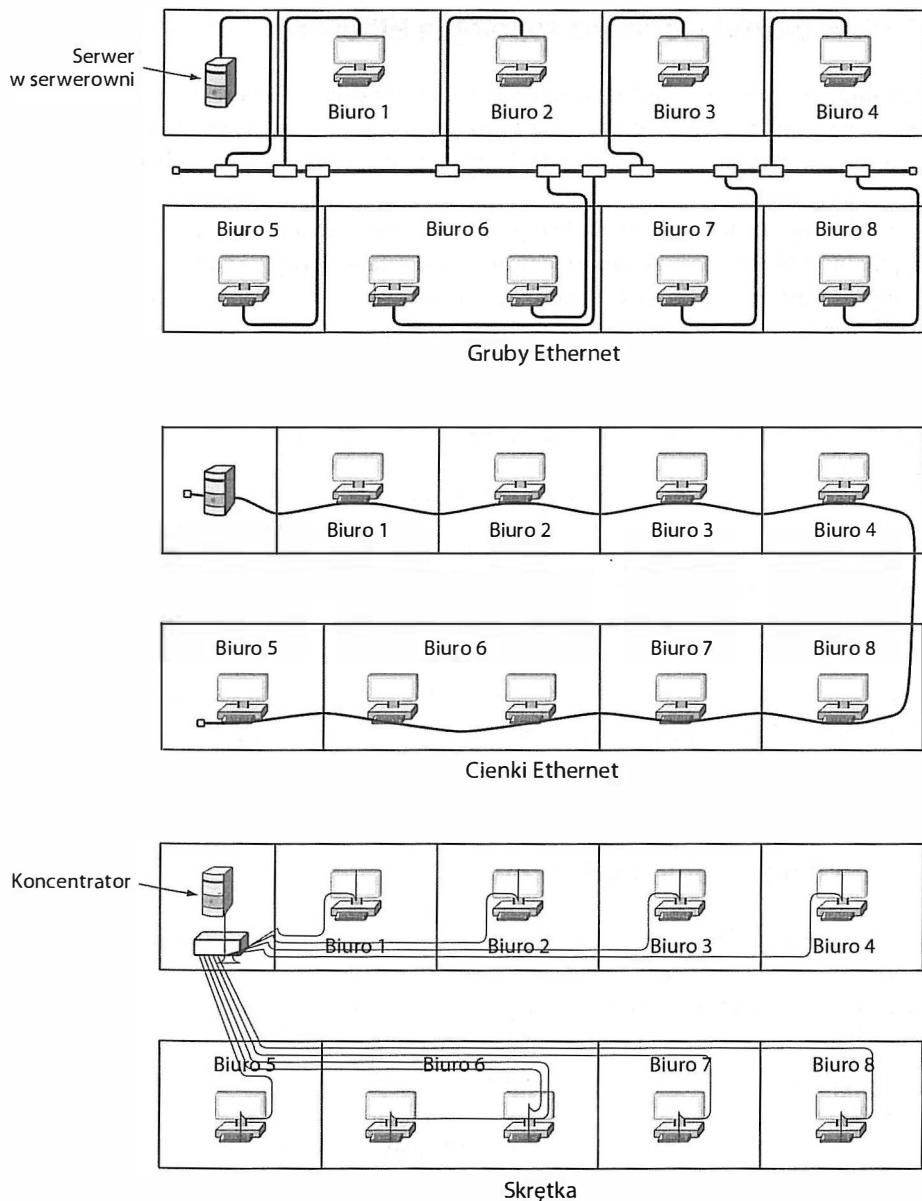
Sposób układania sieci LAN w serwerowni lub laboratorium akademickim nie ma szczególnego znaczenia. Jednak w przypadku instalacji budynkowych trzeba wziąć pod uwagę wiele parametrów, w tym rodzaj i liczbę przewodów, zasięg sieci oraz jej koszt. Trzy wersje okablowania ethernetowego odpowiadają trzem podstawowym praktycznym konfiguracjom sieci LAN. Przykłady rozłożenia kabli na jednym piętrze budynku pokazano na rysunku 15.7.

Z rysunku wynika, że budowa sieci na bazie skrętki wymaga ułożenia znacznie większej liczby kabli między biurami a punktem centralnym. Warto więc zadbać o właściwe oznaczenia poszczególnych przyłączyc.

## 15.12. Odmiany okablowania i przepustowości

Od czasu opracowania pierwszego łącza ethernetowego bazującego na skrętce wprowadzono wiele zmian wpływających na jakość transmisji oraz efektywność ekranowania przewodów. Modyfikacje te umożliwiły zwiększenie przepustowości sieci. W tabeli 15.1

<sup>41</sup> Rysunek 13.4 jest zamieszczony na stronie 250.



Rysunek 15.7. Różne warianty okablowania sieci LAN na jednym piętrze budynku

widoczne jest zestawienie trzech rodzajów skrętek ethernetowych oraz odpowiadających im szybkości transmisyjnych.

Z analizy zestawienia wynika, że pierwsza wersja Ethernetu bazującego na skrętce otrzymała oznaczenie **10BaseT**. Liczba 10 odpowiada w niej przepustowości łącząca 10 Mb/s. Kolejna wersja standardu jest znana jako szybki Ethernet, a jej oficjalne oznaczanie to **100BaseT** z uwagi na przepustowość 100 Mb/s. Trzecia wersja, nazywana gigabitowym Ethernetem, zapewnia przekazywanie danych z szybkością 1 Gb/s (czyli 1000 Mb/s).

**Tabela 15.1.** Rodzaje okablowania ethernetowego, przepustowości i kategorie skrętki

Oznaczenie	Nazwa	Przepustowość	Skrętka
10BaseT	Ethernet	10 Mb/s	Kategoria 3
100BaseT	Fast Ethernet (szybki Ethernet)	100 Mb/s	Kategoria 5
1000BaseT	Gigabit Ethernet (gigabitowy Ethernet)	1 Gb/s	Kategoria 6

Często łącza tego typu oznacza się skrótem **Gig-E**. W sieciach o wyższych przepustowościach koncentratory zostały zastąpione **przełącznikami** (więcej informacji na temat przełączników znajduje się w rozdziale 17.). Ponadto standardy wydajniejszych rozwiązań zawierają zastrzeżenie, że interfejsy sieciowe powinny mieć możliwość automatycznego wykrywania przepustowości łącza, co gwarantuje poprawne współdziałanie z urządzeniami starszego typu. Jeśli więc jedna wtyczka kabla jest przyłączona do urządzenia pracującego zgodnie ze standardem 10BaseT, a druga do interfejsu o przepustowości 1000BaseT, nowsze urządzenie automatycznie wykryje różnicę i zmniejszy szybkość transmisji do 10 Mb/s.

## 15.13. Złącza kabli ethernetowych

Skrętki ethernetowe są zakończone wtyczkami **RJ45**, czyli szerszymi wersjami wtyczek RJ11, znanych z przyłczy telefonicznych. Wtyczka RJ45 może być włożona do gniazda tylko w jeden sposób, a stanowiący element wtyczki zaczep zapobiega wysunięciu się kabla z gniazda. Nie można więc niepoprawnie przyłączyć kabla, a po podłączeniu nie ma ryzyka rozłączenia obwodu.

Gotowe kable (z założonymi wtyczkami) o różnej długości można bez trudu kupić w sklepie komputerowym. Niemniej przy wyborze odpowiedniego trzeba pamiętać, że istnieją dwa rodzaje kabli: **proste i z przeplotem**. Kable z przeplotem (stosowane do łączenia dwóch przełączników) łączą wyprowadzenia jednego interfejsu sieciowego z innymi wyprowadzeniami drugiego interfejsu. Kabel prosty (służący do łączenia komputera z przełącznikiem) łączy określone wyprowadzenia jednego gniazda RJ45 z dokładnie tymi samymi wyprowadzeniami drugiego gniazda RJ45. Wyprowadzenie 1. jest więc łączone z wyprowadzeniem 1. itd. Choć większość nowoczesnych kart sieciowych może wykrywać użycie niewłaściwego kabla i odpowiednio dostosować swoje działanie, pomyłka w doborze kabla może spowodować, że dane urządzenie nie będzie mogło korzystać z sieci.

Aby ułatwić technikom przygotowywanie kabli do użycia w sieci, przewody w skrętkach kategorii 5 i 6 są oznaczane różnymi kolorami izolacji. Lista kolorów oraz ich znaczenie zostały przedstawione w tabeli 15.2<sup>42</sup>.

<sup>42</sup> Skróty funkcji odpowiadają operacjom nadawania (TX), odbierania (RX) oraz komunikacji dwukierunkowej (BI) na każdej z czterech linii transmisyjnych.

Tabela 15.2. Znaczenie kolorów izolacji w kablach zakończonych wtyczkami RJ45

Wyprowadzenie we wtyczce RJ45	Kolor przewodu	Funkcja
1	biało-zielony	TX_D1+
2	zielony	TX_D1-
3	biało-pomarańczowy	RX_D2+
4	niebieski	BI_D3+
5	biało-niebieski	BI_D3-
6	pomarańczowy	RX_D2-
7	biało-brązowy	BI_D4+
8	brązowy	BI_D4-

## 15.14. Podsumowanie

Technologia Ethernet została opracowana w latach 70. ubiegłego wieku i stała się standardem sieci lokalnych. Ramki transmitowane w sieciach Ethernet zawierają 14-bajtowy nagłówek, w którym znajdują się: 48-bitowy adres docelowy, 48-bitowy adres źródłowy oraz 16-bitowe pole typu pakietu. Organizacja IEEE zdefiniowała własny format ramki (IEEE 802.3) z dodatkowym 8-bajtowym nagłówkiem. Jednak rozwiążanie to nie jest powszechnie wykorzystywane.

Pole typu umożliwia demultiplesowanie ramek odbieranych w stacji docelowej. Podczas formowania ramki nadawca określa typ pola danych, dzięki czemu odbiorca może przekazać pozyskane informacje do odpowiedniego modułu ich przetwarzania.

Od czasu publikacji pierwszej wersji standardu format ramki nie podlegał jakimkolwiek modyfikacjom. Nie można tego powiedzieć o okablowaniu, które zostało całkowicie zmienione. Jego rozwój doprowadził do powstania trzech ogólnych kategorii. Pierwszą z nich jest gruby Ethernet, w którym wykorzystuje się grube kable współosiowe oraz transceivery umieszczone poza komputerem. Druga kategoria to cienki Ethernet, bazujący na cienkim kablu współosiowym, rozciągniętym między kolejnymi jednostkami sieciowymi. W rozwiązaniach tego typu moduły nadawczo-odbiorcze są elementami karty sieciowej. Trzecia kategoria obejmuje systemy, w których wspólny kabel został zastąpiony centralnym urządzeniem (nazywanym **koncentratorem** lub **przełącznikiem**), a do przyłączania komputerów służy skrętka sieciowa. Powstała sieć ma fizyczną topologię gwiazdy, a logiczną topologię magistrali.

Podobnie jak wcześniejsze wersje Ethernetu, rozwiążanie bazujące na skrętce zapewnia przepustowość 10 Mb/s i jest oznaczane jako 10BaseT. Kolejna wersja, o formalnym oznaczeniu 100BaseT, działa z szybkością 100 Mb/s i jest nazywana szybkim Ethernetem. Trzecia wersja z kolei (gigabitowy Ethernet) umożliwia przekazywanie danych z przepływnością 1 000 Mb/s, czyli 1 Gb/s. Urządzenia o wyższych przepustowościach automatycznie wykrywają łączą o niższych szybkościach transmisyjnych i ograniczają przepływność emitowanych strumieni danych.

## ZADANIA

- 15.1. Jaki jest maksymalny rozmiar ramki ethernetowej z uwzględnieniem wartości CRC?
- 15.2. W jaki sposób w standardzie Ethernet wykorzystywane jest pole typu?
- 15.3. Jaki jest maksymalny rozmiar pola danych w standardzie 802.3?
- 15.4. W jaki sposób odbiornik może ustalić, czy ramka została sformatowana zgodnie ze standardem Ethernet, czy 802.3?
- 15.5. W której części ramki umieszczany jest nagłówek LLC/SNAP (jeśli w ogóle jest wykorzystywany)?
- 15.6. W jaki sposób można przyłączyć komputer do grubego Ethernetu?
- 15.7. W jaki sposób przyłączano komputery do cienkiego Ethernetu?
- 15.8. Czym jest koncentrator? Jakie okablowanie jest wykorzystywane w sieciach zawierających koncentratory?
- 15.9. Wyszukaj w internecie informacje na temat koncentratorów i przełączników. Które urządzenie (przełącznik czy koncentrator) wybierzesz, jeśli obydwa będą miały taką samą szybkość transmisji danych i tę samą cenę? Uzasadnij odpowiedź.
- 15.10. Podaj przykład sieci o topologii logicznej innej niż fizyczna.
- 15.11. Która forma okablowania ethernetowego wymaga zastosowania największej liczby kabli?
- 15.12. Jakie kategorie kabli zapewniają przepustowość 10 Mb/s, 100 Mb/s i 1000 Mb/s?

# Zawartość rozdziału

- 16.1. Wprowadzenie 287
- 16.2. Podział sieci bezprzewodowych 287
- 16.3. Sieci osobiste (PAN) 288
- 16.4. Pasmo ISM w sieciach LAN i PAN 288
- 16.5. Technologie bezprzewodowych sieci lokalnych i Wi-Fi 289
- 16.6. Techniki rozpraszania widma 290
- 16.7. Inne standardy bezprzewodowych sieci LAN 291
- 16.8. Architektura bezprzewodowej sieci LAN 292
- 16.9. Nakładanie obszarów, stwarzyszanie się urządzeń  
i format ramki 802.11 293
- 16.10. Koordynacja działań punktów dostępowych 293
- 16.11. Rywalizacja o dostęp i obsługa bezkolizyjna 294
- 16.12. Technologie bezprzewodowych sieci MAN i standard WiMAX 296
- 16.13. Technologie i standardy sieci PAN 298
- 16.14. Inne technologie komunikacji na niedużych odległościach 300
- 16.15. Technologie bezprzewodowych sieci WAN 300
- 16.16. Klastry komórek i wielokrotne wykorzystywanie częstotliwości 302
- 16.17. Generacje technologii komórkowych 303
- 16.18. Technologia satelitarna VSAT 306
- 16.19. Satelity GPS 307
- 16.20. Radio programowe i przyszłość technologii bezprzewodowych 308
- 16.21. Podsumowanie 309

# 16

## *Technologie sieci bezprzewodowych*

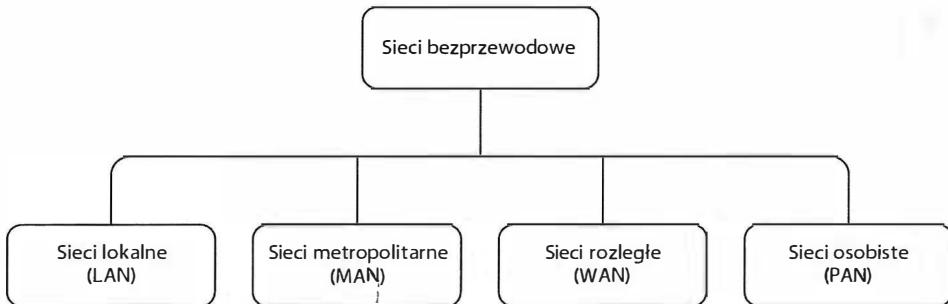
### **16.1. Wprowadzenie**

Rozdziały tej części książki koncentrują się na technologiach sieciowych stosowanych w systemach pakietowych. Zawierają omówienie technik przełączania pakietów, a także charakterystykę modelu sieci zaproponowanego przez organizację IEEE. W poprzednim rozdziale zostały opisane rozwiązania przewodowe właściwe dla sieci LAN.

Tematem tego rozdziału są natomiast sieci bezprzewodowe. Wśród omawianych zagadnień uwzględniono liczne propozycje rozwiązań z dziedziny komunikacji bezprzewodowej, które umożliwiają wymianę danych na różnych odległościach i są stosowane w wielu komercyjnych systemach telekomunikacyjnych. W przeciwieństwie do sieci przewodowych, transmisje bezprzewodowe są realizowane z wykorzystaniem wielu technologii, z których część ma bardzo podobne charakterystyki.

### **16.2. Podział sieci bezprzewodowych**

W skład rozwiązań z zakresu komunikacji bezprzewodowej wchodzą sieci o różnych rozmiarach i różnych sposobach działania. Różnorodność technologii wynika częściowo z regulacji rządowych dotyczących udostępniania częstotliwości radiowych na potrzeby transmisji danych. Wykorzystanie niektórych zakresów częstotliwościowych wymaga uzyskania specjalnego zezwolenia. Z kolei komunikacja w innych obszarach widma częstotliwościowego nie jest objęta licencjonowaniem. Sytuacja ta jest przyczyną powstania wielu niezależnych od siebie technologii bezprzewodowych, które wciąż są uzupełniane o nowe warianty. Ogólna kategoryzacja systemów komunikacji bezprzewodowej bazuje na rodzajach sieci, co zostało pokazane na rysunku 16.1.



Rysunek 16.1. Podział technologii sieci bezprzewodowych

### 16.3. Sieci osobiste (PAN)

Poza trzema rodzajami sieci znymi z rozwiązań przewodowych (LAN, MAN i WAN) w komunikacji bezprzewodowej wyróżnia się także **sieci osobiste** (PAN — ang. *Personal Area Network*). Technologia PAN zakłada wymianę danych na bardzo krótkich odległościach i obejmuje transmisję między urządzeniami należącymi do jednej osoby. System PAN umożliwia na przykład komunikację między telefonem komórkowym a zestawem słuchawkowym. Znajduje również zastosowanie w łączeniu komputera z urządzeniami towarzyszącymi, takimi jak myszka lub klawiatura.

Rozwiązania PAN można podzielić na trzy grupy. Ich nazwy wraz z krótkim opisem zostały wymienione w tabeli 16.1. W dalszej części rozdziału znajduje się szczegółowe omówienie każdego mechanizmu (uwzględniające również konkretne standardy PAN).

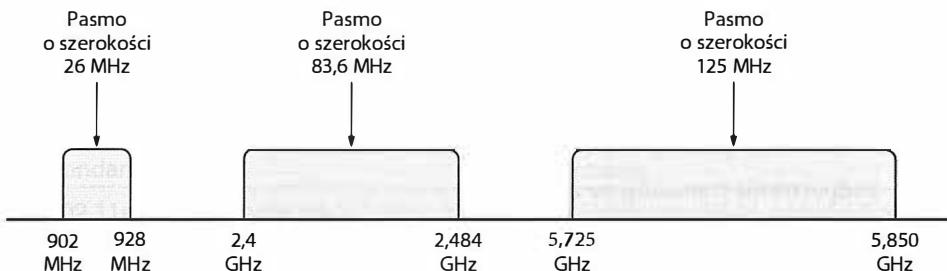
Tabela 16.1. Trzy podstawowe rodzaje sieci osobistych

Rodzaj	Przeznaczenie
Bluetooth	Komunikacja na bardzo małych odległościach między urządzeniami takimi jak myszka lub zestaw słuchawkowy oraz komputerami bądź telefonami komórkowymi.
Podczerwień	Komunikacja między urządzeniami pozostającymi w bezpośredniej widoczności względem siebie. Zazwyczaj ogranicza się do przesyłania danych między pilotem a komputerem lub elementami kina domowego.
Komunikacja w paśmie ISM	Komunikacja na częstotliwościach przeznaczonych do wykorzystania w przemyśle, nauce i medycynie. Narażona na zakłócenia elektromagnetyczne.

### 16.4. Pasmo ISM w sieciach LAN i PAN

Rządy wielu krajów zarezerwowały trzy przedziały częstotliwościowe do wykorzystania przez przedstawicieli **przemysłu, nauki i medycyny** (ISM — ang. *Industrial, Scientific and Medical*). Komunikacja w paśmie ISM oznacza wykorzystanie częstotliwości, które nie

podlegają licencjonowaniu i nie należą do żadnego operatora. Są więc dostępne dla producentów urządzeń, co znajduje swoje odzwierciedlenie w mnogości urządzeń LAN i PAN. Zakresy częstotliwości pasm ISM zostały wymienione na rysunku 16.2.



Rysunek 16.2. Przedziały częstotliwości pasm ISM wraz z informacją o szerokości każdego z pasm

## 16.5. Technologie bezprzewodowych sieci lokalnych i Wi-Fi

Komunikacja bezprzewodowa w sieciach lokalnych jest realizowana na różnych częstotliwościach, z różnymi modulacjami i szybkościami transmisyjnymi. Większość stosowanych rozwiązań jest zgodna z opracowanymi przez IEEE standardami z grupy IEEE 802.11. W 1999 roku sieciami tego typu zajęła się również grupa producentów sprzętu bezprzewodowego, która powołała do życia stowarzyszenie Wi-Fi Alliance, czyli organizację non profit, której celem jest testowanie i certyfikowanie urządzeń spełniających standardy 802.11. Dzięki intensywnym działaniom marketingowym większość odbiorców utożsamia sieci LAN z nazwą Wi-Fi. W tabeli 16.2 przedstawiono najważniejsze standardy IEEE znajdujące się w kręgu zainteresowania stowarzyszenia Wi-Fi.

Tabela 16.2. Najważniejsze standardy komunikacji bezprzewodowej certyfikowane przez stowarzyszenie Wi-Fi

Standard IEEE	Pasmo częstotliwościowe	Przepustowość	Modulacja	Multipleksacja
pierwotny 802.11	2,4 GHz	1 lub 2 Mb/s	FSK	DSSS
	2,4 GHz	1 lub 2 Mb/s	FSK	FHSS
	Podczerwień	1 lub 2 Mb/s	PPM	brak
802.11a	5,725 GHz	od 6 do 54 Mb/s	PSK lub QAM	OFDM
802.11b	2,4 GHz	5,5 i 11 Mb/s	PSK	DSSS
802.11g	2,4 GHz	22 i 54 Mb/s	różne	OFDM

## 16.6. Techniki rozpraszania widma

Pojęcie **rozpraszania widma** wystąpiło w rozdziale 11. Zgodnie z zawartymi w tym rozdziale informacjami rozpraszanie widma polega na wykorzystaniu wielu częstotliwości do przesyłania danych. Nadajnik rozkłada strumienie danych na kilka częstotliwości, a odbiornik zbiera informacje z kilku częstotliwości i łączy je w jeden wynikowy zbiór danych.

Rozpraszanie widma ma na celu:

- zwiększenie ogólnej przepustowości systemu,
- uodpornienie transmisji na zakłócenia.

W tabeli 16.3 wymieniono trzy najważniejsze techniki multipleksacji stosowane w sieciach bezprzewodowych Wi-Fi.

Tabela 16.3. Najważniejsze techniki multipleksacji w sieciach Wi-Fi

Skrót nazwy	Znaczenie	Opis
DSSS	Rozpraszanie widma przez kluczowanie bezpośrednie (ang. <i>Direct Sequence Spread Spectrum</i> )	Działanie podobne do CDMA, w którym nadawca mnoży ciąg danych przez specjalny ciąg kodowy, powodując rozszerzenie zakresu częstotliwościowego sygnału. Odbiornik wykorzystuje tę samą sekwencję kodową do odtworzenia danych.
FHSS	Rozpraszanie widma przez skakanie po częstotliwościach (ang. <i>Frequency Hopping Spread Spectrum</i> )	Nadajnik emituje informacje na zmieniających się odpowiednio częstotliwościach. Odbiornik stosuje tę samą sekwencję zmian częstotliwości do odtworzenia strumienia danych.
OFDM	Ortogonalne zwielokrotnianie częstotliwościowe (ang. <i>Orthogonal Frequency Division Multiplexing</i> )	Transmisja wielotonowa, w której pasmo jest podzielone na wiele częstotliwości nośnych. Dzięki odpowiedniemu doborowi częstotliwości transmitowane sygnały nie zakłócają się wzajemnie.

Każda technika ma pewne wady i zalety. Rozwiązanie OFDM cechuje się największą elastycznością w zastosowaniu. Technika DSSS gwarantuje najwyższą wydajność, a FHSS zapewnia odporność na zakłócenia. Projektanci rozwiązań transmisyjnych dobierają więc wariant multipleksacji właściwy do konkretnych zastosowań. Na przykład w celu wykorzystania mechanizmów DSSS i FHSS opracowano dwie wersje standardu 802.11. Podsumowując:

Techniki rozpraszania widma umożliwiają poprawne funkcjonowanie bezprzewodowych sieci LAN w środowiskach o dużym poziomie zakłóceń.

## 16.7. Inne standardy bezprzewodowych sieci LAN

Organizacja IEEE opracowała wiele standardów sieci bezprzewodowych uwzględniających różne techniki komunikacji. Każda specyfikacja definiuje zakres częstotliwości rozwiązań, modulację i typ multipleksacji oraz przepustowość połączeń. W tabeli 16.4 widnieje wykaz najważniejszych standardów wraz z krótką charakterystyką każdego z nich.

Tabela 16.4. Najważniejsze standardy z grupy 802.11 i ich przeznaczenie

Standard	Przeznaczenie
802.11e	Zwiększoną jakość usług (na przykład gwarancje niskiej wartości fluktuacji opóźnień).
802.11h	Podobny do 802.11a, ale z dodatkową kontrolą szerokości widma i mocy nadawczej (przeznaczony do stosowania w Europie).
802.11i	Zwiększone bezpieczeństwo. Uwzględnia standard zaawansowanego szyfrowania (AES — ang. <i>Advanced Encryption Standard</i> ). Znany jako standard WPA2.
802.11k	Zarządzanie zasobami radiowymi, w tym sterowanie mocą.
802.11n	Przepustowość powyżej 100 Mb/s. Przeznaczony do obsługi aplikacji multimedialnych (głównie wideo). Szybkość transmisji może osiągać 500 Mb/s.
802.11p	Specjalistyczny system komunikacji na krótkich odległościach (DSRC — ang. <i>Dedicated Short-Range Communication</i> ), przeznaczony do wymiany danych między samochodami jadącymi po autostradzie lub między samochodami i urządzeniami infrastruktury autostradowej.
802.11r	Zwiększoną zdolność do przemieszczania się urządzeń między obszarami działania punktów dostępowych (bez utraty łączności).
802.11s	Propozycja sieci siatkowej, której węzły automatycznie ustanawiają połączenia między sobą i przekazują pakiety.

W 2007 roku organizacja IEEE zebrała wszystkie specyfikacje z grupy 802.11 i wydała jeden dokument o nazwie **802.11-2007**, w którym opisano podstawowe założenia komunikacji bezprzewodowej i dołączono załączniki charakteryzujące jej poszczególne warianty.

Najważniejszą rzeczą do zapamiętania jest to, że:

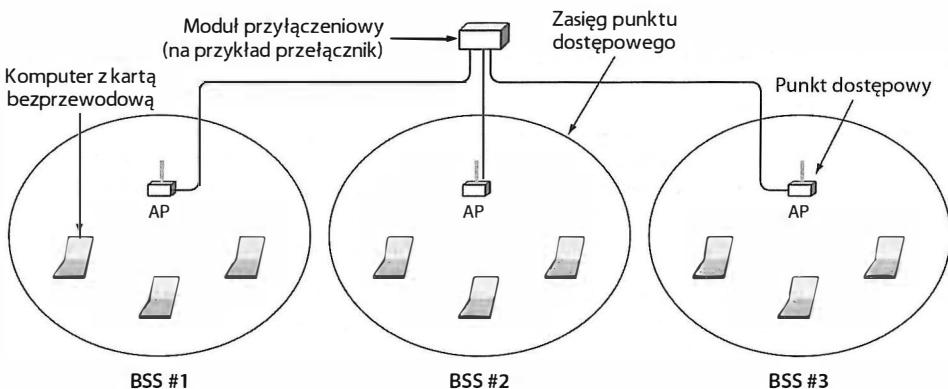
*Zaproponowano i opracowano wiele wariantów komunikacji w ramach standardu 802.11. Każdy z nich ma pewne wady i zalety.*

## 16.8. Architektura bezprzewodowej sieci LAN

Bezprzewodowa sieć LAN składa się z trzech podstawowych elementów: **punktu dostępowego** (nazywanego niekiedy **stacją bazową**), komponentu przyłączeniowego (takiego jak przełącznik lub router, do którego jest przyłączony punkt dostępowy) oraz zbioru **stacji bezprzewodowych**, nazywanych również **węzłami bezprzewodowymi**. Połączenia w sieci bezprzewodowej są zaliczane do jednej z dwóch kategorii:

- Ad hoc — jednostki bezprzewodowe komunikują się ze sobą bez udziału punktów dostępowych.
- Połączenia infrastrukturalne — stacje bezprzewodowe wymieniają dane jedynie z punktami dostępowymi, które pośredniczą w przekazywaniu pakietów.

W praktyce rzadko stosuje się rozwiązania z typu ad hoc. Firmy i dostawcy usług internetowych instalują zazwyczaj kilka punktów dostępowych, z którymi komunikują się wszystkie komputery pozostające w zasięgu poszczególnych urządzeń. Na rysunku 16.3 przedstawiono przykład rozwiązania, które mogłoby zostać wdrożone w budynku przedsiębiorstwa lub uniwersytetu.



Rysunek 16.3. Architektura infrastrukturalna w bezprzewodowej sieci LAN

Połączenia przewodowe doprowadzane do punktów dostępowych są zazwyczaj wykonywane w formie skrętkowego Ethernetu. Zbiór komputerów pozostających w zasięgu punktu dostępowego jest nazywany **podstawowym zbiorem usługowym** (BSS — ang. *Basic Service Set*)<sup>43</sup>. W konfiguracji przedstawionej na rysunku wydzielono trzy zbiory BSS, odpowiadające kolejnym punktom dostępowym.

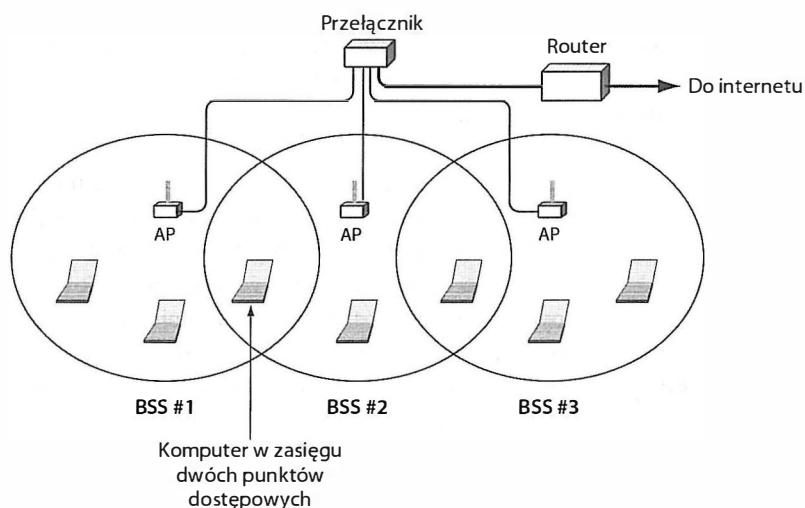
Podsumowując:

Większość bezprzewodowych sieci LAN jest zbudowana zgodnie z architekturą infrastrukturalną, w której każdy komputer komunikuje się z punktem dostępowym (stacją bazową).

<sup>43</sup> Obszar objęty zasięgiem punktu dostępowego jest czasami nazywany **komórką** (podobnie jak w telefonii komórkowej).

## 16.9. Nakładanie obszarów, stwarzyszenie się urządzeń i format ramki 802.11

Działanie sieci w architekturze infrastrukturalnej komplikuje wiele czynników. Jeśli dwa punkty dostępowe są za bardzo od siebie oddalone, powstaje między nimi **martwa strefa**, czyli obszar, w którym nie można uzyskać połączenia bezprzewodowego. Z drugiej strony, zbyt mała odległość między punktami dostępowymi powoduje nakładanie się obszarów objętych zasięgiem, a to z kolei umożliwia jednostkom bezprzewodowym komunikację z dwoma urządzeniami jednocześnie. Ponadto większość bezprzewodowych sieci LAN wymaga dostępu do internetu. Konieczne jest zastosowanie komponentu, który zagwarantuje kablowe przyłączenie punktu dostępowego do routera sieciowego. Przykładowa architektura sieci została pokazana na rysunku 16.4.

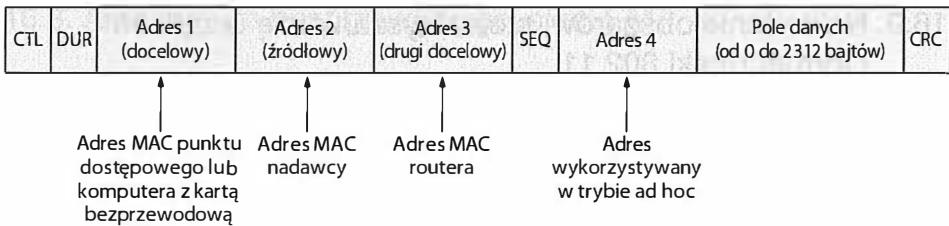


Rysunek 16.4. Przykład sieci z nakładającymi się obszarami działania punktów dostępowych

Aby wyeliminować problem nakładających się obszarów działania punktów dostępowych, sieci zgodne ze standardem 802.11 nakładają na jednostkę sieciową obowiązek **stwarzyszenia się** z pojedynczym punktem dostępowym. Oznacza to, że dana stacja może przesyłać ramki tylko do jednego punktu dostępowego, który z kolei przekaże je do innych jednostek sieci. Format ramki 802.11 został przedstawiony na rysunku 16.5. Widać na nim, że działanie w konfiguracji infrastrukturalnej wymaga zapisania w ramce adresu MAC punktu dostępowego oraz adresu routera internetowego.

## 16.10. Koordynacja działań punktów dostępowych

Po zapoznaniu się z zamieszczonymi wcześniej informacjami można dojść do wniosku, że działanie punktów dostępowych powinno być koordynowane. Ale w jakim zakresie? Wiele pierwotnych rozwiązań konstrukcyjnych było niezwykle skomplikowanych ze



Rysunek 16.5. Format ramki 802.11 stosowanej w bezprzewodowych sieciach LAN

względzie na to, że punkty dostępowe zapewniały możliwość niezakłóconego przemieszczania się między obszarami ich działania, tak jak w przypadku telefonii komórkowej. Urządzenia komunikowały się ze sobą, dbając o to, by przemieszczający się komputer bez problemu „przeszedł” z obszaru objętego zasięgiem jednego punktu dostępowego do obszaru obsługiwanej przez inny punkt dostępowy. Jednym ze sposobów realizacji tego zadania było mierzenie poziomu sygnału i przekazywanie jednostki pod kontrolę kolejnego urządzenia, gdy moc sygnału rejestrowana w pierwotnym punkcie dostępowym okazywała się niższa od mierzonej w drugim punkcie.

Jednak z czasem producenci zaczęli oferować tańsze urządzenia, które nie miały zaimplementowanych funkcji przekazywania kontroli nad stacjami roboczymi. Posunięcia takie uzasadniano tym, że poziom sygnału nie oddaje w pełni zdolności stacji do przemieszczania się, że komputery przenośne mogą we własnym zakresie obsługiwać przełączanie z jednego punktu dostępowego do drugiego oraz że sieć przewodowa, do której przyłączane są punkty dostępowe, dysponuje dostatecznie dużą pojemnością, aby można w niej było implementować bardziej skoncentrowane mechanizmy koordynacji pracy urządzeń. Stosowanie punktów dostępowych o mniejszej złożoności jest szczególnie uzasadnione w systemach, w których występuje tylko jeden taki komponent.

Podsumowując:

*Produkowane są dwa rodzaje punktów dostępowych — bardziej rozbudowane, które umożliwiają bezproblemowe przełączanie się stacji między obszarami radiozymi, oraz tańsze, które działają niezależnie i przenoszą na komputery obowiązek stwarzyszania się z kolejnymi punktami dostępowymi.*

## 16.11. Rywalizacja o dostęp i obsługa bezkolizyjna

W pierwotnym standardzie 802.11 zdefiniowano dwa ogólne mechanizmy zarządzania dostępem do kanału radiowego:

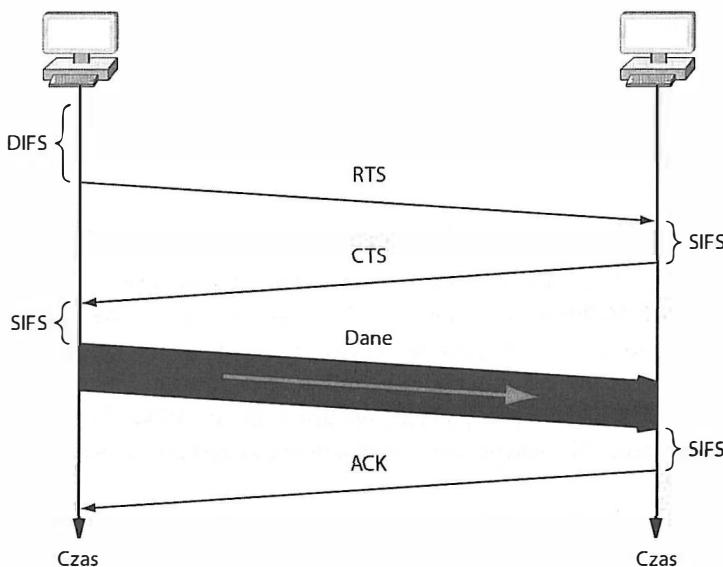
- Funkcję koordynacji dostępu w punkcie dostępowym (PCF — ang. *Point Coordination Function*), zapewniającą bezkolizyjną obsługę stacji roboczych,
- Rozproszoną funkcję koordynacji dostępu (DCF — ang. *Distributed Coordination Function*), dopuszczającą rywalizację o dostęp.

Pierwsza z wymienionych funkcji nakłada na punkt dostępowy obowiązek nadzorowania stacji działających w obszarze BSS w celu zagwarantowania transmisji, które nie będą się wzajemnie nakładały. Punkt dostępowy może na przykład przydzielić każdej ze stacji oddzielną częstotliwość nadawczą. W praktyce mechanizm PCF nie jest jednak stosowany.

Funkcja rozproszonego zarządzania dostępem wymusza na jednostkach działających w obszarze BSS korzystanie z protokołu dostępu swobodnego. Jak wiadomo z rozdziału 14., w sieciach bezprzewodowych występuje **problem ukrytych stacji** (polegający na tym, że wymiana danych realizowana między dwoma stacjami może nie zostać wykryta przez trzecią jednostkę). Wiadomo również, że rozwiązaniem tego problemu jest zastosowanie algorytmu CSMA/CA, który nakłada na stacje obowiązek wymiany komunikatów RTS i CTS przed rozpoczęciem każdej emisji pakietu. W standardzie 802.11 zdefiniowano dodatkowo kilka parametrów operacji, które zostały pominięte w opisie zamieszczonym w rozdziale 14. Na przykład wyznaczone zostały trzy zależności czasowe:

- SIFS — krótka przerwa międzyramkowa (ang. *Short Inter-Frame Space*) o wartości 10 $\mu$ s.
- DIFS — przerwa międzyramkowa algorytmu DCF (ang. *DCF Inter-Frame Space*) o wartości 50 $\mu$ s.
- Czas trwania szczeliny nadawczej (slotu) o wartości 20 $\mu$ s.

Parametr SIFS wyznacza czas oczekiwania stacji przed rozpoczęciem nadawania potwierdzenia lub innej odpowiedzi. Wartość DIFS, będąca sumą wartości SIFS i czasu dwóch szczelin nadawczych, określa czas nieaktywności stacji przed próbą rozpoczęcia transmisji. Praktyczne wykorzystanie wymienionych parametrów zostało zilustrowane na rysunku 16.6.



Rysunek 16.6. Działanie algorytmu CSMA/CA z uwzględnieniem parametrów SIFS i DIFS

Najważniejsze jest to, że:

*Stosowany w sieciach Wi-Fi algorytm CSMA/CA uwzględnia zależności czasowe, które określają czas oczekiwania stacji przed przesaniem pakietu inicjującego transmisję oraz przed odesaniem odpowiedzi.*

Odległości między stacjami oraz zakłócenia elektromagnetyczne uniemożliwiają rozróżnienie słabych sygnałów, interferencji i kolizji. Dlatego w sieciach Wi-Fi nie jest wykorzystywany mechanizm detekcji kolizji. Urządzenia nie próbują więc wykrywać przypadków nakładania się sygnałów podczas emisji. Oczekują natomiast na komunikat potwierdzający dostarczenie ramki (ACK). Jeśli informacja ACK nie zostanie przesłana, nadawca zakłada, że wyemitowana ramka została utracona, i uruchamia procedurę **wstrzymania transmisji** (ang. *backoff*), podobną do stosowanej w Ethernetie przewodowym. W praktyce w sieciach złożonych z kilku stacji, w których nie występują zakłócenia elektromagnetyczne, mechanizm retransmisji nie jest wykorzystywany. Niemniej w bardziej rozbudowanych konfiguracjach może dochodzić do częstej utraty pakietów, co wymaga stosowania retransmisji.

## 16.12. Technologie bezprzewodowych sieci MAN i standard WiMAX

W ogólnym ujęciu technologie MAN nie okazały się komercyjnym sukcesem. Wyjątkiem jest jeden standard komunikacji bezprzewodowej — rozwiązanie opisane przez organizację IEEE w specyfikacji **802.16**. Producenci urządzeń pracujących w standardzie 802.16 nadali mu nazwę **WiMAX**, od angielskich słów *World-wide Interoperability for Microwave Access*, czyli ogólnoświatowy standard współdziałania w dostępie mikrofalowym. Ponadto powołali do życia **Forum WiMAX**, które ma promować tę technologię.

Standard WiMAX został opracowany w dwóch wersjach, różniących się ogólnymi założeniami. Poszczególne warianty rozwiązania określa się jako:

- stały WiMAX,
- mobilny WiMAX.

**Stały WiMAX.** Określenie to odnosi się do systemów zbudowanych zgodnie ze specyfikacją IEEE **802.16-2004**, która nosi również nazwę **802.16d**. Słowo **stały** informuje o tym, że w standardzie nie uwzględniono mechanizmu zmiany punktu dostępowego. Jest on więc przeznaczony do realizacji połączeń między dostawcami usług internetowych a odbiorcami, którzy nie zmieniają swojej lokalizacji (na przykład do przyłączania sieci domowej do internetu). Nie nadaje się natomiast do wykorzystania w sieci telefonii komórkowej.

**Mobilny WiMAX.** Określenie to odnosi się do systemów utworzonych zgodnie ze specyfikacją **802.16-2005**, która nosi również nazwę **802.16e**. Słowo **mobilny** informuje o tym, że w standardzie uwzględniono mechanizmy przełączania jednostek między punk-

tami dostępowymi. Dzięki temu mobilne warianty systemu WiMAX nadają się do stosowania w środowiskach, w których pracują komputery przenośne oraz telefony komórkowe.

Rozwiązanie WiMAX zapewnia szerokopasmową łączność o różnorakim przeznaczeniu. Część dostawców usług internetowych planuje zastosowanie tej technologii w formie łączów dostępowych na odcinku „ostatniej mili”. Inni dostrzegają w niej potencjał łączenia różnych lokalizacji (szczególnie na terenie miasta) i świadczenia usług transmisji danych ogólnego przeznaczenia. Kolejne możliwe zastosowanie to łączenie serwerowni dostawców usług internetowych z ich zdalnymi modułami (na przykład ze stacjami bazowymi). Lista ewentualnych zastosowań została przedstawiona w tabeli 16.5.

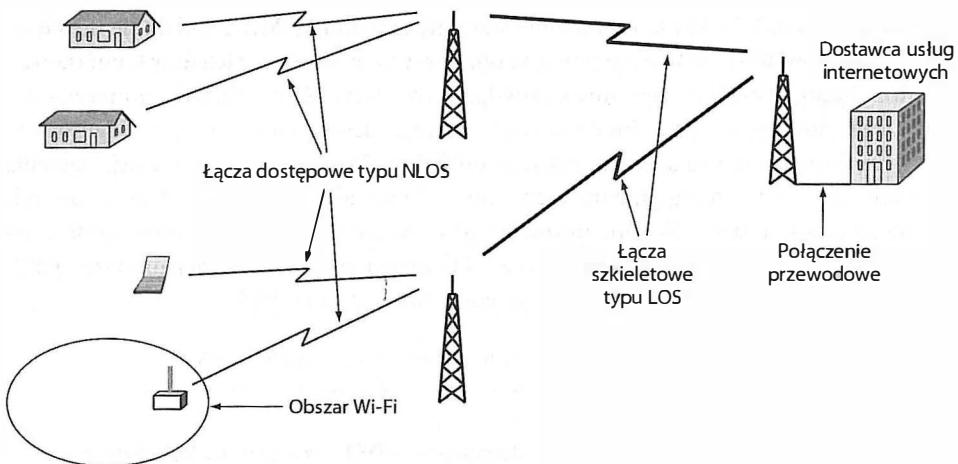
Tabela 16.5. Potencjalne zastosowania technologii WiMAX

<b>Łącza dostępowe</b>
<ul style="list-style-type: none"><li>• Łącza ostatniej mili — alternatywa dla połączeń DSL i modemów kablowych.</li><li>• Wysokowydajne połączenia dla użytkowników zmieniających swoją lokalizację.</li><li>• Unifikacja łączów przeznaczonych do transmisji danych i świadczenia usług telekomunikacyjnych.</li><li>• Zapasowe łącze dostępu do internetu.</li></ul>
<b>Łącza międzysieciowe</b>
<ul style="list-style-type: none"><li>• Łącza szkieletowe między punktami dostępowymi Wi-Fi a siecią dostawcy usług internetowych.</li><li>• Prywatne połączenia między sieciami korporacyjnymi.</li><li>• Połączenia między małymi a dużymi firmami dostawców usług internetowych.</li></ul>

Instalacje WiMAX realizujące zadanie szkieletowych cechują się najwyższą przepływnością bitową i wykorzystują częstotliwości wymagające bezpośredniej widoczności między urządzeniami biorącymi udział w transmisji danych. Połączenia tego typu opisuje się skrótem LOS (ang. *Line-Of-Sight*). Stacje LOS są zazwyczaj montowane na masztach lub na dachach budynków. Instalacje zapewniające dostęp do internetu bazują na stałych lub mobilnych rozwiązaniach WiMAX, w których nie jest konieczna bezpośrednia widoczność między urządzeniami. Do ich oznaczenia stosuje się skrót NLOS (ang. *Non-Line-Of-Sight*). Przykład obydwu rodzajów połączeń pokazano na rysunku 16.7.

Oto lista najważniejszych cech technologii WiMAX:

- Pracuje na licencjonowanych częstotliwościach (należących do operatorów telekomunikacyjnych).
- Każda komórka pokrywa obszar o promieniu od 3 do 10 km.
- Wykorzystywane jest ortogonalne zwielokrotnianie częstotliwości.
- Gwarantowana jest określona jakość usługi (na przykład podczas transmisji glosu lub sekwencji wizyjnych).
- Przepływność na krótkich odległościach może osiągać 70 Mb/s.
- Przepustowość łączów długodystansowych wynosi 10 Mb/s (do 10 km).



Rysunek 16.7. Wykorzystanie technologii WiMAX w łączach szkieletowych i dostępowych  
Podsumując:

*Technologia WiMAX doskonale sprawdza się w bezprzewodowych sieciach MAN jako forma łącza szkieletowego, a także w charakterze stałego lub mobilnego łącza dostępowego. W przypadku zastosowania urządzeń WiMAX jako elementów łącz dostępowego nie jest konieczna bezpośrednią widoczność między nimi.*

### 16.13. Technologie i standardy sieci PAN

Organizacja IEEE przypisała standardom PAN numer 802.15. Każda z form wymiany danych w sieciach osobistych jest opracowywana przez osobną grupę roboczą lub konsorcjum firm. Najważniejsze rozwiązania z zakresu sieci PAN zostały przedstawione w tabeli 16.6.

Tabela 16.6. Standardy sieci PAN zdefiniowane przez organizację IEEE

Standard	Przeznaczenie
802.15.1a	Technologia Bluetooth (1 Mb/s; 2,4 GHz)
802.15.2	Współistnienie rozwiązań PAN (brak wzajemnych zakłóceń)
802.15.3	Wysokowydajne sieci PAN (55 Mb/s; 2,4 GHz)
802.15.3a	Ultraszerokopasmowe (UWB — ang. <i>Ultra Wide Band</i> ) sieci PAN (110 Mb/s; 2,4 GHz)
802.15.4	Technologia Zigbee — sieci o niskiej przepustowości stosowane w zdalnym sterowaniu
802.15.4a	Sieci o niskiej przepustowości i małym zapotrzebowaniu na energię

**Bluetooth.** Standard IEEE 802.15.1a powstał po opracowaniu przez producentów urządzeń technologii **Bluetooth** jako mechanizmu bezprzewodowego dostarczania danych na niewielkich odległościach. Do cech charakterystycznych rozwiązań Bluetooth trzeba zaliczyć:

- zastąpienie kabli połączeniami bezprzewodowymi (na przykład w zestawach słuchawkowych lub myszkach komputerowych);
- wykorzystanie pasma 2,4 GHz;
- realizację połączeń na krótkich odległościach (do 5 metrów; w niektórych odmianach standardu do 10 lub 50 metrów);
- podział na urządzenia **nadrzędne** (ang. *master*) i **podrzędne** (ang. *slave*);
- zarządzanie pracą urządzeń podrzędnych z urządzeń nadzorujących;
- przepustowość do 721 kb/s.

**Ultraszerokopasmowe** sieci PAN. Komunikacja UWB bazuje na założeniu rozproszenia strumieni danych na kilka częstotliwości, dzięki czemu potrzeba mniej energii do przekazania informacji na tę samą odległość. Oto lista najważniejszych cech technologii:

- wykorzystanie szerokiego widma częstotliwościowego;
- małe zapotrzebowanie na energię elektryczną;
- krótki zasięg (od 2 do 10 metrów);
- przenikanie przez przeszkody (na przykład przez mury);
- przepustowość 110 Mb/s przy odległości 10 m oraz 500 Mb/s przy odległości 2 m;
- niemożność zakończenia prac nad pojedynczym standardem w ramach organizacji IEEE.

**Zigbee.** Standard Zigbee (802.15.4) został opracowany jako zunifikowany mechanizm bezprzewodowego zdalnego sterowania, szczególnie na potrzeby urządzeń przemysłowych. Nie wymaga wysokich przepustowości, ponieważ zdalne sterowanie ogranicza się zazwyczaj jedynie do przesyłania krótkich poleceń. Oto jego cechy charakterystyczne:

- standard zdalnego sterowania, a nie bezprzewodowej transmisji danych;
- zastosowanie w zakładach przemysłowych oraz systemach automatyki domowej;
- wykorzystanie trzech pasm częstotliwościowych (868 MHz, 915 MHz i 2,4 GHz);
- przepustowość 20, 40 lub 250 kb/s w zależności od częstotliwości nadawczej;
- niskie zapotrzebowanie na energię elektryczną;
- trzy poziomy zabezpieczeń transmisji danych.

## 16.14. Inne technologie komunikacji na niedużych odległościach

Istnieje kilka dodatkowych technologii komunikacji na krótkich odległościach, które nie są klasyfikowane jako mechanizmy PAN. Przykładem mogą być rozwiązania związane z transmisją w podczerwieni — InfraRED. Odpowiadają one za zdalne sterowanie urządzeniami i wymianę danych z niewielkimi przepływnościami. Z kolei technologie z grupy RFID znajdują zastosowanie w obsłudze różnego rodzaju czujników.

**InfraRed.** Systemy InfraRED są wykorzystywane w zdalnym sterowaniu oraz w rozwiązańach, w których mogą zastępować kable (na przykład jako sposób przyłączania myszki komputerowej). Standardy komunikacji w podczerwieni zostały opracowane przez stowarzyszenie Infrared Data Association (IrDA) i są powszechnie wykorzystywane w praktyce. Oto najważniejsze cechy technologii IrDA:

- Specyfikacja obejmuje rozwiązania o różnych przepustowościach i różnym przeznaczeniu.
- Zasięg w rzeczywistych systemach wynosi od jednego do kilku metrów.
- Transmisja kierunkowa o kącie akceptacji 30°.
- Przepustowość w zakresie od 2,4 kb/s (sterowanie) do 16 Mb/s (wymiana danych).
- Niskie zapotrzebowanie na energię elektryczną.
- Sygnał odbija się od przeszkód, ale ich nie przenika.

**Identyfikacja na częstotliwościach radiowych** (RFID — ang. *Radio Frequency Identification*). Technologia RFID wykorzystuje interesującą formę komunikacji bezprzewodowej, w której informacje identyfikacyjne zapisane w **transponderach** mogą być pobierane przez odbiornik. Najważniejsze cechy rozwiązania to:

- Istnieje ponad 140 standardów RFID zależnych od konkretnych aplikacji.
- Pasywne układy RFID pobierają energię z sygnału emitowanego przez czytnik.
- Aktywne elementy RFID zawierają baterię, która wystarcza na 10 lat pracy.
- Ograniczona odległość (układy aktywne mają większy zasięg niż układy pasywne).
- Możliwość wykorzystania zarówno częstotliwości niższych niż 100 MHz, jak i częstotliwości z przedziału 868 – 954 MHz.
- Zastosowanie w kontroli stanów magazynowych, czujnikach, nośnikach danych uwierzytelniających itp.

## 16.15. Technologie bezprzewodowych sieci WAN

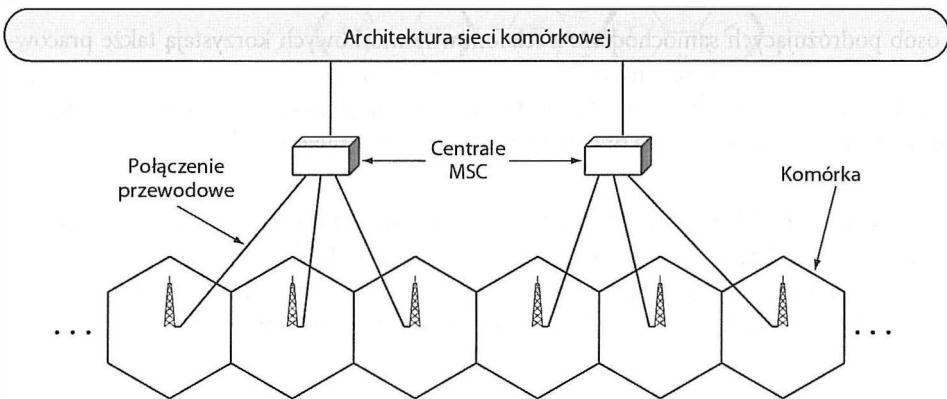
Technologie bezprzewodowych sieci WAN można podzielić na dwie kategorie:

- systemy komórkowe,
- systemy satelitarne.

### 16.15.1. Systemy komórkowe

Systemy komórkowe zostały opracowane przede wszystkim z myślą o realizacji połączeń głosowych z przemieszczającymi się użytkownikami. Z tego względu projektowano je w taki sposób, aby pojedyncze komórki miały połączenie z publiczną siecią telefoniczną. Obecnie coraz częściej systemy komórkowe są wykorzystywane do świadczenia usług transmisji danych i dostępu do internetu.

W ujęciu architektonicznym komórki są wyznaczane przez stacje bazowe, a pracą grupy komórek (najczęściej sąsiadujących ze sobą) zarządza **centrala sieci komórkowej** (MSC — ang. *Mobile Switching Center*). Centrala śledzi położenie użytkowników sieci i realizuje operacje związane z przekazywaniem kontroli nad urządzeniami z jednej komórki do kolejnej. Na rysunku 16.8 przedstawiono przykład rozmieszczenia komórek wzdłuż drogi.

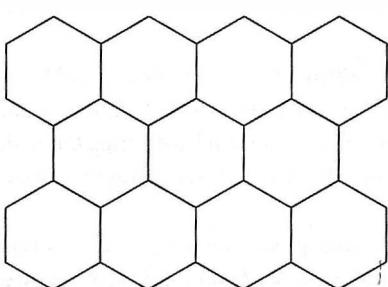


Rysunek 16.8. Przykład architektury sieci komórkowej

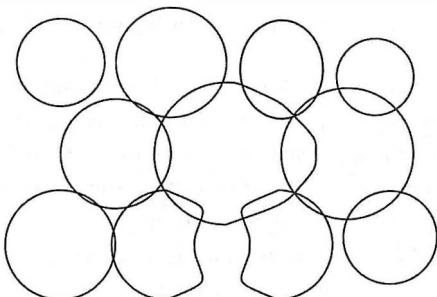
Jeśli użytkownik przemieszcza się między komórkami jednej centrali MSC, operacja przekazania urządzenia klienckiego jest realizowana wewnętrz danej centrali. Jeżeli jednak abonent sieci przemieszcza się między większymi regionami geograficznymi, w przełączanie zaangażowane są dwie centraly MSC.

Teoretycznie najefektywniejsze pokrycie terenu sygnałem radiowym uzyskuje się wówczas, gdy komórki mają formę sześcioboku foremnego. W praktyce jednak zasięg stacji bazowych nie jest idealny. Na większości masztów instaluje się anteny **dookólne**, czyli talkie, które swoim zasięgiem obejmują obszar koła. Kształt ten jest dodatkowo modyfikowany przez przeszkody oraz sygnały zakłócające, które mogą powodować tłumienie w niektórych rejonach geograficznych. W rezultacie część komórek się na siebie nakłada lub pozostają obszary nieobjęte zasięgiem żadnej stacji bazowej. Idealne i rzeczywiste pokrycie terenu sygnałem radiowym pokazano na rysunku 16.9.

Kolejny element technologii komórkowych, który trzeba uwzględnić, to różna gęstość komórek. Na obszarach rolniczych, na których spodziewana częstość występowania telefonów komórkowych jest niewielka, komórki mają duże rozmiary — pojedyncza stacja bazowa obsługuje duży obszar geograficzny. Z kolei w miejscowościach gęsto zaludnionych należy się spodziewać znacznie większej liczby telefonów komórkowych w takim samym obszarze. Wystarczy wyobrazić sobie dzielnicę w dużym mieście. Oprócz przechodniów



(a)



(b)

**Rysunek 16.9.** Idealna sieć komórkowa (a)  
oraz rzeczywiste pokrycie terenu z nałożeniami i lukami (b)

i osób podróżujących samochodami z telefonów komórkowych korzystają także pracownicy biur i mieszkający okolicznych domów. Dlatego zamiast idealnego podziału na komórki o jednakowych rozmiarach w praktyce stosuje się podział na komórki o różnych wielkościach (mniejsze są stosowane przede wszystkim w obszarach miejskich).

Choć sieć komórkowa jest często przedstawiana jako zbiór sześcioboków foremnych, kształty komórek w rzeczywistych systemach zależą od liczby telefonów i ewentualnych przeszkód terenowych, które wprowadzają nieregularność w kształtach i powodują nakładanie się obszarów lub występowanie przerw w pokryciu sygnałem.

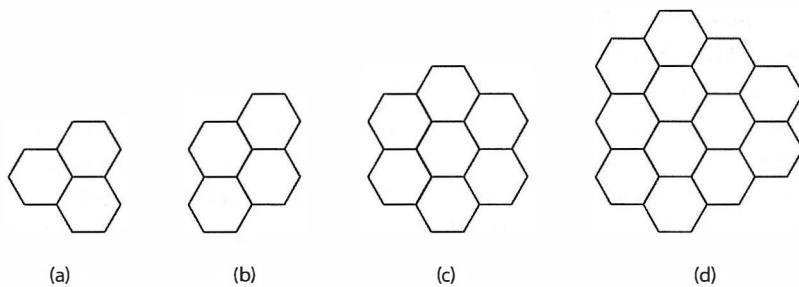
## 16.16. Klastry komórek i wielokrotne wykorzystywanie częstotliwości

Komunikacja w sieciach komórkowych bazuje na twierdzeniu, że:

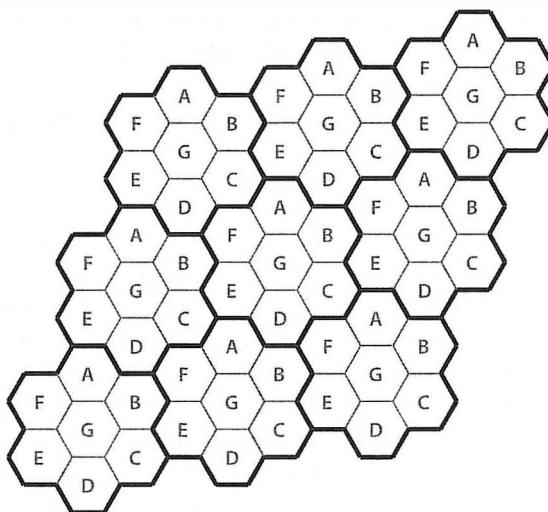
Aby zagwarantować niski poziom interferencji, sąsiednie komórki muszą korzystać z różnych częstotliwości.

Chcąc praktycznie wykorzystać powyższe twierdzenie, projektanci systemów komórkowych posługują się **klastrami**, które umożliwiają powielanie określonej kombinacji komórek. Na rysunku 16.10 przedstawiono klastry o rozmiarach 3, 4, 7 i 12 komórek (są to najczęściej stosowane rozmiary klastrów).

Z geometrii wynika, że zastosowanie któregokolwiek z przedstawionych kształtów pozwala na pokrycie dowolnie dużego obszaru bez żadnych luk. Ponadto, jeśli każdej komórce zostanie przypisana inna częstotliwość pracy, powielanie wzoru zagwarantuje, że sąsiadujące ze sobą komórki nigdy nie będą wykorzystywały tej samej częstotliwości. Na rysunku 16.11 został zaprezentowany przykład powielenia klastra składającego się z siedmiu komórek. Litera znajdująca się każdym z sześcioboków symbolizuje częstotliwość przypisaną danej komórce.



Rysunek 16.10. Typowe klastry komórek

Rysunek 16.11. Przykład przydziału częstotliwości w sieci,  
w której powielono klaster złożony z siedmiu komórek

Każda litera odpowiada jednej częstotliwości, a każda komórka klastra ma własną częstotliwość. Jak nietrudno zauważać, powielanie klastra gwarantuje, że znajdujące się obok siebie komórki mają różne częstotliwości.

## 16.17. Generacje technologii komórkowych

W przemyśle telekomunikacyjnym technologie komórkowe zostały podzielone na cztery generacje i oznaczone jako 1G, 2G, 3G i 4G. Wyróżniono również dwie wersje pośrednie — 2,5G i 3,5G. Oto ich cechy:

- **1G.** Systemy pierwszej generacji były dostępne od lat 70. do lat 80. ubiegłego wieku. Do przenoszenia głosu wykorzystywano sygnały analogowe, a same telefony miały charakter **przenośnych komórkowych radiotelefonów**.

**2G i 2,5G.** Początek drugiej generacji telefonii komórkowej datuje się na początki lat 90. Rozwiązań tego typu są wykorzystywane do dzisiaj. Główna różnica między

systemami 1G i 2G polega na zastąpieniu transmisji analogowej przekazem cyfrowym. Oznaczenie **2,5G** (lub **2+**) jest stosowane do oznaczania systemów 2G z elementami rozwiązań 3G.

- **3G i 3,5G.** Trzecia generacja systemów komórkowych jest dostępna od początku XXI wieku. Jej głównym przeznaczeniem jest realizacja usług wymagających większych przepustowości. Systemy 3G umożliwiają pobieranie danych z szybkością między 400 kb/s a 2 Mb/s. Znajdują więc zastosowanie w aplikacjach związanych z przeglądaniem stron WWW czy wymianą zdjęć. Rozwiązania 3G pozwalają na korzystanie z tych samych telefonów w Europie, Ameryce Północnej i Japonii.
- **4G.** Urządzenia czwartej generacji są produkowane od 2008 roku. Zapewniają one swoim użytkownikom transmisję w czasie rzeczywistym strumieni multimedialnych, takich jak przekazy telewizyjne lub pliki wideo. Dodatkowo telefony 4G zawierają komponenty innych technologii (na przykład Wi-Fi lub połączeń satelitarnych), a oprogramowanie urządzenia automatycznie wybiera najlepszy mechanizm transmisyjny spośród dostępnych w danej chwili.

Rozwój technologii komórkowych doprowadził do powstania wielu niezależnych rozwiązań i standardów. Wraz z rozpoczęciem prac nad systemami drugiej generacji różne grupy badawcze zaczęły proponować różne sposoby postrzegania telefonii komórkowej i przygotowywały różne standardy. W Europie organizacja o nazwie European Conference of Postal and Telecommunications Administrators wybrała technologię TDMA, stanowiącą podstawę **globalnego systemu komunikacji mobilnej** (GSM — ang. *Global System for Mobile Communications*), i opracowała system, który miał mieć ogólnoswiatowy charakter. Jednocześnie w Stanach Zjednoczonych każdy operator telekomunikacyjny wdrożył własne rozwiązania. Firma Motorola opracowała system TDMA znany pod nazwą **iDEN**. Większość amerykańskich i japońskich operatorów przyjęła rozwiązania CDMA, które zostały następnie znormalizowane i oznaczone jako **IS-95A**. W Japonii opracowano również system TDMA znany jako **PDC**. Wykaz najważniejszych standardów 2G oraz niektóre rozwiązania 2,5G są widoczne w tabeli 16.7. Inne rozwiązania, nieuwzględnione w zestawieniu, odegrały mniej istotną rolę w rozwoju tej gałęzi telekomunikacji.

Każdy z wymienionych standardów opisuje pewien mechanizm komunikacji, który umożliwia świadczenie wielu usług. Na przykład **ogólna usługa pakietowej transmisji radiowej** (GPRS — ang. *General Packet Radio Service*) jest dostępna dla abonentów sieci GSM i IS-136. Włączenie funkcji GPRS powoduje, że klient może uruchamiać różnego rodzaju usługi, które pracują w systemie GPRS. Do przekazywania krótkich komunikatów tekstowych służy usługa **SMS**, do realizacji połączeń internetowych można wykorzystać usługę **WAP**, a za przesyłanie wiadomości multimedialnych odpowiada usługa **MMS**. Operatorzy telekomunikacyjni zazwyczaj pobierają dodatkowe opłaty za korzystanie z usług GPRS, często zależne od ilości przesłanych danych (liczonej w megabajtach).

Po wprowadzeniu mechanizmów GPRS producenci zaczęli prace nad bardziej wyrafinowanymi technikami modulacji i multipleksacji, które pozwoliłyby na zwiększenie szybkości transmisyjnych. W wyniku tych działań powstał system **EDGE**, znany również jako **rozszerzony GPRS** (EGPRS — ang. *Enhanced GPRS*), który umożliwia przekazywanie danych z przepływnością 473,6 kb/s. Z kolei jego następcą — system **EDGE Evolution** — zapewnia transmisję z szybkością 1 Mb/s.

Tabela 16.7. Najważniejsze technologie komórkowe drugiej generacji

Technika	Standard	Generacja
GSM	GSM	2G
	GPRS	2,5G
	EDGE (EGPRS)	2,5G
	EDGE Evolution	2,5G
	HSCSD	2,5G
CDMA	IS-95A	2G
	IS-95B	2,5G
TDMA	iDEN	2G
	IS-136	2G
	PDC	2G

Jednocześnie operatorzy telekomunikacyjny zaczęli myśleć o trzeciej generacji telefonii komórkowej. Nie było bowiem wątpliwości, że użytkownicy telefonów chcieli, by ich urządzenia działały na całym świecie. Wywarli więc skuteczny nacisk na producentów urządzeń, którzy wybrali spośród rozwiązań 2G kilka kluczowych standardów (IS-136, PDC, IS-95A oraz EDGE) i opracowali technologię UMTS, wykorzystującą szerokopasmową technikę CDMA (WCDMA — ang. *Wideband CDMA*). Na bazie rozwiązania IS-95B powstał natomiast system **CDMA 2000**. Dalszy rozwój telefonii komórkowej obrazuje zestawienie w tabeli 16.8.

Tabela 16.8. Technologie komórkowe trzeciej generacji

Technika	Standard	Poprzednik
WCDMA	UMTS	IS-136, IS-95A, EDGE, PDC
	HSDPA	UMTS
CDMA 2000	1xRTT	IS-95B
	EVDO	1xRTT
	EVDV	1xRTT

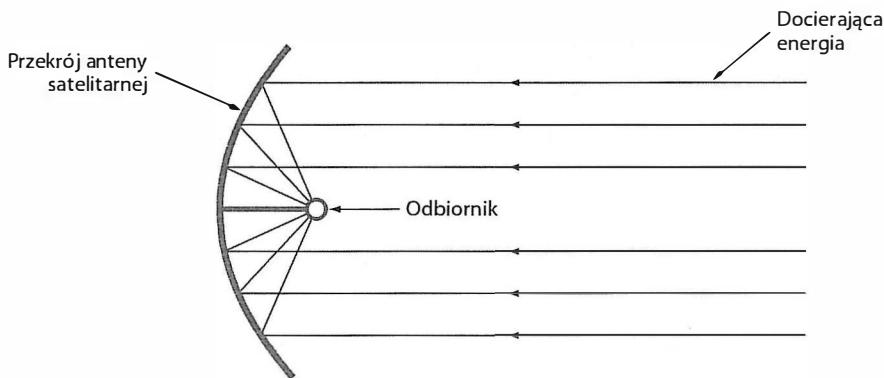
Ostatecznie w charakterze usług transmisji danych trzeciej generacji wykorzystuje się kilka konkurencyjnych względem siebie rozwiązań. Technologie EVDO i EVDV powstały niemal jednocześnie. Każda z nich łączy w sobie techniki CDMA i FDM, które razem zwiększą ogólną wydajność systemu. Rozwiązanie EVDO zostało wdrożone na większą skalę. Samo występuje jednak w dwóch wersjach, które różnią się szybkością transmisji danych (2,4 Mb/s i 3,1 Mb/s). Alternatywna technologia, o nazwie HSDPA, zapewnia

pobieranie informacji z szybkością 14 Mb/s<sup>44</sup>. Oczywiście, większa przepustowość jest związana z większymi opłatami naliczanymi przez operatorów telekomunikacyjnych.

## 16.18. Technologia satelitarna VSAT

W rozdziale 7. zostały opisane trzy rodzaje systemów satelitarnych (LEO, MEO i GEO). Z kolei w rozdziale 14. omówiono różne mechanizmy zarządzania dostępem do kanału komunikacyjnego, w tym technikę rezerwacji, która jest stosowana w łączności satelitarnej zgodnej z mechanizmem TDMA. Ten podrozdział stanowi pewne podsumowanie wcześniejszych rozważań, ponieważ zawiera opisy konkretnych technologii satelitarnych.

Najważniejszym elementem komunikacji satelitarnej jest antena paraboliczna nazywana **talerzem**. Paraboliczny kształt oznacza, że energia elektromagnetyczna pochodząca z satelity jest koncentrowana w jednym punkcie. Skierowanie talerza w stronę satelity i umieszczenie odbiornika w ognisku anteny gwarantuje odbiór sygnału o wysokiej jakości. Na rysunku 16.12 przedstawiono przekrój anteny satelitarnej. Zaprezentowano także zasadę odbijania się sygnału od powierzchni talerza i koncentrowania energii w odbiorniku.



Rysunek 16.12. Odbicie sygnału od talerza anteny parabolicznej

W początkowej fazie rozwoju systemów komunikacji satelitarnej w celu zwiększenia poziomu odbieranego sygnału budowano stacje naziemne wyposażone w anteny, których średnica przekraczała często trzy metry. Rozwiązań tego typu były akceptowalne w przypadku łączów transatlantyckich zarządzanych przez firmy telekomunikacyjne. Jednak odbiorcy indywidualni i niewielkie firmy nie mogli sobie pozwolić na umieszczanie stacji naziemnych na terenie posesji. Istotna zmiana nastąpiła wraz z opracowaniem nowej technologii, która charakteryzowała się **bardzo małymi rozmiarami terminala** (VSAT — ang. *Very Small Aperture Terminal*). Średnica typowej anteny VSAT nie przekracza bowiem metra.

<sup>44</sup> Opracowano również technologię HSUPA, która zapewnia większą przepustowość podczas wysyłania danych, ale nie cieszyła się ona dużym zainteresowaniem.

Wiele firm wykorzystuje technologię VSAT do łączenia swoich sklepów. W Stanach Zjednoczonych z rozwiązań tego korzystają sieci aptek Walgreens i CVS, restauracje szybkiej obsługi Pizza Hut i Taco Bell, a także sieci handlowe takie jak Wal Mart. Usługi VSAT zapewniają użytkownikom dostęp zarówno do internetu, jak i przekazów telewizyjnych.

W połączeniach satelitarnych VSAT wykorzystuje się trzy zakresy częstotliwościowe, z których wynikają różne moce sygnałów odbiorczych, różna podatność na zakłócenia wywołane deszczem lub innymi zjawiskami atmosferycznymi, a także różny obszar pokrycia sygnałem powierzchni Ziemi. Cechy charakterystyczne każdego pasma zestawiono w tabeli 16.9.

Tabela 16.9. Zakresy częstotliwościowe technologii VSAT i ich cechy charakterystyczne

Pasmo	Częstotliwości	Pokrycie	Siła sygnału	Wpływ deszczu
C	3 – 7 GHz	Duże	Mała	Średni
Ku	10 – 18 GHz	Średnie	Średnia	Umiarkowany
Ka	18 – 31 GHz	Małe	Duża	Istotny

## 16.19. Satelity GPS

Systemy **globalnego systemu pozycjonowania** (GPS — ang. *Global Positioning System*) dostarczają precyzyjnych informacji na temat czasu i położenia. Choć nie są elementami komputerowych systemów komunikacji, dane na temat lokalizacji są coraz częściej wykorzystywane w transmisji danych realizowanej przez urządzenia mobilne. Do najważniejszych cech systemu należy zaliczyć:

- Dokładność pomiaru w przedziale od 20 do 2 metrów (wersje wojskowe zapewniają większą precyzję).
- Wykorzystanie 24 satelitów na orbicie Ziemi.
- Rozmieszczenie satelitów na sześciu płaszczyznach orbitalnych.
- Dostarczanie informacji synchronizacyjnych, wykorzystywanych w niektórych sieciach transmisji danych.

Zasada ustalania położenia obiektu na Ziemi nie jest szczególnie skomplikowana. Wszystkie satelity GPS okrążają Ziemię po znanych orbitach. Odbiornik może więc określić swoją pozycję, obliczając odległość do satelity. W jaki sposób? Wyobraźmy sobie zbiór punktów oddalonych od satelity 1 na odległość  $D_1$ . Zbiór ten wyznacza sferę. Wyobraźmy sobie analogiczną sferę wyznaczoną przez punkty oddalone na odległość  $D_2$  od satelity 2. Obiekt, który jest odległy o  $D_1$  od jednego satelity i  $D_2$  od drugiego, znajduje się na okręgu wyznaczonym przez przecięcie sfery. Jeśli jednocześnie jest oddalony na odległość  $D_3$  od trzeciego satelity, musi się znajdować w miejscu przecięcia trzeciej sfery z wyznaczonym wcześniej okręgiem. To założenie spełniają jedynie dwa punkty. Jednak tylko jeden z tych punktów leży na powierzchni Ziemi, a drugi odpowiada jakiemuś punktowi w przestrzeni kosmicznej. Wybranie właściwego punktu nie stanowi więc problemu.

Aby obliczyć odległość, odbiornik GPS wykorzystuje zasady fizyki newtonowskiej, zgodnie z którymi jest ona równa prędkości pomnożonej przez czas. Prędkość jest stała (prędkość rozchodzenia się światła wynosi około  $3 \cdot 10^8$  m/s). W celu obliczenia czasu każdy odbiornik GPS wyznacza czas lokalny, a każdy satelita uwzględnia w przesyłanych danych **znacznik czasu** pobierany z własnego bardzo dokładnego zegara. Odbiornik może odjąć znacznik czasu od czasu lokalnego i na tej podstawie określić czas, w którym informacja była transmitowana.

## 16.20. Radio programowe i przyszłość technologii bezprzewodowych

We wszystkich opisanych w tym rozdziale technologiach transmisji bezprzewodowej niezbędne jest wykorzystanie specjalnie zaprojektowanych urządzeń radiowych. Antena, nadajnik i odbiornik danego urządzenia są przystosowane do pracy na określonych częstotliwościach z określona modulacją i techniką zwielokrotniania. Nowoczesne telefony komórkowe mogą korzystać z sieci GSM, Wi-Fi i CDMA. Muszą jednak zawierać w swojej budowie trzy niezależne systemy radiowe, które są w odpowiednich momentach włączane i wyłączone.

Tradycyjne moduły radiowe zostaną w najbliższej przyszłości zastąpione przez komponenty **programowe**, których działanie będzie zależne od programu wykonywanego przez procesor. W tabeli 16.10 wymieniono najważniejsze aspekty transmisji radiowej, które mogą pozostać pod kontrolą **programowego radia** (ang. *software radio*).

**Tabela 16.10.** Cechy transmisji radiowej definiowane przez radio programowe

Cecha transmisji	Opis
Częstotliwość	Dokładny zbiór częstotliwości wykorzystywanych w danym czasie
Moc	Ilość energii emitowanej przez nadajnik
Modulacja	Kodowanie sygnału i kodowanie kanałowe oraz modulacja
Zwielokrotnianie	Połączenie technik CDMA, TDMA, FDMA i innych
Kierunek rozchodzenia się sygnału	Anteny można dostrajać do wybranych kierunków propagacji sygnału
Protokół MAC	Wszystkie operacje związane z ramkowaniem i wyznaczeniem adresów

Działanie radia programowego jest możliwe dzięki kilku innym rozwiązaniom: wstrajanym filtrom analogowym oraz mechanizmowi zarządzania wieloma antenami. Wspomniane filtry są dostępne na rynku w formie analogowych układów scalonych. Pozwalały one na wybranie odpowiedniej częstotliwości i mocy sygnału. Kodowanie sygnałowe i modulacja należą z kolei do zadań **cyfrowych procesorów sygnałowych** (DSP — ang. *Digital Signal Processors*). Najciekawszym aspektem funkcjonowania programowych

modułów radiowych jest zarządzanie wieloma antenami. Działanie modułu nie ogranicza się jedynie do wybierania jednej z anten w danym czasie. Możliwe jest natomiast korzystanie z wielu anten jednocześnie, co pozwala na **multipleksację przestrzenną**, czyli odbieranie sygnału z określonego kierunku lub nadawanie go w określonym kierunku. Systemy wykorzystujące wiele anten do nadawania i odbierania są oznaczane skrótem **MIMO** (ang. *Multiple-Input Multiple-Output*), oznaczającym wiele wejść i wiele wyjść.

Radia programowe przeszły fazę badań i są wdrażane przez armię amerykańską. Jednocześnie trwają prace w ramach projektów **Universal Software Radio Peripheral** oraz **GNU Radio** nad „cywilną” wersją technologii. Zanim opisywany system pojawi się w formie produktów komercyjnych, trzeba rozwiązać kilka problemów. Po pierwsze, konieczne jest obniżenie kosztu (dzisiaj wynosi on około 1000 dolarów). Po drugie, musi zostać określona polityka wykorzystania widma częstotliwościowego. Urządzenia emitujące energię elektromagnetyczną podlegają certyfikowaniu, które gwarantuje, że nie zakłócają innych środków komunikacyjnych (na przykład telefony komórkowe nie zakłócają częstotliwości policyjnych lub służb ratowniczych). Możliwość reprogramowania takiego radia oznacza ryzyko pobrania wirusa, który mógłby generować zakłócenia na kanałach przeznaczonych dla służb specjalnych. Cały czas trwają więc prace nad ograniczeniem mocy nadawczej radia na określonych częstotliwościach.

## 16.21. Podsumowanie

Istnieje wiele technologii komunikacji bezprzewodowej, które są wykorzystywane w budowie sieci LAN, PAN, MAN i WAN. Organizacja IEEE opracowała standardy kilku rozwiązań przeznaczonych dla sieci LAN i MAN. W systemach Wi-Fi wykorzystuje się specyfikację IEEE 802.11 wraz z różnymi jej odmianami, oznaczanymi za pomocą sufiksów takich jak 802.11b lub 802.11g. Bezprzewodowe sieci LAN można tworzyć w trybie ad hoc lub infrastrukturalnym (z wykorzystaniem punktów dostępowych). Ramka połączenia bezprzewodowego przechowuje zarówno adres MAC punktu dostępowego, jak i adres MAC routera, do którego punkt dostępowy jest podłączony.

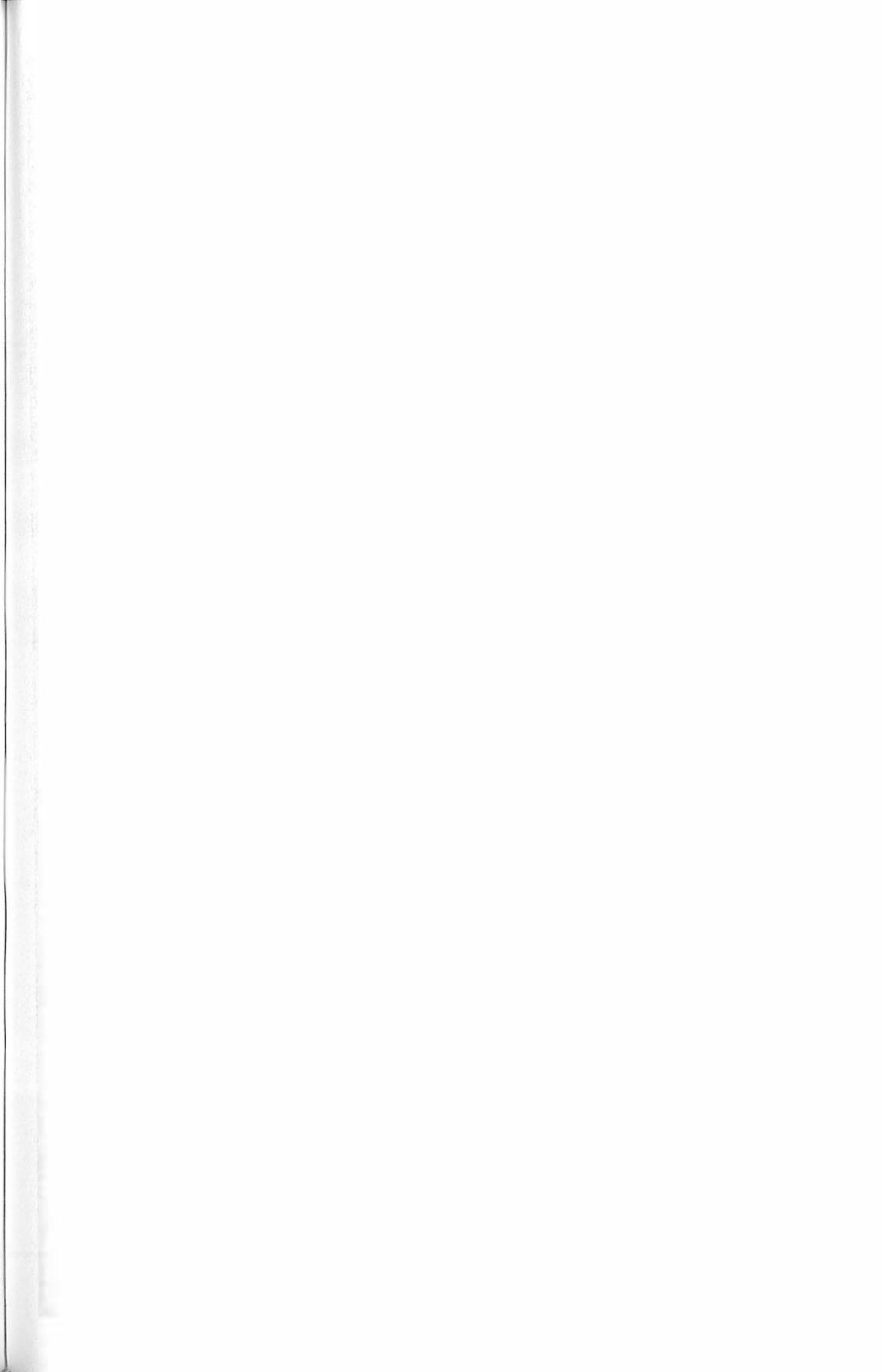
Technologie bezprzewodowe nie są wykorzystywane jedynie w sieciach LAN. Znajdują zastosowanie również w systemach MAN i PAN. Najważniejszym rozwiązaniem tego typu w sieciach MAN jest WiMAX. Łączy WiMAX można używać jako łączy szkieletowych lub dostępowych. Wśród rozwiązań PAN zauważalna jest znacznie większa różnorodność. Do najważniejszych technologii należy tutaj zaliczyć Bluetooth, UWB, Zigbee oraz IrDA. Inny wariant komunikacji bezprzewodowej zawiera specyfikację RFID, która opisuje systemy ułatwiające na przykład inwentaryzację i spedycję.

Bezprzewodowe sieci WAN bazują na rozwiązaniach komórkowych i satelitarnych. Technologie komórkowe są zaliczane do grupy 1G (połączenia analogowe), 2G (cyfrowe przekazywanie głosu), 3G (cyfrowe przekazywanie głosu i danych) lub 4G (wysokowydajne połączenia głosowe i transmisji danych). System VSAT umożliwia wykorzystanie łącz satelitarnych w niewielkich firmach i przez odbiorców indywidualnych.

Najnowsze systemy łączności bezprzewodowej są opracowywane w formie programowych modułów radiowych, które nadzorują różne aspekty transmisji radiowej. W czasie pisania książki radia programowe były bardzo kosztowne i przeznaczone jedynie do zastosowań wojskowych.

## ZADANIA

- 16.1. Wymień trzy technologie PAN i krótko je scharakteryzuj.
- 16.2. Wymień trzy zakresy częstotliwości wykorzystywanych w rozwiązaniach LAN i PAN.
- 16.3. Jaki jest cel stowarzyszenia Wi-Fi?
- 16.4. Podaj prefiks liczbowy opracowanych przez organizację IEEE standardów sieci Wi-Fi.
- 16.5. Wymień trzy techniki rozpraszania widma i opisz każdą z nich.
- 16.6. Wyszukaj w internecie informacje o technice OFDM i przygotuj jej jednoakapitowy opis.
- 16.7. Wymień i scharakteryzuj standardy IEEE przeznaczone do stosowania w sieciach LAN.
- 16.8. Dlaczego w większości bezprzewodowych sieci LAN stosuje się pracę w trybie infrastrukturalnym, a nie ad hoc?
- 16.9. Dlaczego komputer musi się stowarzyszać z wybranym punktem dostępowym?
- 16.10. Nagłówek ramki 802.11 zawiera dwa pola adresów docelowych. Opisz przeznaczenie każdego z nich.
- 16.11. Do czego służą parametry SIFS i DIFS?
- 16.12. Wymień dwie odmiany technologii WiMAX i opisz ich przeznaczenie.
- 16.13. Czym jest Zigbee i jakie ma zastosowanie?
- 16.14. Wymień cechy charakterystyczne technologii UWB.
- 16.15. Czy połączenia IrDA nadają się do transmisji plików? Uzasadnij odpowiedź.
- 16.16. Czym jest RFID i jakie ma zastosowanie?
- 16.17. Do czego jest przyłączona stacja bazowa telefonii komórkowej?
- 16.18. Co to jest klaster komórek? Do czego służą klastry komórek?
- 16.19. Wymień cztery generacje technologii komórkowych i opisz każdą z nich.
- 16.20. Czym jest GSM i jakie standardy obejmuje?
- 16.21. Wymień technologie komórkowe trzeciej generacji, które wykorzystują zwielokrotnianie kodowe.
- 16.22. Czym jest system VSAT?
- 16.23. Dlaczego antena satelitarna ma kształt paraboliczny?
- 16.24. Wymień trzy pasma częstotliwościowe wykorzystywane w komunikacji satelitarnej i określ wpływ pogody na każde z nich.
- 16.25. Ile satelitów pracuje w systemie GPS i jaka jest dokładność lokalizacji w systemie GPS?
- 16.26. Jakich danych dostarcza system GPS poza informacjami o położeniu?
- 16.27. Jakie parametry łączności są wyznaczane przez radio programowe?



# Zawartość rozdziału

- 17.1. Wprowadzenie 313
- 17.2. Budowa sieci LAN i ograniczenia w jej zasięgu 313
- 17.3. Modemy optyczne 314
- 17.4. Regeneratorы 315
- 17.5. Mosty 315
- 17.6. Filtrowanie ramek 316
- 17.7. Dlaczego warto używać mostów? 317
- 17.8. Rozproszone drzewo rozpinające 318
- 17.9. Przełączanie i przełączniki warstwy 2. 319
- 17.10. Przełączniki sieci VLAN 321
- 17.11. Funkcje mostu w innych urządzeniach 322
- 17.12. Podsumowanie 322

# *Rozszerzenie sieci LAN — modemy optyczne, regeneratorы, mosty i przełączniki*

## **17.1. Wprowadzenie**

W poprzednich rozdziałach opisane zostały topologie sieci LAN oraz sposoby układania okablowania. Typowa sieć LAN rozciąga się na długości około kilkuset metrów, co oznacza, że może objąć swym zasięgiem budynek lub niewielki kampus.

W tym rozdziale przedstawiono dwie koncepcje, bardzo istotne w projektowaniu sieci — mechanizmy umożliwiające zwiększenie zasięgu sieci LAN oraz przełączanie. Omówione zostały również regeneratorы, mosty oraz algorytm drzewa rozpinającego, który eliminuje pętle w sieci.

## **17.2. Budowa sieci LAN i ograniczenia w jej zasięgu**

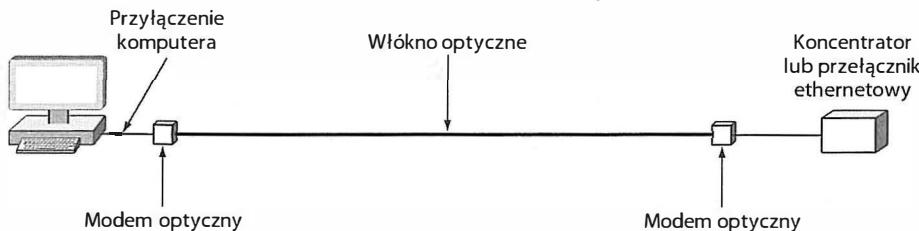
Ograniczenie zasięgu łączy jest fundamentalnym elementem każdego projektu sieci LAN. Opracowując nową technologię sieciową, inżynierowie dążą do uzyskania pewnej pojemności, maksymalnej wartości opóźnień i zasięgu przy określonym koszcie. Ograniczenie odległości między urządzeniami wynika z tego, że każda jednostka sieciowa jest przystosowana do emitowania określonej ilości energii. Nadmierne zwiększenie odległości powoduje więc, że stacja zdalna nie odbiera dostatecznie silnego sygnału i zaczynają się pojawiać błędy w transmisji.

*Maksymalna długość łącza jest jednym z najważniejszych parametrów technologii LAN. Urządzenia sieci LAN nie mogą działać poprawnie, gdy łączące je kable są zbyt długie.*

### 17.3. Modemy optyczne

Producenci urządzeń sieciowych opracowali wiele sposobów na zwiększenie zasięgu sieci LAN. Ogólnie jednak rozwiązania te nie sprowadzają się jedynie do zwiększenia siły sygnału i wydłużenia kabli. Większość wymaga zastosowania specjalnych interfejsów sprzętowych i uzupełnienia sieci o dodatkowe komponenty, które zapewnią przekazywanie sygnałów na większe odległości.

Najprostszy sposób rozszerzenia sieci polega na użyciu włókna światłowodowego i pary **modemów optycznych**. Rozwiązanie to pozwala na przyłączenie komputera do zdalnej sieci Ethernet, tak jak to zostało pokazane na rysunku 17.1.



Rysunek 17.1. Przyłączenie komputera do zdalnej sieci za pomocą modemów optycznych

Każdy z modemów optycznych zawiera moduły sprzętowe, które realizują dwa zadania — pobierają pakiety z interfejsu ethernetowego i wysyłają je za pośrednictwem włókna optycznego oraz pobierają pakiety docierające światłowodem i przekazują je do interfejsu ethernetowego<sup>45</sup>. Dzięki temu, że obydwa modemy udostępniają standardowe porty sieci LAN, zarówno w komputerze, jak i w zdalnym urządzeniu sieciowym można wykorzystać typowe karty sieciowe.

Podsumowując:

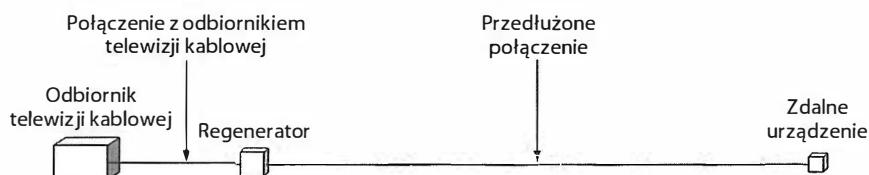
*Połączenie między komputerem i zdalną siecią LAN (taką jak Ethernet) można zrealizować za pomocą dwóch modemów optycznych i włókna światłowodowego.*

<sup>45</sup> W praktycznych implementacjach zazwyczaj instaluje się dwa włókna światłowodowe zapewniające jednoczesną transmisję w obydwu kierunkach.

## 17.4. Regeneratory

**Regenerator** jest urządzeniem analogowym przeznaczonym do przesyłania sygnałów sieci LAN na dużych odległościach. Jego zadanie nie polega na interpretowaniu treści pakietu lub choćby kodowania sygnałowego. Sprowadza się natomiast do wzmacniania odebranego sygnału i wysłania w odtworzonej postaci.

Regeneratory były często stosowane w pierwszych wersjach Ethernetu. Znajdują również zastosowanie w innych technologiach sieci LAN. Ostatnio producenci urządzeń wprowadzili na rynek regeneratory z odbiornikami podczerwieni. Dzięki temu odbiornik promieniowania podczerwonego może być oddalony od komputera. Rozwiązanie takie mogłoby zostać wykorzystane na przykład w sytuacji, w której pilot do odbiornika telewizji kablowej jest w innym pokoju niż sam odbiornik. Regenerator mógłby wówczas przedłużyć połączenie, tak jak to zostało pokazane na rysunku 17.2.



Rysunek 17.2. Przedłużenie połączenia z odbiornikiem podczerwieni

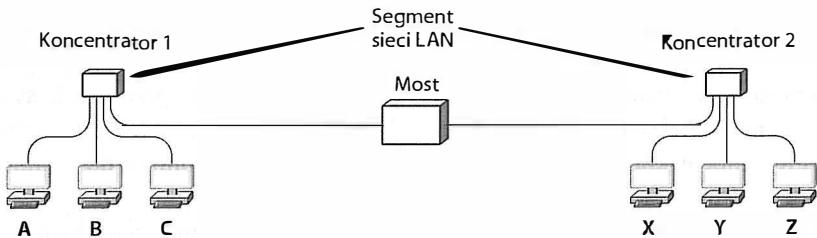
Podsumowując:

**Regenerator** jest analogowym urządzeniem, które zwiększa zasięg sieci LAN. Jego zadanie polega na wzmacnianiu odbieranych sygnałów i wysyłaniu ich do kolejnego urządzenia.

## 17.5. Mosty

**Most** jest urządzeniem, które łączy dwa segmenty sieci LAN (na przykład dwa koncentratory) i przekazuje pakiety pomiędzy nimi. W każdym segmencie nasłuchuje danych w **trybie zbiorczym** (ang. *promiscuous mode*). Oznacza to, że odbiera wszystkie pakiety przekazywane w danym segmencie. Po odebraniu ramki w jednym segmencie sieci przekazuje kopię ramki do drugiego segmentu. Dwa segmenty sieci połączone mostem działają więc tak, jakby stanowiły jedną sieć LAN — komputer przyłączony do jednego segmentu może wysyłać ramki do dowolnych komputerów z obydwu segmentów. Ramki rozgłoszeniowe są zawsze dostarczane do wszystkich komputerów w obydwu segmentach. Komputery nie muszą wiedzieć, czy są przyłączone do wspólnego segmentu sieci LAN, czy są rozdzielone mostem.

Początkowo mosty były sprzedawane jako oddzielne urządzenia o dwóch portach sieciowych. Obecnie stanowią element składowy innych urządzeń, takich jak modemy kablowe. Budowa mostu została przedstawiona na rysunku 17.3.



Rysunek 17.3. Sześć komputerów przyłączonych do dwóch segmentów sieci LAN połączonych mostem

Podsumowując:

*Most jest urządzeniem łączącym dwa segmenty sieci LAN. Jego zadanie polega na przekazywaniu ramek z jednego segmentu do drugiego. Komputery nie muszą wieć, czy są przyłączone do wspólnego segmentu, czy są rozdzielone mostem.*

## 17.6. Filtrowanie ramek

Praca mostu nie polega na bezwarunkowym przekazywaniu kopii każdej ramki z jednego segmentu sieci LAN do drugiego. Głównym jego zadaniem jest bowiem **filtrowanie** ramek na podstawie zawartych w nich adresów MAC. Most analizuje docelowe adresy i nie przekazuje ramek do drugiego segmentu, jeśli nie jest to konieczne. Oczywiście, jeśli w sieci dopuszczalne jest stosowanie rozgłoszeń lub multiemisji, urządzenie musi przekazywać kopie stosownych ramek między segmentami tak, jakby sieć składała się z pojedynczego segmentu LAN.

Skąd most wie, które komputery znajdują się w danym segmencie? Większość urządzeń tego typu to **mosty adaptacyjne** (**mosty uczące się**), które automatycznie zapamiętują lokalizację komputerów na podstawie adresów źródłowych ramek. Gdy most odbierze ramkę w jednym z segmentów, odczytuje zapisany w nagłówku adres nadawcy i dodaje go do listy jednostek przyłączonych do danego segmentu. Następnie odczytuje adres docelowy i na jego podstawie podejmuje decyzję o tym, czy przekazać ramkę do drugiego segmentu. Most zapamiętuje więc, że komputer znajduje się w określonym segmencie, gdy otrzyma ramkę z tego komputera.

Przeanalizujmy działanie opisanego mechanizmu na przykładzie z rysunku 17.3. W tabeli 17.1 przedstawiono sekwencję wymienianych pakietów, gromadzone przez most informacje o lokalizacji stacji oraz sposób przetwarzania pakietu (tj. wybór segmentu dla pakietu).

Działanie mostu można podsumować w następujący sposób:

*Most adaptacyjny wykorzystuje źródłowe adresy MAC ramek do zapamiętania lokalizacji nadawcy. Z kolei na podstawie adresów docelowych podejmuje decyzje o przekazaniu ramki do odpowiedniego segmentu sieci.*

**Tabela 17.1.** Proces uczenia się mostu w sieci złożonej z dwóch segmentów (komputer A, B i C pracują w jednym segmencie, a komputery X, Y i Z w drugim)

Zdarzenie	Segment 1	Segment 2	Emisja ramki
Uruchomienie mostu	-	-	-
Transmisja z A do B	A	-	Obydwa segmenty
Transmisja z B do A	A, B	-	Tylko segment 1
Rozgłoszenie z X	A, B	X	Obydwa segmenty
Transmisja z Y do A	A, B	X, Y	Obydwa segmenty
Transmisja z Y do X	A, B	X, Y	Tylko segment 2
Transmisja z X do Z	A, B	X, Y	Obydwa segmenty
Transmisja z Z do X	A, B	X, Y, Z	Tylko segment 2

## 17.7. Dlaczego warto używać mostów?

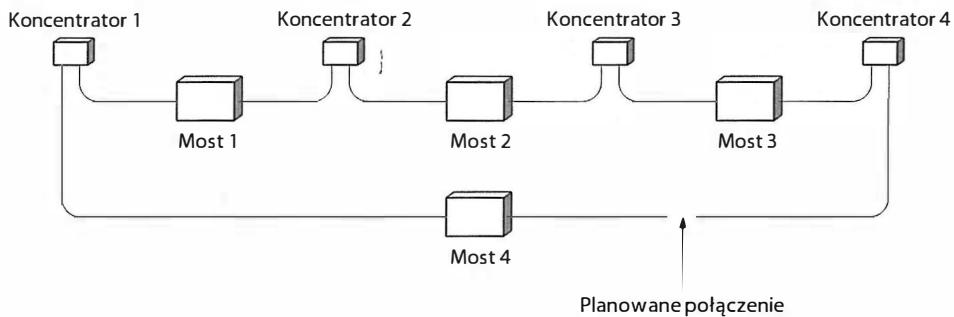
Gdy most zapamięta położenie wszystkich komputerów sieci, wydajność takiej sieci znacznie przekracza wydajność pojedynczego segmentu LAN. Wynika to z tego, że most umożliwia jednoczesną transmisję danych w każdym segmencie. Odnosząc się do rysunku 17.3, można stwierdzić, że nic nie stoi na przeszkodzie, aby komputer A wysłał pakiet do komputera C w tym samym czasie, gdy komputer X wysyła pakiet do komputera Y. Most otrzymuje obydwa pakiety, ale nie przenosi ich do kolejnych segmentów, ponieważ każdy z pakietów jest adresowany do jednostki znajdującej się w tym samym segmencie co nadawca. Zadanie mostu ogranicza się więc jedynie do odrzucenia dwóch ramek bez dalszego przetwarzania.

*Dzięki temu, że most pracuje jednocześnie w dwóch segmentach, wymiana danych między jednostkami jednego segmentu może zachodzić w tym samym czasie, gdy trwa wymiana danych między jednostkami drugiego segmentu.*

Zdolność mostu do lokalizowania transmisji pozwala na separowanie budynków sieci kampusowej. Większość przypadków wymiany danych ma charakter lokalny (komputer znacznie częściej komunikuje się z drukarką znajdującej się w tym samym budynku niż z drukarką zainstalowaną w innym budynku). Dzięki mostowi transmisja danych między budynkami zachodzi tylko wtedy, gdy jest to konieczne. Funkcja separowania segmentów jest również zaimplementowana w modemach, które pełnią funkcję mostu między siecią lokalną a siecią dostawcy usług internetowych.

## 17.8. Rozproszone drzewo rozpinające

Przeanalizujmy konfigurację sieci przedstawioną na rysunku 17.4. Cztery segmenty LAN są połączone za pomocą trzech mostów. Administrator decyduje się na zainstalowanie czwartego mostu. Zakładamy, że komputery (niepokazane na rysunku) są przyłączone do każdego z koncentratorów.



Rysunek 17.4. Sieć, do której dodawany jest czwarty most

Przed przyłączeniem czwartego mostu sieć działa zgodnie z przewidywaniami — każdy komputer może przekazywać ramki do dowolnie wybranej jednostki, może również generować rozgłoszenia dostarczane do wszystkich stacji. Rozgłoszenia i multiemisja są poprawnie obsługiwane dzięki temu, że most zawsze przekazuje ramki, które zawierają adres rozgłoszeniowy lub multiemisji. Dołączenie czwartego urządzenia okazuje się problemem, ponieważ prowadzi do powstania pętli. Jeśli choć jeden z mostów nie przestanie przekazywać rozgłoszeń, kopie ramek rozgłoszeniowych będą krążły w sieci w nieskończoność. Powielane nieustannie ramki będą również dostarczane do wszystkich komputerów.

Aby zapobiec niekończącemu się krążeniu ramek w sieci, w mostach implementuje się algorytm wyznaczający **rozproszone drzewo rozpinające** (DST — ang. *Distributed Spanning Tree*). W działaniu algorytmu mosty są przedstawiane jako węzły grafu, a sam graf ma formę drzewa (drzewo jest grafem pozbawionym pętli). Mechanizm został opracowany przez firmę Digital Equipment Corporation w 1985 roku z przeznaczeniem do wykorzystania w sieci Ethernet. Obecnie jest znany pod nazwą **protokołu drzewa rozpinającego** (STP — ang. *Spanning Tree Protocol*). Działanie protokołu składa się z trzech faz:

- wyboru mostu głównego,
- obliczenia najkrótszych tras,
- przekazywania pakietów.

Aby sformować drzewo, mosty ethernetowe komunikują się ze sobą, wykorzystując adres multiemisji zarezerwowany dla protokołu STP<sup>46</sup>:

01:80:C2:00:00:00

<sup>46</sup> Zgodnie z obowiązującą konwencją adresy ethernetowe są zapisywane w formacie szesnastkowym, a każda para cyfr jest oddzielana od następnej za pomocą dwukropka.

Pierwsza faza polega na wyborze mostu głównego. Operacja nie jest szczególnie skomplikowana. Wszystkie mosty wysyłają pakiety zawierające ich identyfikator (**identyfikator mostu**). Most o najmniejszej wartości identyfikatora staje się mostem głównym. Aby umożliwić administratorowi sieci wpływanie na proces elekcji, w identyfikatorze zawarto 16-bitową (ustawialną) wartość **priorytetu** oraz 48-bitowy adres MAC. Porównując identyfikatory, mosty w pierwszej kolejności sprawdzają priorytety, a następnie adresy MAC. Administrator może wyznaczyć most główny, ustawiając niższą wartość priorytetu niż w innych urządzeniach.

Drugim etapem jest obliczanie najkrótszych tras w sieci. Każdy most wyznacza najkrótszą trasę do mostu głównego. W rezultacie drzewo obejmuje wszystkie mosty drzewa rozpinającego.

Po wyznaczeniu drzewa rozpinającego mosty rozpoczynają przekazywanie pakietów. Interfejsy, przez które przebiega najkrótsza trasa do mostu głównego, są włączone, a interfejsy znajdujące się poza tymi trasami pozostają zablokowane (żadne pakiety użytkowe nie są wówczas przekazywane przez te interfejsy).

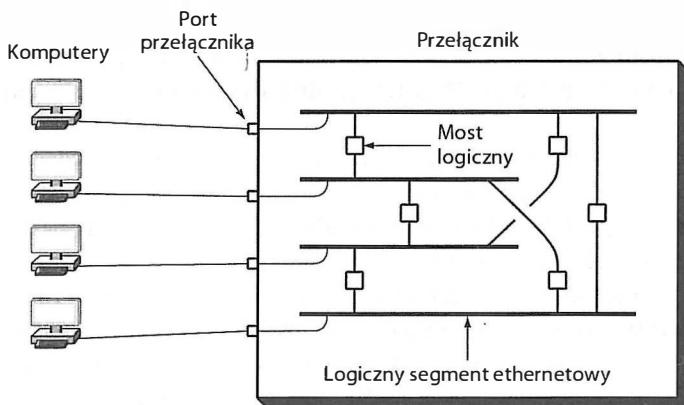
Istnieje kilka odmian protokołów drzewa rozpinającego. W 1990 roku organizacja IEEE opracowała standard o nazwie 802.1d, który został uaktualniony w 1998 roku. Ta sama organizacja przygotowała specyfikację 802.1q, która opisuje działanie algorytmu drzewa rozpinającego w niezależnych od siebie sieciach logicznych współdzielących jedno medium transmisyjne (bez ryzyka pomyłki drzewa lub interfejsu). Firma Cisco opracowała własny algorytm **drzewa rozpinającego w każdym VLAN-ie** (PVST — ang. *Per-VLAN Spanning Tree*)<sup>47</sup>, który zaktualizowała później (tworząc mechanizm PVST+) w taki sposób, aby był on zgodny ze standardem 802.1q. W 1998 roku organizacja IEEE przygotowała specyfikację 802.1w (**szybki protokół drzewa rozpinającego** [ang. *Rapid Spanning Tree Protocol*]), która zakłada skrócenie uzyskania spójności sieci po zmianie w topologii. Szybki protokół drzewa rozpinającego został opisany w standardzie 802.1d-2004 i obecnie zastępuje mechanizm STP. Istnieją także wersje algorytmu przeznaczone do stosowania w bardziej złożonych konfiguracjach sieci VLAN — **protokół wielu instancji drzewa rozpinającego** (MISTP — ang. *Multiple Instance Spanning Tree Protocol*) oraz **protokół wielu drzew rozpinających** (MSTP — ang. *Multiple Spanning Tree Protocol*). Rozwiązanie MSTP zostało włączone do standardu IEEE 802.1q-2003.

## 17.9. Przełączanie i przełączniki warstwy 2.

Informacje o sposobie działania mostów są doskonałym wprowadzeniem do rozważań na temat mechanizmów, które stanowią podstawę funkcjonowania nowoczesnych sieci Ethernet, czyli na temat **przełączania**. **Przełącznik ethernetowy**, nazywany niekiedy **przełącznikiem warstwy 2.**, jest urządzeniem elektronicznym o budowie zbliżonej do koncentratora. Podobnie jak koncentrator zawiera wiele portów, do których przyłączane są komputery, i pośredniczy w przekazywaniu ramek między jednostkami sieciowymi. Różnica między koncentratorem a przełącznikiem uwidacznia się w sposobie działania urządzeń.

<sup>47</sup> Sieci VLAN zostały opisane w kolejnym punkcie.

Koncentrator jest komponentem analogowym, który przekazuje sygnały między komputerami. Przełącznik natomiast jest elementem cyfrowym pośredniczącym w transporcie pakietów. Koncentrator można rozpatrywać jako wspólne medium transmisyjne. Natomiast przełącznik odpowiada sieci podzielonej mostami, w której każdy komputer pracuje w osobnym segmencie LAN. Koncepcja umieszczenia logicznych mostów wewnętrznych przełącznika została zilustrowana na rysunku 17.5.

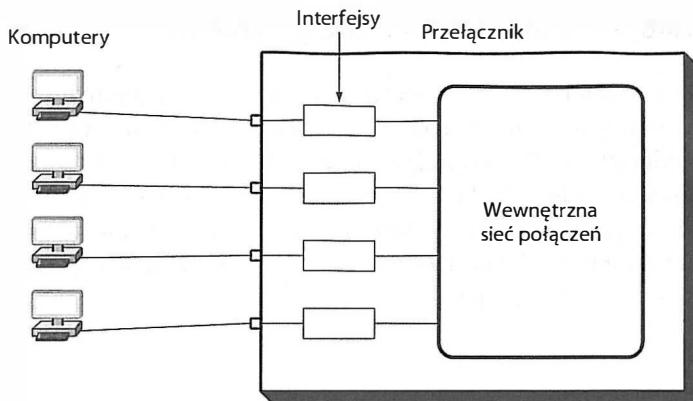


Rysunek 17.5. Ogólna budowa przełącznika sieci LAN

Na rysunku zaprezentowano jedynie ogólną koncepcję budowy przełącznika. W praktyce nie składa się on z wewnętrznych mostów, lecz z **inteligentnych interfejsów** skojarzonych z każdym portem oraz wewnętrznej **sieci połączeń** (ang. *fabric*), która umożliwia jednoczesny transfer strumieni między kilkoma parami interfejsów. Sam interfejs zawiera procesor, pamięć i inne komponenty niezbędne do odbierania nadchodzących pakietów, analizowania tabeli przełączania i wysyłania pakietów za pośrednictwem wewnętrznej sieci połączeń do odpowiedniego portu docelowego. Do zadań interfejsów należy również odbieranie pakietów z sieci połączeń i wysyłanie ich z portu sieciowego. Jeśli więc komputer 1 i komputer 2 w tym samym momencie wyśle dane do komputera 3, jeden z interfejsów (1 lub 2) wstrzyma na chwilę odebrany pakiet, aby drugi z pakietów mógł zostać dostarczony do jednostki docelowej. Rozwiążanie to przedstawiono na rysunku 17.6.

Przełączniki są dostępne w wielu rozmiarach. Najtańsze z nich są niezależnymi urządzeniami umożliwiającymi przyłączenie do czterech komputerów, co wystarcza do połączenia komputera z drukarką i dwoma innymi jednostkami (na przykład ze skanerem). W sieciach korporacyjnych wykorzystuje się znacznie większe przełączniki, które pozwalają na przyłączanie dziesiątek tysięcy komputerów i innych urządzeń firmowych.

Główna zaletą stosowania przełączników zamiast koncentratorów jest równoległość przetwarzania pakietów. Koncentrator może realizować tylko jedną transmisję w danym czasie. Natomiast przełącznik umożliwia jednoczesne transmitowanie wielu pakietów, jeśli transmisje te są od siebie niezależne (tylko jeden pakiet może być emitowany przez określony port w danym czasie). Jeśli więc przełącznik ma  $N$  portów przyłączonych do  $N$  komputerów, równolegle może być realizowanych  $N/2$  transmisji.



Rysunek 17.6. Architektura przełącznika

*Dzięki przetwarzaniu pakietów, a nie sygnałów i wykorzystaniu wewnętrznej sieci połączeń przełącznik o N portach może realizować maksymalnie  $N/2$  jednoczesnych transmisji.*

## 17.10. Przełączniki sieci VLAN

Standardowe przełączniki zostały rozszerzone o funkcję wirtualizacji sieci, która doprowadziła do powstania nowej grupy urządzeń — **przełączników wirtualnych sieci lokalnych (przełączników VLAN)**. Umożliwiają one administratorom emulowanie w jednym urządzeniu kilku niezależnych przełączników. Administrator wybiera kilka portów i przypisuje do pierwszej sieci wirtualnej (VLAN 1), następnie wskazuje kilka innych portów i umieszcza je w drugiej sieci wirtualnej (VLAN 2) itd. Gdy komputer jednej sieci VLAN 1 wyśle rozgłoszenie, kopia ramki zostanie dostarczona tylko do komputerów pracujących w tej samej sieci wirtualnej (VLAN 1). Po odpowiednim skonfigurowaniu przełącznik sieci VLAN działa tak, jakby został podzielony na kilka niezależnych przełączników.

Przypisywanie komputerów do różnych **domen rozgłoszeniowych** jest jednak konieczne jedynie w dużych sieciach korporacyjnych oraz w systemach dostawców usług internetowych. Tylko w takich przypadkach administrator musi mieć pewność, że pewna grupa komputerów ma możliwość komunikowania się oraz że jednostki spoza tej grupy nie otrzymują wymienianych pakietów. Jako przykład przeanalizujmy sieć, w której komputery zarządu firmy mają być oddzielone od komputerów pracowników za pomocą zapory sieciowej<sup>48</sup>. Wydzielenie oddzielnej sieci VLAN dla komputerów zarządu umożliwi zainstalowanie zapory sieciowej.

<sup>48</sup> Zapory sieciowe zostały opisane w rozdziale 30.

## 17.11. Funkcje mostu w innych urządzeniach

Choć z zamieszczonego omówienia wynika, że mosty są niezależnymi urządzeniami, funkcja mostu, jako fundamentalna koncepcja działania sieci, jest implementowana w różnych innych jednostkach. Pewien rodzaj mostu jest elementem składowym modemów DSL lub modemów kablowych, które zapewniają przekazywanie pakietów między abonentem a siecią dostawcy usług internetowych. Inne cechy mostu można odnaleźć w niektórych systemach transmisji bezprzewodowej, odpowiedzialnych za przesyłanie ramek z urządzeń mobilnych do sieci operatora.

A zatem:

*Mimo że mostów nie sprzedaje się już w formie oddzielnych urządzeń, realizowane przez nie funkcje są implementowane w wielu komponentach sieciowych, takich jak modemy dostępowe.*

## 17.12. Podsumowanie

Opracowano wiele mechanizmów pozwalających na rozszerzenie sieci LAN na większe obszary geograficzne. Problem przyłączenia komputera do odległej sieci LAN można na przykład rozwiązać za pomocą pary modemów optycznych. Użytecznymi komponentami są również regeneratorы, które wzmacniają sygnały elektryczne odbierane z jednego segmentu sieci LAN i przekazywane do drugiego segmentu. Urządzeniem odpowiedzialnym za łączenie dwóch segmentów sieci LAN i przekazywanie pakietów jest również most.

Aby zoptymalizować przekazywanie pakietów w sieci, most analizuje adresy MAC zawarte w nagłówkach ramek i zapamiętuje przynależność komputerów do określonych segmentów sieci. Po zgromadzeniu informacji na temat lokalizacji jednostek most nie powiela ramek, które są adresowane do jednostek pracujących w tym samym segmencie, w którym znajduje się nadawca.

Przełącznik ethernetowy przekazuje ramki między przyłączonymi do niego komputerami. Jego działanie przypomina pracę kilku segmentów sieci LAN połączonych za pomocą mostów. W praktyce przełącznik składa się z kilku inteligentnych interfejsów, które wymieniają dane za pośrednictwem wysokoprzepustowej sieci połączeń wewnętrznych. Przewaga przełącznika nad koncentratorem wynika z możliwości równoległego transmitowania wielu pakietów (przy założeniu, że do pojedynczego portu kierowany jest tylko jeden pakiet w danej chwili). Dzięki technice VLAN pojedynczy przełącznik można skonfigurować w taki sposób, aby pełnił funkcję wielu niezależnych urządzeń tego typu.

## ZADANIA

- 17.1. Jakie urządzenia są potrzebne do zwiększenia zasięgu sieci LAN za pomocą włókna optycznego?
- 17.2. Jeśli telewizor jest wyposażony w przewodowy przedłużacz odbiornika podczerwieni, jaka technologia najprawdopodobniej została wykorzystana?
- 17.3. Czy są potrzebne jakiekolwiek zmiany w adresowaniu lub działaniu aplikacji, jeśli dwa komputery zostaną połączone za pomocą mostu? Uzasadnij odpowiedź.
- 17.4. Wymień wszystkie sytuacje, w których most adaptacyjny przekazuje pakiety.
- 17.5. Załóżmy, że w sieci zawierającej mosty wysłano pakiet z nieistniejącym adresem docelowym. W ilu segmentach pakiet będzie transmitowany?
- 17.6. Załóżmy, że sieć składa się z trzech segmentów ethernetowych o przepustowości 100 Mb/s, połączonych dwoma mostami. W każdym segmencie pracuje jeden komputer. Załóżmy również, że dwa komputery wysyłają dane do trzeciego. Jaka jest maksymalna szybkość transmisji danych? A minimalna?
- 17.7. Poszukaj w internecie szczegółowych informacji na temat algorytmu drzewa rozpinającego i napisz program komputerowy, który zasymuluje pracę mostu wyznaczającego drzewo rozpinające.
- 17.8. Czy komputery przyłączone do sieci Ethernet otrzymują pakiety algorytmu STP? Uzasadnij odpowiedź.
- 17.9. Użyj programu do monitorowania pracy sieci, aby zarejestrować ruch w sieci Ethernet, w której działają mosty. Jakie ramki zostaną przechwycone po uruchomieniu mostu?
- 17.10. W połączeniach satelitarnych implementuje się mosty na obydwu końcach połączenia. Dlaczego?
- 17.11. Czy zgodnie z rysunkiem 17.5 dwa komputery przyłączone do sieci LAN mogą jednocześnie nadawać pakiety? Uzasadnij odpowiedź.
- 17.12. Zaproponuj rozbudowanie przełącznika z rysunku 17.5 o piąty port.
- 17.13. Określ (na podstawie wcześniejszego zadania) zależność między liczbą logicznych mostów a liczbą portów przełącznika.
- 17.14. Napisz program komputerowy, który będzie symulował pracę mostu. Niech dwa pliki symulują ramki transmitowane między dwoma segmentami mostu. Przyjmij założenie, że każda ramka ma adres źródłowy i docelowy. Test powinien polegać na naprzemiennym odczytywaniu ramek z jednego i drugiego pliku. Program powinien wyświetlać na ekranie informację o tym, czy przekazuje kopię ramki do drugiego segmentu sieci, czy nie.
- 17.15. Uzupełnij program z poprzedniego zadania o symulację sieci VLAN. Działanie programu powinno rozpoczynać się od odczytania informacji na temat przynależności komputerów do określonych sieci VLAN. Przygotuj pliki z ramkami odpowiadające poszczególnym komputerom (tj. portom przełącznika, przez które ramki będą dostarczane) i odpowiednim adresem docelowym. Zademonstruj działanie mechanizmu przekazywania ramek.
- 17.16. Czy most może odpowiadać za połączenie sieci Wi-Fi z Ethernatem? Czy można do tego celu użyć przełącznika? Uzasadnij odpowiedź.

# Zawartość rozdziału

- 18.1. Wprowadzenie 325
- 18.2. Sieci rozległe 325
- 18.3. Tradycyjna architektura sieci WAN 326
- 18.4. Budowanie sieci WAN 327
- 18.5. Zasada „zapisz i przekaż” 328
- 18.6. Adresacja w sieciach WAN 329
- 18.7. Wyznaczanie następnego skoku 330
- 18.8. Niezależność od źródła 332
- 18.9. Dynamiczne aktualizacje informacji o routingu w sieci WAN 332
- 18.10. Trasy domyślne 333
- 18.11. Wypełnianie tablicy przekazywania 334
- 18.12. Rozproszone mechanizmy wyznaczania tras 335
- 18.13. Wyznaczenie najkrótszej trasy w grafie 337
- 18.14. Problemy routingu 340
- 18.15. Podsumowanie 340

# 18

## *Technologie sieci WAN i routing dynamiczny*

### **18.1. Wprowadzenie**

Tematyka tej części książki koncentruje się na przewodowych i bezprzewodowych technologiach przełączania pakietów. W poprzednim rozdziale opisane zostały rozwiązania pozwalające na rozszerzanie zasięgu sieci LAN. Ten rozdział jest natomiast poświęcony strukturze sieci, która obejmuje dowolnie duży obszar geograficzny. Przedstawiono w nim podstawowe komponenty systemów pakietowych oraz zasadę działania najważniejszego mechanizmu sieci pakietowych, czyli routingu. W omówieniu uwzględnione zostały dwa najważniejsze algorytmy routingu wraz z ich zaletami i wadami. Tematem następnego rozdziału jest przekazywanie pakietów do internetu oraz działanie protokołów routingu, które wykorzystują opisane tutaj algorytmy.

### **18.2. Sieci rozległe**

Zgodnie z zamieszczonymi wcześniej informacjami jeden ze sposobów klasyfikowania technologii sieciowych polega na określeniu ich zasięgu. Wyróżnia się następujące grupy:

- PAN — obejmuje obszar wokół użytkownika.
- LAN — obejmuje obszar budynku lub kampusu.
- MAN — obejmuje obszar miasta.
- WAN — obejmuje obszar kilku miast lub krajów.

Zastanówmy się przez chwilę nad konfiguracją, w której korporacja wykorzystuje łącze satelitarne do połączenia sieci LAN działających w dwóch lokalizacjach. Czy takie rozwiązanie należałoby zaliczyć do grupy sieci WAN, czy sklasyfikować jako rozszerzoną sieć LAN? Czy na odpowiedź miałaby wpływ informacja o tym, że każda z sieci LAN składa się jedynie z komputera i drukarki? Tak, jest to istotna informacja. Najważniejszym elementem odróżniającym technologie WAN od rozwiązań LAN jest ich **skalowalność**. Systemy WAN muszą mieć zdolność rozrastania się i przyłączania w razie konieczności nowych sieci lokalnych rozmieszczonych w różnych odległych rejonach geograficznych. Na przykład sieć WAN powinna umożliwiać połączenie wszystkich komputerów dużej korporacji, której biura i fabryki są rozmieszczone w różnych miejscach na obszarze tysięcy kilometrów kwadratowych. Ponadto danej technologii nie można zaliczyć do grupy WAN, jeśli nie gwarantuje odpowiedniej wydajności połączeń w całej sieci.

Działanie systemów WAN nie ogranicza się jedynie do łączenia wielu komputerów z różnych lokalizacji, ale obejmuje również obowiązek zagwarantowania takiej pojemości sieci, która umożliwi wszystkim jednostkom wzajemną komunikację. Zatem most satelitarny łączący dwa komputery i dwie drukarki jest jedynie rozszerzeniem sieci LAN.

### 18.3. Tradycyjna architektura sieci WAN

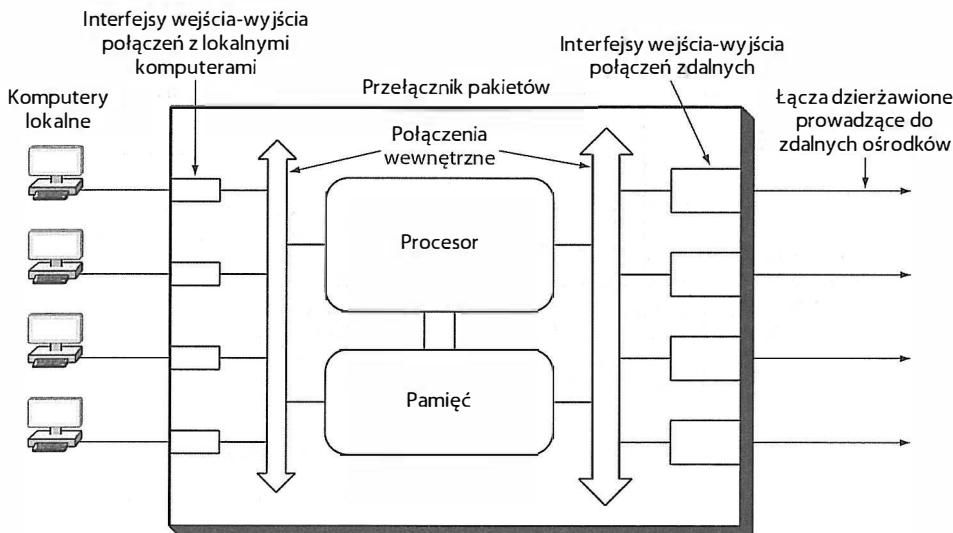
Tradycyjne technologie sieci WAN powstawały przed upowszechnieniem się sieci lokalnych, przed pojawiением się komputerów osobistych oraz przed rozpoczęciem prac nad internetem. Architektura ówczesnych rozwiązań ograniczała się więc do komponentów łączących kilka centrów, w których funkcjonowało po kilka dużych komputerów.

Brak technologii LAN sprawił, że projektanci systemów WAN budowali specjalne urządzenia, które instalowano w każdym z ośrodków. Moduły te, nazywane **przełącznikami pakietów**, zapewniały połączenia między komputerami lokalnymi, a także obsługiwały połączenia transmisji danych z innymi ośrodkami.

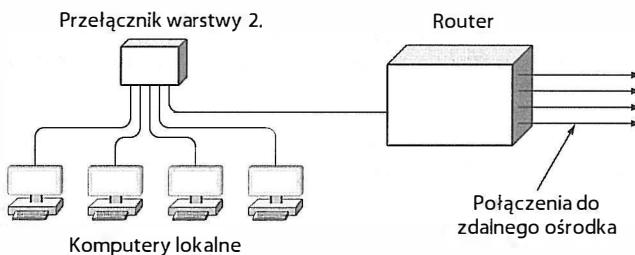
Przełącznik pakietów składa się z niewielkiego komputera, wyposażonego w procesor, pamięć operacyjną oraz urządzenia wejścia-wyjścia, które odbierają i wysyłają pakiety. Pierwsze przełączniki pakietów były budowane na bazie klasycznych komputerów. Jednak wykorzystanie ich w wysokowydajnych połączeniach sieci WAN wymusiło na projektantach systemów zastosowanie urządzeń przeznaczonych specjalnie do tego celu. Wewnętrzna budowa opisywanego przełącznika została pokazana na rysunku 18.1.

Jak nietrudno zauważać na rysunku, przełączniki pakietów zawierają dwa rodzaje urządzeń wejścia-wyjścia. Pierwsze z nich, gwarantujące dużą przepływność danych, obsługują obwody cyfrowe prowadzące do innych przełączników pakietów. Urządzenia drugiego rodzaju, o mniejszej wydajności, służą do przyłączania komputerów do przełącznika.

Wraz z upowszechnieniem się technologii LAN większość przełączników pakietów podzieliła się na dwie części — przełącznik warstwy 2, odpowiedzialny za połączenia z komputerami lokalnymi, oraz router, realizujący wymianę danych z innymi ośrodkami. Szczegółowy opis routerów internetowych znajduje się w czwartej części książki (uwzględniono w niej również te zagadnienia z bieżącego rozdziału, które mają zastosowanie w internecie). Do dalszej analizy wystarczy nam informacja, że komunikację z lokalnymi komputerami można oddzielić od transmisji w łączach WAN. Idea separacji została zilustrowana na rysunku 18.2.



Rysunek 18.1. Budowa tradycyjnego przełącznika pakietów

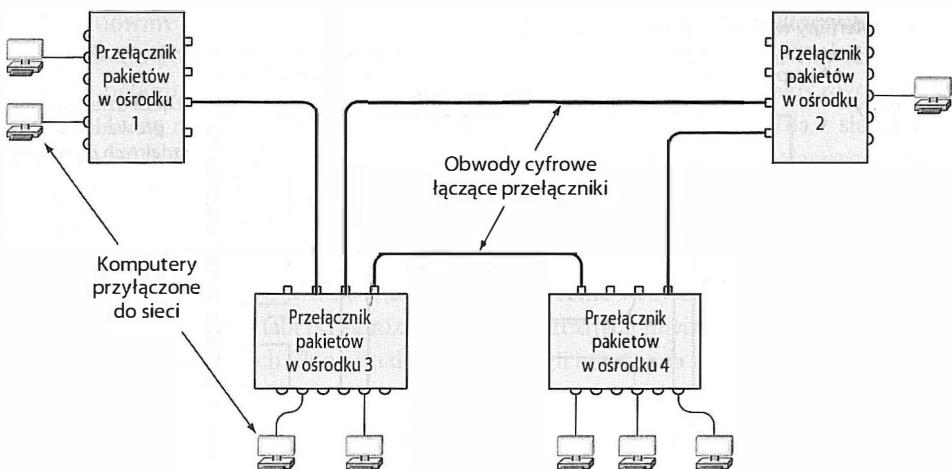


Rysunek 18.2. Nowoczesne połączenie WAN z niezależną obsługą ruchu w sieci LAN

## 18.4. Budowanie sieci WAN

Sieć WAN można zbudować, łącząc pewną liczbę odległych ośrodków. Szczegóły takich połączeń zależą od szybkości transmisji danych, odległości między ośrodkami oraz dopuszczalnego poziomu opóźnień. W wielu systemach WAN wykorzystuje się linie dzierżawione, opisane w rozdziale 12. (na przykład obwody E3 lub OC-12). Dostępne są również inne rozwiązania, w tym łączki mikrofalowe lub satelitarne. Poza wyborem odpowiedniej technologii połączenia projektant systemu musi również zdefiniować jego topologię. Lista opcji jest dłuża. Na rysunku 18.3 przedstawiono przykład połączenia czterech tradycyjnych przełączników pakietów i ośmiu komputerów.

Z analizy rysunku wynika, że połączenia WAN nie muszą być symetryczne — rodzaj połączenia między przełącznikami pakietów oraz przepustowość łączka można dobierać odpowiednio do spodziewanego natężenia ruchu. Wybrane połączenia można również zabezpieczyć na wypadek awarii łączami zapasowymi. Przełącznik pakietów w ośrodku 1 obsługuje tylko jedno połączenie, prowadzące do pozostałe części sieci. Natomiast pozostałe przełączniki są wyposażone w co najmniej dwa zewnętrzne łączki.



Rysunek 18.3. Przykład sieci WAN zbudowanej z przełączników pakietów

*Tradycyjna sieć WAN powstaje z połączenia przełączników pakietów. Przełącznik pakietów zainstalowany w danym ośrodku umożliwia kolejne przyłączanie lokalnych komputerów. Topologia i przepustowość połączeń są dobierane zależnie do spodziewanego natężenia ruchu oraz potrzeby zagwarantowania tras zapasowych.*

## 18.5. Zasada „zapisz i przekaż”

Celem sieci WAN jest umożliwienie jak największej liczby komputerów jednoczesnego przesyłania pakietów. Gwarancją realizacji wielu transmisji naraz jest stosowanie się do zasady **zapisz i przekaż** (ang. *store and forward*). Przetwarzanie zgodne z tą zasadą wymaga od przełącznika **buforowania** pakietów w pamięci podręcznej. Operacja **zapisu** jest realizowana w momencie odebrania pakietu — moduł wejścia-wyjścia stanowiący element składowy przełącznika pakietów zapisuje odbierane pakiety w pamięci urządzenia. Operacja **przekazania** rozpoczyna się wraz z zapisaniem pakietu w pamięci. Procesor urządzenia analizuje pakiet, ustala sieć docelową i wysyła interfejsem prowadzącym do wyznaczonego miejsca docelowego.

Systemy, w których zaimplementowano mechanizm „zapisz i przekaż”, utrzymują wszystkie łącza w stanie aktywności, co zwiększa ogólną wydajność rozwiązania. Ponadto, w przypadku dostarczania większej liczby pakietów do tego samego urządzenia wyjściowego nie ma potrzeby wstrzymywania strumieni wejściowych. Przełącznik może zapisywać pakiety w pamięci, gdzie będą oczekiwane na gotowość urządzenia docelowego. Jako przykład takiego sposobu działania można przeanalizować wymianę danych w sieci z rysunku 18.3. Założymy, że dwa komputery w ośrodku 1 w tym samym czasie wysyłają dane do komputera 3. Obydwie jednostki dostarczają generowane przez nie pakiety do przełącznika. Wraz z odebraniem każdej porcji danych moduł wejścia-wyjścia przełącznika zapisuje je w pamięci podręcznej i informuje o tym fakcie procesor urządzenia. Procesor sprawdza adres docelowy każdego z pakietów i wybiera trasę do ośrodka 3. Jeśli

interfejs wyjściowy prowadzący do ośrodka 3 jest nieaktywny, transmisja rozpoczyna się bezzwłocznie. Jeżeli jednak moduł wyjściowy jest zajęty, procesor zapisuje pakiet w kolejce wyjściowej danego modułu.

Rozwiązań to można podsumować w następujący sposób:

*Systemy przełączania pakietów w sieciach rozległych wykorzystują technikę „zapisz i przekaź” do umieszczania nadchodzących pakietów w kolejkach, z których pakiety te są pobierane w chwili, gdy przekazanie ich do jednostek docelowych staje się możliwe. Dzięki temu poprawnie obsługiwane są krótkie zbitki generowanych w tym samym czasie pakietów.*

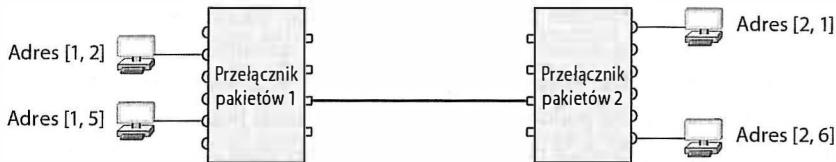
## 18.6. Adresacja w sieciach WAN

Od strony przyłączonego komputera działanie tradycyjnych sieci WAN nie różni się niczym od pracy sieci LAN. W każdej technologii WAN podczas wymiany danych z jednostką lokalną stosowane są dokładnie takie same formaty ramek, jakimi posługują się komputery. Każdemu komputerowi przyłączonemu do sieci WAN przypisany jest pewien adres. Nadawca ramki musi uwzględnić ten adres w treści ramki, gdy wysyła ją do jednostki docelowej.

Choć szczegóły implementacji bywają różne, mechanizm wyznaczania adresów w sieciach WAN pozwala na zastosowanie **adresowania hierarchicznego**. Idea adresowania hierarchicznego polega na wydzieleniu w każdym adresie sieciowym dwóch pól:

[ośrodek, komputer w ośrodku]

W praktyce identyfikacja ośrodka sprawdza się do przypisania mu niepowtarzalnego numeru. To z kolei oznacza, że pierwsza część adresu określa przełącznik pakietów, a druga część identyfikuje komputer. Na rysunku 18.4 przedstawiono przykłady adresów, które zostały przypisane komputerom przyłączonym do dwóch przełączników pakietów.



**Rysunek 18.4.** Przykład hierarchii adresowania — każdy adres zawiera informację o przełączniku pakietów oraz przyłączonym do niego komputerze

Na rysunku każdy adres przedstawiono jako parę liczb całkowitych. Na przykład komputer przyłączony do portu 6 przełącznika 2 ma adres [2,6]. W praktyce adresy są zapisywane w formie pojedynczej wartości binarnej, której część bitów odpowiada przełącznikowi pakietów, a pozostałe bity identyfikują komputer. W części czwartej książki opisano schemat adresowania obowiązujący w internecie. Czytając tę część, będzie można się dowiedzieć o tym, że adresy internetowe są wartościami binarnymi, złożonymi z bitów wyznaczających sieć oraz z bitów odpowiadającymi komputerom przyłączonym do tej sieci.

## 18.7. Wyznaczanie następnego skoku

Szczególne znaczenie hierarchicznego adresowania uwidacznia się, gdy zaczniemy analizować sposób przetwarzania pakietów. Po odebraniu pakietów przełącznik musi wyznaczyć interfejs wyjściowy, za którego pomocą przekaże dane dalej. Jeśli pakiet jest adresowany do komputera lokalnego, przełącznik dostarczy informacje bezpośrednio do wskazanej jednostki. W przeciwnym przypadku musi je przekazać do innego przełącznika za pośrednictwem jednego z łącz wyjściowych. Aby ustalić sposób dostarczania danych, urządzenie analizuje zapisany w pakiecie adres docelowy i wyodrębnia z niego identyfikator przełącznika. Jeśli wartość zapisana w adresie odpowiada identyfikatorowi danego przełącznika, dane są przeznaczone dla jednego z komputerów lokalnych. W przeciwnym przypadku pakiet musi zostać przesłany do innego przełącznika. Algorytm 18.1 opisuje ten mechanizm.

**Algorytm 18.1.** Dwa sposoby postępowania w czasie przekazywania pakietu

Dane:

Pakiet dostarczony do przełącznika pakietów Q

Cel:

Wyznaczenie następnego węzła na trasie

Realizacja:

Wyodrębnienie adresu docelowego z pakietu.

Podział adresu na numer przełącznika (P) oraz identyfikator komputera (C).

`if (P == Q) { /*celiem jest dany przełącznik*/`

`Przekazanie pakietu do komputera lokalnego`

`} else {`

`Wybranie łącza do kolejnego przełącznika i przesłanie pakietu tym łączem.`

`}`

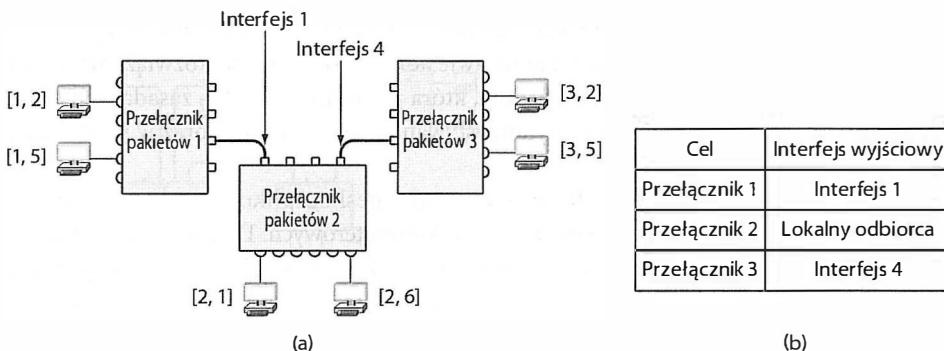
Najważniejszym założeniem jest to, że przełącznik nie musi przechowywać informacji o dostępności wszystkich komputerów. Nie ma też obowiązku wyznaczania całej trasy pakietu przez sieć. Przekazywanie danych sprowadza się do odczytania identyfikatora przełącznika i odszukania łącza, które prowadzi do tego przełącznika.

Przełącznik musi określić jedynie **następny skok** (ang. *next hop*) pakietu. Proces wyznaczania następnego skoku jest podobny do sposobu zestawiania połączeń przez linie lotnicze. Założymy, że pasażer podróżujący z Bydgoszczy do Lizbony musi dwa razy się przesiadać — w Warszawie i Frankfurcie. Mimo że port docelowy (Lizbona) nie zmienia się przez całą podróż, lotnisko następnego skoku jest inne na każdym etapie trasy. Gdy pasażer znajduje się w Bydgoszczy, następny skok to przelot do Warszawy. Gdy wylatuje z Warszawy, portem następnego skoku jest Frankfurt, a stamtąd następny skok wiedzie do Lizbony.

W celu zwiększenia efektywności pracy przełączników pakietów wykorzystują mechanizm odczytu informacji z uprzednio przygotowanej tablicy (ang. *table lookup*). Każdy przełącznik dysponuje bowiem **tablicą przekazywania** (ang. *forwarding table*)<sup>49</sup>. Proces

<sup>49</sup> Puryści językowi nalegają na używanie określenia **tablica przekazywania**, mimo że pierwotnie stosowany był termin **tablica routingu**, którym wielu inżynierów nadal się posługuje.

przekazywania pakietów do urządzenia następnego skoku został zilustrowany na przykładzie zamieszczonym na rysunku 18.5.



Rysunek 18.5. Sieć złożona z trzech przełączników pakietów (a)  
oraz tabela przekazywania pakietów przełącznika 2 (b)

Aby skorzystać z tablicy przekazywania, przełącznik odczytuje adres docelowy pakietu i posługuje się częścią identyfikującą przełącznik jak indeksem w tablicy. Jako przykład przeanalizujmy tabelę z rysunku 18.5b. Gdy urządzenie odbierze pakiet przeznaczony do jednostki [3,5], wyodrębní wartość 3, wyszuka ją w tablicy i wyśle dane za pośrednictwem interfejsu 4 (który prowadzi do przełącznika 3).

Wykorzystanie tylko jednej części dwuelementowego adresu do przekazywania pakietów ma praktyczne konsekwencje. Po pierwsze, skraca się czas wykonywania operacji, ponieważ dane odnośnie przekazywania pakietów są zapisywane w formie klasycznej tablicy o indeksowanych elementach. Nie trzeba jej więc przeszukiwać (wystarczy znać indeks). Po drugie, tablica przekazywania zawiera tylko wpisy odpowiadające przełącznikom (nie obejmuje adresów komputerów). Dzięki temu ma ona znacznie mniejszy rozmiar, co jest szczególnie istotne w przypadku rozległych sieci WAN o wielu komputerach przyłączonych do poszczególnych przełączników.

Podział adresów na dwie części oraz ich hierarchiczność pozwalają przełącznikom pakietów na przetwarzanie jedynie początkowego fragmentu adresu docelowego, aż do momentu dostarczenia danych do przełącznika końcowego (tj. przełącznika, do którego przyłączony jest komputer docelowy). Gdy pakiet dotrze do takiego przełącznika, urządzenie wybiera odpowiedni komputer na podstawie drugiej części adresu, zgodnie z algorytmem 18.1.

Podsumowując:

*Podczas przekazywania pakietów przez sieć WAN analizowana jest jedynie pierwsza część adresu docelowego. Gdy dane zostaną dostarczone do przełącznika końcowego, druga część adresu decyduje o wyborze komputera, do którego pakiet zostanie dostarczony.*

## 18.8. Niezależność od źródła

Przekazywanie danych do kolejnych przełączników nie zależy od źródła pakietu ani od trasy, którą pakiet pokonał przed dostarczeniem go do danego przełącznika. Kolejny skok jest natomiast wyznaczany jedynie na podstawie adresu docelowego. Rozwiążanie to jest zgodne z koncepcją **niezależności od źródła**, która jest fundamentalną zasadą transmisji sieciowych i będzie często niejawnie wykorzystywana w tym rozdziale oraz w rozdziałach kolejnych.

Niezależność od źródła umożliwia budowanie nieskomplikowanych i wydajnych mechanizmów przekazywania danych w sieciach komputerowych. Dzięki temu, że wszystkie pakiety (niezależnie od źródła) są przesyłane tą samą trasą, do wykonania zadania potrzebna jest tylko jedna tablica. Algorytmy wyboru tras nie uwzględniają informacji o źródle pakietów, więc całą operację można sprowadzić do weryfikacji adresu docelowego, który jest zawarty w samym pakiecie. Ponadto mechanizm przekazywania działa jednakowo niezależnie od tego, czy pakiety przetwarzane przez przełącznik pochodzą z komputerów przyłączonych lokalnie, czy z innych przełączników.

## 18.9. Dynamiczne aktualizacje informacji o routingu w sieci WAN

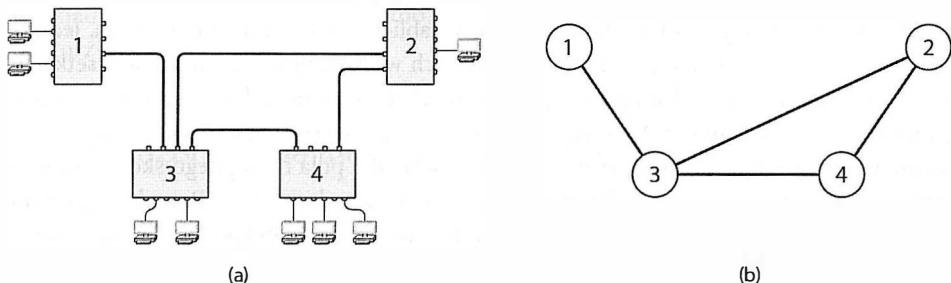
Aby sieć WAN pracowała poprawnie, każdy przełącznik musi dysponować tablicą przekazywania pakietów, na podstawie której dostarcza informacje. Ponadto tablica przekazywania musi gwarantować, że:

- Przekazywanie danych jest poprawne. Tablice przekazywania w każdym przełączniku muszą zawierać poprawne dane na temat kolejnego skoku niezależnie od adresu docelowego.
- Trasy są dobierane optymalnie. Zapisane w przełączniku informacje o następnym skoku muszą zapewniać wybór najkrótszej trasy do określonego punktu docelowego.

Działanie mechanizmu jest dodatkowo utrudnione przez awarie sieci. Na przykład jeśli do danego węzła docelowego wiodą dwie trasy, a jedna z nich stanie się niedostępna z powodu uszkodzenia sprzętowego, system powinien automatycznie zmienić swój sposób działania tak, aby unikać nieaktywnej trasy. Administrator nie może więc zapisać na stałe treści tablic przekazywania. Konieczne jest uruchomienie w przełącznikach oprogramowania, które będzie na bieżąco analizowało stan połączeń i w razie konieczności automatycznie modyfikowało zawartość tablicy. Oprogramowanie odpowiedzialne za automatyczną rekonfigurację tablic nazywa się **oprogramowaniem routingu**.

Najłatwiejszy sposób wyznaczania tras w sieci WAN polega na potraktowaniu tej sieci jak grafu, na którego podstawie program wyliczy najkrótsze trasy do wszystkich węzłów docelowych. Każdy **wierzchołek** (węzeł) w grafie odpowiada jednemu przełącznikowi pakietów pracującemu w sieci (komputery nie są reprezentowane na grafie). Jeśli w sieci

występuje bezpośrednie połączenie między dwoma przełącznikami pakietów, reprezentujące je w grafie węzły powinny zostać połączone **krawędziami** (łączami)<sup>50</sup>. Na rysunku 18.6 pokazano przykładową sieć WAN i odpowiadający jej graf.



Rysunek 18.6. Sieć WAN i odpowiadający jej graf

Wszystkie węzły grafu zostały oznaczone takim samym numerem, jakie mają odpowiadające im przełączniki pakietów. Grafowa reprezentacja sieci okazuje się szczególnie użyteczna do wyznaczania kolejnych węzłów na trasie (w procesie routingu). Pozwala bowiem na zastosowanie mocno rozwiniętej teorii grafów oraz jej wydajnych algorytmów. Takie podejście uniezależnia również analizę od zastosowanego oprogramowania, pozwalając skoncentrować się na zasadniczym problemie.

Algorytm wyznaczania kolejnych węzłów na trasie pakietu musi w pewien sposób operować informacjami o łączach. W przedstawionych dalej przykładach zastosowano notację  $(k, j)$ , która opisuje przejście z węzła  $k$  do węzła  $j$ . Zastosowanie algorytmu routingu w odniesieniu do grafu z rysunku 18.6b daje więc wynik, który pokazano w tabeli 18.1.

Tabela 18.1. Tablica przekazywania pakietów każdego z węzłów z rysunku 18.6b

Cel	Następny skok						
1	-	1	(2,3)	1	(3,1)	1	(4,3)
2	(1,3)	2	-	2	(3,2)	2	(4,2)
3	(1,3)	3	(2,3)	3	-	3	(4,3)
4	(1,3)	4	(2,4)	4	(3,4)	4	-

Węzeł 1
Węzeł 2
Węzeł 3
Węzeł 4

## 18.10. Trasy domyślne

Analiza tablicy przekazywania węzła 1 (widoczna w tabeli 18.1) prowadzi do ciekawego spostrzeżenia — tablica może zawierać wiele wpisów odnoszących się do jednego łącza. Przyczyna takiego stanu staje się oczywista po spojrzeniu na schemat sieci (widoczny na

<sup>50</sup> Rozwiązywanie sieciowe zazwyczaj bazują na teorii grafów, dlatego często przełączniki pakietów nazywa się **węzłami sieci**, a obwody wymiany danych **łączami**.

rysunku 18.6a). Przełącznik 1 ma tylko jedno połączenie z pozostałą częścią sieci. Zatem cały ruch wychodzący musi być przekazywany tym samym łączem. W konsekwencji wszystkie wpisy zawarte w tablicy przekazywania węzła 1, poza wierszem opisującym sam węzeł, zawierają informacje o obowiązku przesyłania pakietów z węzła 1 do 3.

W omawianym przypadku lista duplikatów w tablicy przekazywania jest krótka. Jednak w rzeczywistych sieciach WAN liczba powielonych wpisów może być liczona w setkach. Większość systemów uwzględnia więc specjalny mechanizm, który pozwala na eliminowanie duplikatów. Rozwiążanie polega na definiowaniu **tras domyślnych**, czyli pojedynczego wpisu, który zastępuje grupę wierszy o tej samej wartości pola następnego skoku. Tablica przekazywania może zawierać definicję tylko jednej trasy domyślnej. Ponadto wpis taki ma najniższy priorytet. Jeśli algorytm doboru tras nie znajdzie bezpośredniej informacji o poszukiwanym adresie docelowym, posługuje się danymi trasy domyślnej. W tabeli 18.2 przedstawiono zmodyfikowaną wersję tablic przekazywania, w których zostały uwzględnione trasy domyślne.

**Tabela 18.2.** Zmodyfikowane wersje tablic przekazywania z tabeli 18.1,  
w których trasy domyślne oznaczono symbolem gwiazdki

Cel	Następny skok						
1	-	2	-	1	(3,1)	2	(4,2)
*	(1,3)	4	(2,4)	2	(3,2)	4	-
		*	(2,3)	3	-	*	(4,3)
				4	(3,4)		

Węzeł 1

Węzeł 2

Węzeł 3

Węzeł 4

Korzystanie z tras domyślnych ma charakter opcjonalny. Wpis o trasie domyślnej jest dodawany tylko wtedy, gdy większa liczba przełączników docelowych jest osiągalna przez ten sam węzeł kolejny. Na przykład tablica przekazywania węzła 3 nie zawiera wpisu o trasie domyślnej, ponieważ w każdym wierszu występuje inna wartość kolumny następnego skoku. Niemniej w przypadku węzła 1 widać wyraźną korzyść z zastosowania trasy domyślnej.

## 18.11. Wypełnianie tablicy przekazywania

W jaki sposób pozyskiwane są wartości do tablicy przekazywania? Stosuje się w tym względzie dwa podejścia:

- **Routing statyczny.** Program wyznacza trasy i wprowadza informacje o nich do tablicy podczas uruchamiania przełącznika pakietów. Później nie ulegają one zmianom.
- **Routing dynamiczny.** Program tworzy wstępную tablicę przekazywania podczas uruchamiania przełącznika pakietów, a następnie zmienia zawarte w niej informacje wraz ze zmianą stanu sieci.

Każde z rozwiązań ma pewne wady i zalety. Główną zaletą statycznego routingu jest jego prostota, a tym samym mały narzut obliczeniowy. Do wad trzeba zaliczyć nieelastyczność — trasy statyczne nie są modyfikowane automatycznie w przypadku awarii sieci. Ponieważ duże sieci WAN są budowane w taki sposób, aby występowali w nich połączenia nadmiarowe (na wypadek uszkodzeń sprzętowych), większość systemów tego typu wykorzystuje routing dynamiczny.

## 18.12. Rozproszone mechanizmy wyznaczania tras

Algorytm 18.2 opisuje sposób na utworzenie tablicy przekazywania na podstawie informacji o sieci zapisanej w grafie. W praktyce jednak obliczanie tras w sieci WAN jest realizowane w sposób rozproszony. Oznacza to, że nie ma centralnego oprogramowania, które wyznacza wszystkie najkrótsze trasy. Każdy przełącznik pakietów musi takie obliczenie wykonać lokalnie w odniesieniu do własnej tablicy przekazywania.

Wszystkie przełączniki pakietów współdziałają jednak w wyznaczaniu tras. Odpowiadają za to mechanizmy należące do dwóch poniższych kategorii:

- Routing na bazie informacji o stanie łączy (LSR — ang. *Link-State Routing*), w którym stosuje się algorytm Dijkstry.
- Routing z wykorzystaniem wektorów odległości (DVR — ang. *Distance-Vector Routing*).

Obydwa rozwiązania zostały opisane w dalszych podrozdziałach. Z kolei wykorzystanie ich do określania tras w internecie jest tematem rozdziału 27.

### 18.12.1. Routing na bazie informacji o stanie łączy (LSR)

Routing z wykorzystaniem informacji o **stanie łączy** jest również znany jako routing SPF (od angielskich słów *Shortest Path First*, oznaczających wybieranie w pierwszej kolejności najkrótszej trasy). Według autora algorytmu Edsgera Dijkstry nazwa ta oddaje sposób działania mechanizmu. Jest niestety nieco myląca, ponieważ wszystkie algorytmy routingu mają za zadanie znalezienie najkrótszych tras.

Działanie routingu LSR wymaga od przełączników pakietów okresowego rozsyłania w sieci informacji o stanie łączy między dwoma przełącznikami. Na przykład przełączniki o numerach 5 i 9 muszą sprawdzić stan łącza między nimi, a następnie wysłać do sieci informację o treści „łącze pomiędzy węzłami 5 i 9 działa poprawnie”. Każda informacja statusowa jest dostarczana do wszystkich przełączników. W każdym z urządzeń działa oprogramowanie, które zbiera nadchodzące komunikaty i buduje na ich podstawie graf sieci. Każdy z przełączników uruchamia również specjalną funkcję działającą zgodnie z algorytmem 18.2, której zadanie polega na utworzeniu tablicy przekazywania z danym przełącznikiem pracującym jako węzeł źródłowy.

Algorytm LSR zapewnia rekonfigurację sieci po wystąpieniu awarii sprzętowej. W przypadku uszkodzenia łącza między przełącznikami pakietów urządzenia znajdujące się na jego końcach wykryją awarię i rozesią komunikat informujący o niedostępności danego

**Algorytm 18.2.** Wersja algorytmu Dijkstry wyznaczająca węzły następnego skoku w tablicy przełączania (R) oraz odległości (D) do każdego węzła z danego węzła źródłowego

Dane:

Graf z nieujemnymi wagami przypisanymi do każdej krawędzi oraz ze wskazanym węzłem źródłowym.

Wynik:

Najkrótsze trasy do wszystkich węzłów z węzła źródłowego oraz tablica routingu z węzłami następnego skoku

Realizacja:

Zainicjowanie zbioru  $S$  tak, aby zawierał wszystkie węzły oprócz źródłowego.

Zainicjowanie tablicy  $D$  tak, aby elementowi  $D[v]$  odpowiadała waga krawędzi z węzła źródłowego do węzła  $v$ , o ile taka krawędź istnieje. W przeciwnym przypadku  $D[v]$  ma wartość **nieskończoną**.

Zainicjowanie tablicy  $R$  tak, aby element  $R[v]$  miał wartość  $v$  w przypadku, gdy istnieje krawędź między węzłem źródłowym a  $v$ . W przeciwnym przypadku  $R[v]$  ma wartość zerową.

```
while(zbiór S nie jest pusty) {
    Wybranie takiego węzła u ze zbioru S, aby element D[u]
    miał najmniejszą wartość.
    if (D[u] ma wartość nieskończoną) {
        Błąd: nie istnieje żadna trasa do węzłów ze zbioru S;
        koniec działania
    }
}
```

Usunięcie  $u$  ze zbioru  $S$ .

```
Dla każdego węzła  $v$ , dla którego istnieje krawędź  $(u, v)$  {
    if (v nadal zawiera się w S) {
        c = D[u] + waga(u, v)
        if (c < D[v]) {
            R[v] = R[u]
            D[v] = c;
        }
    }
}
```

odcinka. Wszystkie przełączniki, które odbiorą tę informację, zmienią graf tak, aby odzwierciedlał bieżący stan sieci, i ponownie wyznaczą najkrótsze trasy. Gdy awaria zostanie usunięta, przełączniki wykryją dostępność łączą i rozesią komunikaty o jego dostępności.

### 18.12.2. Routing z wykorzystaniem wektorów odległości (DVR)

Główną alternatywą dla rozwiązań LSR jest mechanizm obliczania **wektorów odległości** (DVR — ang. *Distance Vector Routing*). Podobnie jak w przypadku routingu LSR każdemu łączu sieciowemu przypisywana jest pewna waga, a **odległość** do celu oblicza się jako sumę wag na trasie między dwoma przełącznikami pakietów. Mechanizm DVR, podobnie jak LSR, wymaga okresowej wymiany komunikatów między przełącznikami. Nakłada jed-

nak na przełącznik obowiązek rozsyłania całej listy sieci docelowych znanych przełącznikowi wraz z informacjami o koszcie dostarczenia pakietów do każdej z nich. Transmisja komunikatu DVR sprowadza się do wysłania serii stwierdzeń typu:

Znam trasę do celu X. Odległość ode mnie wynosi Y.

Komunikaty DVR nie są dostarczane do wszystkich urządzeń. W ich wymianę zaangażowane są jedynie przełączniki sąsiednie. Każda wiadomość składa się z kilku pozycji o treści (**cel, odległość**). Przełączniki pakietów przechowują listę wszystkich potencjalnych sieci docelowych wraz z informacjami o odległości do nich oraz identyfikatorami najbliższych węzłów na trasie. Wykaz celów i dane następnego skoku są zapisane w tablicy przekazywania pakietów. Oprogramowanie DVR można więc postrzegać jako rozszerzenie tablicy przekazywania, które zapewnia rejestrację **odległości** do każdego celu.

Po odebraniu komunikatu od węzła  $N$  przełącznik pakietów analizuje każdy wpis. Jeśli którykolwiek z nich opisuje trasę krótszą do danego celu niż zapisana w tablicy przekazywania, stosowny wiersz tablicy jest modyfikowany. Na przykład jeśli sąsiad  $N$  informuje o tym, że zna trasę do celu  $D$  o koszcie pięć, a bieżąca trasa do celu  $D$  ma koszt sto i wiedzie przez węzeł  $K$ , to bieżący identyfikator węzła następnego skoku na trasie do  $D$  zostanie zastąpiony identyfikatorem węzła  $K$ , a koszt będzie odpowiadał kosztowi dojścia do celu  $D$  powiększonemu o koszt dojścia do węzła  $N$ . Zasada aktualizacji informacji o trasach zgodnie z opisany mechanizmem została przedstawiona w algorytmie 18.3.

## 18.13. Wyznaczenie najkrótszej trasy w grafie

Po utworzeniu grafu odpowiadającego rzeczywistej sieci oprogramowanie przełącznika uruchamia **algorytm Dijkstry**<sup>51</sup> do wyznaczenia najkrótszych tras z węzła źródłowego do wszystkich pozostałych węzłów w sieci. Jednocześnie wraz z wyliczaniem tras budowana jest tablica przekazywania pakietów. Algorytm Dijkstry musi zostać uruchomiony w każdym węźle grafu. Oznacza to, że aby wypełnić tablicę przekazywania pakietów w węźle  $P$ , konieczne jest wykonanie procedury obliczeniowej z węzłem  $P$  jako węzłem źródłowym.

Popularność algorytmu Dijkstry wynika przede wszystkim z tego, że można go stosować niezależnie od tego, jak zostanie zdefiniowana **najkrótsza trasa**. Mechanizm nie wymaga bowiem oznaczania krawędzi grafu wartościami odpowiadającymi rzeczywistym odległościom geograficznym. Każdej krawędzi można natomiast przypisać dowolną nieujemną wartość nazywaną **wagą**. Odległość między dwoma węzłami jest natomiast liczona jako suma wag wszystkich krawędzi występujących na trasie między tymi węzłami. Najważniejsze jest jednak to, że:

*Dzięki wykorzystaniu wag do wyznaczania najkrótszych tras w algorytmie Dijkstry mogą być stosowane dowolne miary, niekoniecznie odpowiadające odległościom geograficznym.*

<sup>51</sup> Nazwa algorytmu pochodzi od nazwiska jego twórcy — Edsgera Dijkstry.

**Algorytm 18.3.** Algorytm wektora odległości przeznaczony do wyznaczania najkrótszych tras

Dane:

Lokalna tablica przekazywania pakietów z określonymi odległościami w każdym wpisie, odległość do każdego sąsiada oraz komunikat dostarczony od jednego z sąsiadów.

Wynik:

Aktualizowana tablica przekazywania pakietów.

Realizacja:

Odczytanie wartości **odległości** z każdego wpisu w tablicy przekazywania.

Zainicjowanie tablicy przekazywania pojedynczym wpisem, w którym identyfikator **celu** odpowiada lokalnemu przełącznikowi, pole **następnego skoku** jest nieużywane, a **odległość** wynosi zero.

Pętla nieskończona {

Oczekiwanie na odbiór komunikatu od sąsiada. Niech nadawca będzie przełącznik N.

Dla każdego wpisu w komunikacie {

Niech V będzie identyfikatorem celu, a D odległością do celu.

Obliczenie C jako wartości D powiększonej o wagę łączącej, którym został przesłany komunikat.

Sprawdzenie i uaktualnienie lokalnej tablicy routingu:  
if (nie ma trasy do V) {

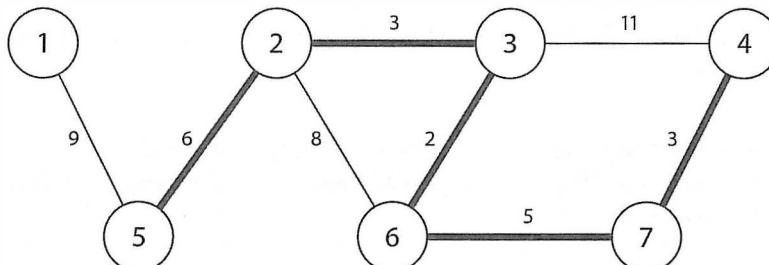
Dodanie trasy do lokalnej tablicy routingu (z danym celem).

} else if (istnieje trasa z kolejnym węzłem N) {  
Zastąpienie wartości odległości w istniejącej trasie wartością C.

} else if (istnieje trasa o odległości większej niż C) {  
Zmiana kolejnego węzła na N i odległości na C.  
}  
}

}

Koncepcja uniwersalnych wag została przedstawiona na rysunku 18.7, na którym widać liczby całkowite reprezentujące odległości oraz trasę między dwoma węzłami o najniższej sumie wag.



Rysunek 18.7. Przykład grafu z wagami przypisanymi do krawędzi oraz zaznaczoną najkrótszą trasą między węzłami 4 i 5

Algorytm Dijkstry operuje zbiorem węzłów ( $S$ ), dla których obliczane są minimalne odległości i węzły następnego skoku. Początkowo zbiór składa się ze wszystkich węzłów z wyjątkiem węzła źródłowego. Działanie algorytmu trwa aż do wyczerpania zbioru. W każdej iteracji usuwany jest jeden węzeł — ten, który ma najmniejszą odległość do węzła źródłowego. Wraz z usunięciem węzła  $u$  sprawdzana jest odległość między węzłem źródłowym i sąsiadami  $u$ , którzy pozostają w zbiorze. Jeśli trasa z węzła źródłowego przez  $u$  do wybranego sąsiada ma mniejszą wagę niż trasa dotychczas obowiązująca, odległość do sąsiada węzła  $u$  jest uaktualniana. Po usunięciu wszystkich elementów ze zbioru  $S$  obliczone są najmniejsze odległości do wszystkich węzłów sieci. Wyznaczona jest również tablica przekazywania, która zawiera wszystkie możliwe do wykorzystania trasy.

Implementacja algorytmu Dijkstry nie nastręcza trudności. Poza zbiorami danych przechowującymi informacje o budowie grafu mechanizm wymaga przygotowania trzech dodatkowych struktur przeznaczonych na bieżące odległości do każdego węzła, dane następującego skoku w ramach każdej trasy oraz zbiór pozostałych do przeanalizowania węzłów. Węzły warto ponumerować od 1 do  $n$  (zgodnie z rysunkiem 18.7). Zwiększa to wydajność algorytmu, ponieważ numery węzłów można potraktować jak indeksy struktury danych. W zaprezentowanym przykładzie wykorzystano dwie tablice ( $D$  i  $R$ ), które są indeksowane z użyciem numerów węzłów —  $i$ -ty element tablicy  $D$  przechowuje bieżącą wartość minimalnej odległości między węzłem źródłowym a węzłem  $i$ , natomiast  $i$ -ty element tablicy  $R$  odpowiada następnemu skokowi na trasie do węzła  $i$ . Zbiór  $S$  można przedstawić za pomocą listy dwukierunkowej z numerami węzłów. Taka struktura zapewnia łatwe przeszukiwanie zbioru i usuwanie jego elementów.

Sposób wyznaczania najkrótszych tras w grafie został opisany w algorytmie 18.2. Wykorzystano w nim funkcję  $waga(i, j)$ , która zwraca wagę krawędzi między węzłami  $i$  i  $j$ . Funkcja ta musi również zwracać pewną zarezerwowaną wartość (interpretowaną jako **nieskończoność**) w przypadku braku krawędzi między węzłami  $i$  i  $j$ . W praktyce nieskończoność może być reprezentowana przez dowolną wartość większą niż suma wag wszystkich krawędzi. Najprostszy sposób doboru wartości polega na dodaniu 1 do sumy wszystkich wag.

Możliwość przypisywania dowolnych wag krawędziom grafu oznacza, że jeden algorytm znajduje zastosowanie w przetwarzaniu różnych miar odległości. Na przykład w niektórych technologiach WAN odległość określa się jako liczbę przełączników na trasie do określonego miejsca docelowego. Aby użyć opisanego algorytmu, wystarczy więc przypisać każdej krawędzi w grafie wagę 1. W innych rozwiązańach WAN wagi odzwierciedlają pojemność poszczególnych łącz. A dzięki możliwości przypisywania wag poszczególnym łączom administrator zyskuje wpływ na sposób przekazywania pakietów w sieci. Jako przykład przeanalizujmy przypadek, w którym między dwoma przełącznikami pakietów istnieją dwie niezależne trasy. Jedna z nich powinna mieć status trasy **podstawowej**, a druga **zapasowej**. Aby zagwarantować odpowiedni dobór połączeń, administrator może przypisać niższą wagę do jednego łącza i wyższą do drugiego. Oprogramowanie zarządzające routingu skonfiguruje wówczas tablice przekazywania w taki sposób, aby trasa o niższej wadze była wykorzystywana zawsze, gdy jest dostępna. W razie jej uszkodzenia wybrana zostanie trasa zapasowa.

## 18.14. Problemy routingu

Teoretycznie obydwa algorytmy routingu (LSR i DVR) wyznaczają najkrótsze trasy w sieci. Obydwa rozwiązania zapewniają również uzyskanie **spójności** sieci, czyli stanu, w którym wszystkie przełączniki pakietów dysponują poprawnymi informacjami o jej budowie. Nie wyklucza to jednak pewnych problemów. Jeśli jeden z komunikatów LSR zostanie utracony, dwa przełączniki pakietów mogą wyznaczyć różne trasy jako najkrótsze. Problemy charakterystyczne dla mechanizmu DVR mają znacznie poważniejsze konsekwencje. Uszkodzenie łącza między węzłami może bowiem spowodować powstanie **pętli routingu**, w której wyniku pakiety bezustannie krążą między przełącznikami.

Jedną z głównych przyczyn niewłaściwego działania protokołów DVR jest to, że przełączniki pakietów otrzymują informacje o trasach, które same wysłały. Oto przykład. Założymy, że jedno z urządzeń rozeszło do swoich sąsiadów informację „znam trasę do celu  $D_1$  o koszcie 3”. Gdy połączenie prowadzące do celu  $D_1$  zostanie przerwane, przełącznik usunie wpis o  $D_1$  z tablicy przekazywania (lub oznaczy go jako błędny). Urządzenia sąsiednie zostały jednak wcześniej poinformowane o istnieniu trasy. Jeśli więc krótko po wystąpieniu awarii jeden z przełączników sąsiednich prześle komunikat z informacją „znam trasę do celu  $D_1$  o koszcie 4”, wiadomość zostanie uznana za wiarygodną, co z kolei doprowadzi do powstania pętli routingu.

Większość praktyczne wykorzystywanych mechanizmów routingu uwzględnia zabezpieczenia przed powstawaniem pętli routingu. W rozwiązaniu DVR stosuje się technikę **dzielonego horyzontu**, zgodnie z którą przełącznik nie odsyła informacji do węzłów, od których tę informację otrzymał. Ponadto w większości systemów wprowadza się histerezę, która nie pozwala na zbyt częste modyfikowanie informacji o routingu. Jednak mimo wszystko w dużych sieciach, w których wykorzystuje się wiele połączeń często ulegających awariom, problemy z routingu nadal mogą występować.

## 18.15. Podsumowanie

Technologie WAN znajdują zastosowanie w budowaniu sieci, które rozciągają się na dużych obszarach geograficznych. Tradycyjne systemy WAN składają się z urządzeń elektronicznych (nazywanych przełącznikami pakietów) połączonych ze sobą łączami telekomunikacyjnymi. Przełącznik pakietów składa się z procesora, pamięci operacyjnej i urządzeń wejścia-wyjścia. Interfejs jest z kolei przyłączany do komputera lokalnego lub innego przełącznika pakietów.

Sieci pakietowe działają zgodnie z zasadą „zapisz i przekaż”. Oznacza to, że nadchodzący pakiet jest zapisywany w pamięci przełącznika, skąd jest pobierany przez procesor w momencie, gdy można go przesłać do wybranego węzła docelowego. Dostarczanie pakietów do właściwych urządzeń wymaga utrzymywania specjalnej struktury danych nazywanej tablicą przekazywania. W tablicy tej zapisane są informacje o każdym węźle docelowym oraz o najbliższym węźle na trasie do określonego węzła docelowego. Nie ma w niej informacji o zwykłych komputerach.

Sieć WAN można przedstawić za pomocą grafu, w którym każdy węzeł odpowiada przełącznikowi pakietów, a każda krawędź reprezentuje łącze telekomunikacyjne. Odwzorowanie sieci za pomocą grafu okazuje się niezwykle użyteczne, ponieważ jest niezależne od szczegółów implementacyjnych sieci i pozwala na wypełnienie tablic przekazywania pakietów informacjami o trasach. Oprogramowanie zarządzające routingiem (przekazywaniem pakietów) działa zazwyczaj zgodnie z mechanizmami stanu łączca (LSR) lub wektorów odległości (DVR). W rozwiązaniu LSR każdy przełącznik pakietów dostarcza do wszystkich innych przełączników dane na temat stanu własnych łącz. Do wyznaczania najkrótszych tras jest wykorzystywany algorytm Dijkstry. Mechanizmy DSR wymuszają na przełącznikach pakietów okresowe rozsyłanie do węzłów sąsiednich listy zarejestrowanych węzłów docelowych wraz z kosztami dostarczania pakietów do danego węzła. Urządzenia sąsiednie analizują nadchodzące ogłoszenia i jeśli otrzymają informacje o trasach z niższymi kosztami, modyfikują zawartość tablicy przekazywania.

## ZADANIA

- 18.1. Z czego składa się typowy przełącznik pakietów? Do jakich łącz jest przyłączany?
- 18.2. Na jakie dwa elementy dzielony jest nowoczesny przełącznik pakietów?
- 18.3. Czy komputer może wykorzystać interfejs ethernetowy do komunikacji w sieci WAN? Uzasadnij odpowiedź.
- 18.4. Jaka jest minimalna liczba łącz cyfrowych potrzebna do połączenia  $N$  ośrodków w sieć WAN?
- 18.5. Wyjaśnij zasadę „zapisz i przekaż”.
- 18.6. Z jakich dwóch części składa się adres w sieci WAN?
- 18.7. Na rysunku 18.4 przedstawiono sposób przydzielu adresów komputerom przyłączonym do przełącznika pakietów. Założmy, że jeden z interfejsów przełącznika ulega uszkodzeniu i administrator przełącza komputer do innego nieużywanego interfejsu. Czy bieżąca konfiguracja zapewni poprawne działanie sieci? Uzasadnij odpowiedź.
- 18.8. Napisz program komputerowy, który pobierze jako dane wejściowe tablicę przekazywania oraz zbiór pakietów, a w wyniku swojego działania wygeneruje komunikaty o tym, w jaki sposób poszczególne pakiety zostałyby przekazane. Program powinien poprawnie obsługiwać również pakiety o niewłaściwym adresie.
- 18.9. Wyobraź sobie sieć WAN z dwoma przełącznikami pakietów. Każdy przełącznik zawiera w tablicy przekazywania wpisy odnoszące się do wszystkich adresów lokalnych (tzn. uwzględnia adresy wszystkich komputerów, które są przyłączone do przełącznika) oraz wpis o trasie domyślnej prowadzącej do drugiego przełącznika. W jakich przypadkach taka konfiguracja będzie działała poprawnie, a w jakich nie?
- 18.10. Jaka jest korzyść ze stosowania routingu dynamicznego?
- 18.11. Napisz program komputerowy, który będzie wyszukiwał najkrótszą trasę w grafie za pomocą algorytmu Dijkstry.
- 18.12. Wymień dwa podstawowe rozwiązania stosowane w rozproszonym wyznaczaniu tras. Opisz zasadę ich działania.

- 18.13. Programy działające w dwóch przełącznikach pakietów wymieniających informacje o routingu DVR muszą posługiwać się jednakowymi formatami komunikatów. Opracuj specyfikację takiego formatu. Podpowiedź: uwzględnij różnice w sposobie reprezentowania informacji w różnych komputerach.
- 18.14. Uzupełnij poprzednie zadanie o program, który będzie obsługiwał komunikaty o danym formacie. Nakłoń innego studenta do napisania analogicznego programu i sprawdź, czy wymiana informacji przebiega poprawnie.
- 18.15. Czy zawartość tablicy przekazywania jest zawsze zmieniana po odebraniu komunikatu DRV od węzła sąsiedniego? Uzasadnij odpowiedź.
- 18.16. Co to jest pętla routingu?



## *Zawartość rozdziału*

- 19.1. Wprowadzenie 345
- 19.2. Technologie łączy dostępowych 345
- 19.3. Technologie sieci LAN 347
- 19.4. Technologie sieci WAN 349
- 19.5. Podsumowanie 352

# 19

## *Technologie sieciowe — przeszłość i teraźniejszość*

### **19.1. Wprowadzenie**

W poprzednich rozdziałach przedstawiono najważniejsze rozwiązania z zakresu transmisji danych i komunikacji sieciowej. Uwzględniono w nich podział na technologie wykorzystywane w dostępie do internetu oraz te, które są stosowane w rdzeniu internetowym. Omówienie bazuje na klasycznym podziale sieci przewodowych i bezprzewodowych na systemy LAN, MAN i WAN.

Przez lata w każdej z grup pojawiło się wiele nowych technologii sieciowych. Niektóre z nich mimo początkowego sukcesu zostały zapomniane, inne znajdują zastosowanie do dzisiaj. Tematem tego rozdziału jest kilka najważniejszych rozwiązań, ich funkcje i cechy charakterystyczne. Towarzyszące opisom przykłady dają możliwość wyobrażenia sobie, jak wiele różnych technologii zostało opracowanych oraz jak często się one zmieniają.

### **19.2. Technologie łączy dostępowych**

W pierwszych rozdziałach tej części opisane zostały najważniejsze technologie łączy dostępowych (modemy DSL i modemy kablowe). W praktyce wykorzystywanych jest również wiele innych rozwiązań, w tym mechanizmy przekazywania danych w liniach energetycznych oraz systemu dostępu bezprzewodowego. W kolejnych punktach przedstawiono kilka najważniejszych technologii z tej grupy.

### 19.2.1. Synchroniczna sieć optyczna (SONET/SDH)

Sieci SONET (SDH) oraz związane z nimi mechanizmy TDM zostały opracowane z myślą o przenoszeniu cyfrowego sygnału rozmów telefonicznych. Stały się jednak także standardem obwodów cyfrowych w sieci szkieletowej internetu. Możliwość zbudowania systemu w formie pierścienia (w warstwie fizycznej) zwiększa jego niezawodność (wynikającą z istnienia nadmiarowych połączeń). Urządzenia sieci mogą bowiem wykrywać uszkodzenia i automatycznie je eliminować — nawet w przypadku uszkodzenia jednego z odcinków pierścienia dane są poprawnie przekazywane do jednostek systemu. Przyłączenie sieci lokalnej do pierścienia SONET (SDH) wymaga zastosowania specjalnego elementu nazywanego **multiplekserem add/drop**. Nazwa pochodzi od angielskich słów opisujących przeznaczenie multipleksera, czyli wprowadzanie lub kończenie obwodów cyfrowych, które są łączone z kolejnymi multiplekserami w pierścieniu. Do multipleksowania obwodów w ramach włókna światłowodowego stosuje się technikę zwielokrotniania w dziedzinie czasu. Dzięki hierarchicznemu podziałowi przepustowości kanałów istnieje możliwość wydzielenia w pierścieniu połączenia o określonej przepływności (na przykład obwodu E3).

### 19.2.2. Połączenia optyczne (OC)

Standardy OC definiują mechanizmy sygnalizacji stosowane w łączach optycznych pierścienia SONET (SDH). Odnoszą się one do połączeń o przepustowościach wyższych niż te, które są opisane w standardach T i E hierarchii SDH. Obwód OC może na przykład zostać wydzierżawiony przez firmę do połączenia jej dwóch ośrodków. Dostawcy usług internetowych (klasyfikowani jako dostawcy pierwszego poziomu) dysponują zazwyczaj obwodami OC-192 (10 Mb/s) oraz OC-768 (40 Mb/s) stanowiącymi szkielet internetu.

### 19.2.3. Cyfrowe łącza abonenckie (DSL) i modemy kablowe

Obydwie wymienione w tytule punktu technologie są obecnie podstawowym mechanizmem szerokopasmowego dostępu do internetu w małych firmach i domach odbiorców indywidualnych. Rozwiązania DSL zakładają wykorzystanie istniejących linii telefonicznych. Modemy kablowe z kolei bazują na istniejącej infrastrukturze telewizji kablowej. Szybkość transmisji danych w łączach DSL zawiera się w przedziale od 1 do 6 Mb/s i jest zależna od odległości między siedzibą abonenta a centralą telefoniczną. Przepustowość modemów kablowych wynosi nawet 52 Mb/s, ale pasmo jest współdzielone przez większą liczbę użytkowników. Obydwa rozwiązania są postrzegane jako przejściowe. Będą stosowane do czasu, gdy stanie się możliwe doprowadzanie do odbiorców łącz światłowodowych.

### 19.2.4. WiMAX i Wi-Fi

Nazwa Wi-Fi odnosi się do zbioru technologii transmisji bezprzewodowych, które zapewniają dostęp do internetu w domach, kawiarenkach, hotelach, terminalach lotniskowych i w innych lokalizacjach. Kolejne opracowywane standardy Wi-Fi zapewniają coraz większą przepustowość połączeń.

WiMAX jest rozwiązaniem, które dopiero zyskuje popularność. Pozwala na budowanie bezprzewodowych sieci MAN. Może być stosowane zarówno jako forma łączna dostępowego, jak i jako łącze szkieletowe<sup>52</sup>. Dwa profile działania pozwalają na obsługę stacjonarnych i mobilnych stacji końcowych.

### 19.2.5. Łącza satelitarne VSAT

Technologie VSAT zapewniają dostęp do internetu (za pomocą anteny satelitarnej o średnicy mniejszej niż 3 metry) za pośrednictwem satelitów indywidualnym gospodarstwom oraz niewielkim firmom. Łączność satelitarna gwarantuje duże przepustowości, ale nie-stety wiąże się również z dużymi opóźnieniami.

### 19.2.6. Komunikacja w sieciach energetycznych (PLC)

Komunikacja w sieciach energetycznych (PLC — ang. *Power Line Communication*) polega na wykorzystaniu wysokich częstotliwości do przekazywania danych w ramach istniejącej infrastruktury. Choć wymaga dalszych prac badawczych, już dziś znajduje szerokie zastosowanie w przyłączaniu abonentów do internetu.

## 19.3. Technologie sieci LAN

Wraz z pojawiением się idei sieci LAN kilka grup badawczych przedstawiło projekty takich rozwiązań oraz zbudowało pewne systemy prototypowe. Prace nad różnymi wersjami technologii LAN trwały przez dwadzieścia lat. Kilka rozwiązań zyskało w tym czasie dużą popularność, odnosząc jednocześnie sukces komercyjny. Ostatnio można jednak zaobserwować proces ujednolicenia systemów LAN. Nie należy się więc spodziewać powstania nowych rozwiązań z tej dziedziny.

### 19.3.1. Token Ring — sieć firmy IBM

Pierwsze prace nad rozwiązaniami LAN obejmowały badanie mechanizmu przenoszenia znaczników jako sposobu na sterowanie dostępem do medium transmisyjnego. Firma IBM wykorzystała wyniki tych badań do opracowania technologii sieci LAN nazywanej **Token Ring**. Pierwsza wersja standardu Token Ring zakładała komunikację z przepływnością 4 Mb/s. Konkurencją dla niej był standard Ethernet, umożliwiający transmisję z przepustowością 10 Mb/s. W późniejszym okresie firma IBM wprowadziła na rynek unowocześnioną wersję rozwiązań Token Ring o przepustowości 16 Mb/s. Mimo mniejszej szybkości transmisji danych oraz wyższych kosztów sieci Token Ring przez wiele lat stanowiły główny mechanizm wymiany informacji w działach informatycznych większości korporacji.

---

<sup>52</sup> Połączenie ze zdalnej lokalizacji lub łącze pomiędzy punktem dostępowym a centralą operatora telekomunikacyjnego.

### 19.3.2. FDDI i CDDI

W latach 80. ubiegłego wieku stało się jasne, że dwie najważniejsze wówczas technologie sieci LAN (Ethernet z przepustowością 10 Mb/s oraz Token Ring z przepustowością 16 Mb/s) nie gwarantują dostatecznej wydajności transmisji danych, aby spełnić coraz większe oczekiwania odbiorców. Rozpoczęto więc prace nad standardem FDDI (rozproszonej sieci danych opartej na łączach optycznych), który pozwalał na zwiększenie przepływności w sieciach LAN do 100 Mb/s. Inżynierowie przekonywali wówczas, że uzyskanie tak dużych szybkości transmisyjnych wymaga zastąpienia okablowania miedzianego łączami optycznymi, i sugerowali doprowadzanie włókna światłowodowego do każdej jednostki końcowej. Ponadto, w celu zapewniania niezawodności sieci zaproponowali instalowanie dwóch pierścieni o przeciwnym kierunku transmisji danych. Ewentualne przerwanie jednego pierścienia FDDI umożliwiało automatyczne zestawienie trasy zapasowej i ominięcie uszkodzonego fragmentu sieci. Wraz z technologią FDDI wprowadzono pierwsze przełączniki sieci LAN, które umożliwiały przyłączanie komputerów bezpośrednio do rdzeniowego systemu FDDI. Uzyskiwano w ten sposób fizyczną topologię gwiazdy oraz logiczną topologię pierścienia.

Dzięki największej przepustowości w transmisji danych oraz nadmiarowości gwarantującej niezawodność sieci technologia FDDI zyskała dużą popularność przede wszystkim jako sposób na łączenia komputerów w centrach obliczeniowych. Jednak wysoki koszt i konieczność zatrudnienia wysoko wykwalifikowanych pracowników do układania kabli światłowodowych sprawiły, że większość organizacji nie zdecydowała się na zastąpienie okablowania miedzianego włóknami optycznymi. Wraz z postępem prac nad szybkim Ethernetem twórcy rozwiązania FDDI zaproponowali zmodyfikowaną wersję systemu FDDI o nazwie CDDI, w którym połączenia światłowodowe zastąpiono kablami miedzianymi. Ostatecznie jednak, z uwagi na znacznie niższy koszt rozwiązań ethernetowych, technologia FDDI zniknęła z rynku.

### 19.3.3. Ethernet

Technologia Ethernet wygrała wyścig i zdominowała dzisiejszy rynek rozwiązań LAN. Spośród wszystkich instalacji LAN systemy ethernetowe stanowią przytaczającą większość. Tak naprawdę jednak Ethernet już dawno nie jest stosowany, ale został zastąpiony nowymi rozwiązaniami, które nadal są nazywane Ethernetem. Nietrudno przecież zauważać, że nie ma żadnego związku między grubymi kablami współosiowymi i sygnalizacją właściwą dla połączeń radiowych, którymi cechowały się pierwsze systemy Ethernet, a okablowaniem i sygnalizacją stosowanymi w dzisiejszych sieciach gigabitowych. Zmiana nie dotyczy tylko szybkości transmisji, ale obejmuje również fizyczną i logiczną topografię — kable zostały zastąpione koncentratorami, koncentratory przełącznikami ethernetowymi, a przełączniki ethernetowe przełącznikami VLAN.

## 19.4. Technologie sieci WAN

W skład rozwiązań WAN wchodzi wiele prac eksperymentalnych i produkcyjnych. Kilka z nich (prezentujących różnorodność technologii) zostało opisanych w dalszych punktach podrozdziału.

### 19.4.1. ARPANET

Pakietowe sieci WAN mają około pięćdziesiąt lat. W końcowce lat 60. ubiegłego stulecia powołano do działania Agencję Zaawansowanych Projektów Badawczych (ARPA), której celem było opracowanie mechanizmów pracy sieciowej dla Departamentu Obrony Stanów Zjednoczonych. Większa część pracy została poświęcona na ustalenie, czy technika przełączania pakietów będzie wartościowa dla wojska. Utworzona w wyniku tych działań sieć ARPANET stała się jedną z pierwszych pakietowych sieci WAN. System ARPANET łączył akademickie i przemysłowe ośrodki badawcze. Choć z dzisiejszej perspektywy połączenia w ramach sieci ARPANET były mało wydajne (linie dzierżawione łączące przełączniki pakietów umożliwiały transmisję danych z szybkością 56 kb/s), umożliwiały sprawdzenie koncepcji, algorytmów i terminologii, które są wykorzystywane do dzisiaj.

Wraz z rozpoczęciem projektu Internet, system ARPANET stał się rdzeniem sieci, który służył naukowcom do komunikacji i prowadzenia dalszych badań. W styczniu 1983 roku agencja ARPA nakazała wszystkim przyłączonym ośrodkom zaprzestanie korzystania z pierwotnych protokołów ARPANET i uruchomienie mechanizmów obsługi protokołów internetowych. Sama sieć ARPANET stała się więc pierwszym szkieletem internetu.

### 19.4.2. X.25

Jednym z pierwszych rozwiązań WAN był opracowany przez **Międzynarodową Unię Telekomunikacyjną** (ITU — ang. *International Telecommunications Union*) standard połączeń, wykorzystywany przez operatorów telekomunikacyjnych. Organizacja ITU działała w tym czasie pod nazwą **Międzynarodowy Komitet Doradczy do spraw Telefonii i Telegrafii** (CCITT — ang. *Consultative Committee for International Telephone and Telegraph*), dlatego opisywanie rozwiązania określa się często mianem standardu CCITT X.25. Technologia X.25 zyskała dużą popularność w Europie, a nieco mniejszą w Stanach Zjednoczonych.

Założenia specyfikacji są zgodne z klasyczną budową sieci WAN — sieć składa się z dwóch lub większej liczby przełączników pakietów, które są połączone ze sobą za pomocą łączy dzierżawionych. Komputery są przyłączane bezpośrednio do przełączników pakietów. W rozwiązaniach X.25 stosuje się zasadę komunikacji połączeniowej, charakterystyczną dla połączeń telefonicznych — komputer musi ustawić połączenie przed rozpoczęciem transmisji danych.

Ponieważ standard X.25 został opracowany przed upowszechnieniem się komputerów osobistych, wiele sieci X.25 służyło przede wszystkim do łączenia terminali ASCII ze zdalnymi komputerami wielozadaniowymi. Gdy użytkownik wprowadzał dane za pomocą

klawiatury, interfejs X.25 przechwytywał informacje o naciskaniu klawiszy, zapisywał je w pakietach X.25, a następnie przekazywał przez sieć do zdalnej jednostki. Analogicznie, dane wynikowe generowane przez komputer zdalny były przekazywane za pośrednictwem sieci X.25 do jednostki lokalnej, która z kolei wyświetlała je na ekranie użytkownika. Mimo promowania rozwiązań X.25 przez firmy telekomunikacyjne wysoki koszt i niska wydajność komunikacji spowodowały, że systemy X.25 zostały zastąpione przez inne technologie WAN.

#### 19.4.3. Frame Relay

Operatorzy rozległych sieci telekomunikacyjnych opracowali kilka technologii transportu danych na dużych odległościach. Jedną z nich jest technologia **Frame Relay**. Zakłada ona przekazywanie danych w blokach o maksymalnym rozmiarze 8 KB. Jedną z przyczyn dopuszczenia tak dużych rozmiarów ramek była chęć zastosowania sieci Frame Relay do łączenia segmentów LAN. Korporacja mieszcząca się w dwóch ośrodkach (w różnych miastach) mogła wykupić usługę Frame Relay w każdym z miast, a następnie wykorzystać sieć Frame Relay do przekazywania pakietów między sieciami LAN działającymi w dwóch lokalizacjach. Projektanci rozwiązania zastosowali zasadę komunikacji połączeniowej, która była akceptowalna dla organizacji o kilku siedzibach. System Frame Relay był więc bardzo popularny, aż do czasu opracowania tańszych mechanizmów.

Idea zastosowania technologii Frame Relay jako sposobu na łączenie sieci LAN oznaczała również konieczność zwiększenia przepustowości połączeń. Twórcy rozwiązania przewidywali, że ich transmisja danych będzie realizowana z przepływnością od 4 do 100 Mb/s (czyli z szybkością transmisji w ówczesnych sieciach LAN). W praktyce jednak wysoki koszt usługi Frame Relay sprawił, że odbiorcy byli zainteresowani połączeniami o niższej przepustowości (2 Mb/s lub 56 kb/s).

#### 19.4.4. SMDS

Podobnie jak Frame Relay, SMDS (usługa szybkiej transmisji pakietowej) jest usługą przekazywania danych na dużych odległościach oferowaną przez operatorów telekomunikacyjnych. Jej działanie bazuje na standardzie IEEE 802.6DQDB. Sam system jest uznawany za poprzednika standardu ATM. Mechanizm SMDS jest przeznaczony do przenoszenia danych, a nie sygnałów głosowych. Ponadto został zaprojektowany z myślą o dużych szybkościach transmisji. Jak wiadomo, transmisja nagłówków pakietów często zajmuje znaczną część dostępnego pasma. Aby zmniejszyć narzut transmisyjny, w rozwiązaniu SMDS zastosowano nagłówki o małym rozmiarze i ograniczono maksymalną długość pola danych do 9188 oktetów. Zdefiniowano również specjalny interfejs sprzętowy, który odpowiadał za przyłączanie komputerów do sieci. Interfejs ten umożliwiał dostarczanie danych do komputera z taką samą szybkością, z jaką komputer zapisywał je w pamięci operacyjnej.

Systemy SMDS działają przede wszystkim w sieciach o przepustowościach większych niż 1 Mb/s (większych niż w połączeniach Frame Relay). Technologie SMDS i Frame Relay różni również sposób wykorzystania. Transmisja SMDS ma charakter bezpołączeniowy,

co zapewnia jej większą elastyczność. Niemniej większość firm telekomunikacyjnych woli rozwiązania połączeniowe. Zakres zastosowań SMDS był więc bardzo ograniczony, co doprowadziło do wycofania mechanizmu z użytku.

#### 19.4.5. ATM

W reakcji na powstanie internetu przemysł telekomunikacyjny opracował standard ATM i nadał temu wydarzeniu wielki rozgłos. Wprowadzeniu systemów ATM na rynek w latach 90. towarzyszyło przekonanie, że zastąpią one wszystkie technologie WAN i LAN i doprowadzą do pełnej unifikacji systemów komunikacyjnych na świecie. Poza przekazywaniem danych rozwiązania ATM były przystosowane do transmisji sekwencji wizyjnych oraz głosu. Ponadto twórcy mechanizmu ogłosili, że nadaje się on do stosowania w sieciach o przepustowościach znacznie wyższych niż zapewniane przez jakiekolwiek obowiązujące wówczas technologie.

Nowym pomysłem (wprowadzonym wraz z ATM-em) było **przełączanie etykiet**. Standard ATM jest rozwiązaniem połączeniowym, jednak pakiety nie zawierają typowych adresów. W każdym z nich jest natomiast zapisany identyfikator nazywany **etykietą**. Wartość etykiety może być zmieniana przy każdym przejściu pakietu przez przełącznik. W chwili ustanowienia połączenia na każdym jego odcinku wybierane są etykiety opisujące dane połączenia, a następnie są one zapisywane w tablicy przełączania. Dzięki temu po odebraniu pakietu przełącznik wyszukuje w tablicy bieżącą etykietę i zastępuje ją nową wartością. W teorii przełączanie etykiet można zrealizować sprzętowo znacznie szybciej niż w przypadku klasycznego przełączania pakietów.

Aby zapewnić uniwersalność rozwiązania, twórcy standardu uwzględnili wiele dodatkowych funkcji ATM, włączając w to mechanizmy gwarantowania odpowiedniego poziomu obsługi w połączeniach między stacjami końcowymi (wyznaczanie minimalnej przepływności oraz maksymalnych opóźnień). Gdy rozpoczęła się faza wdrażania rozwiązania, inżynierowie uświadomili sobie, że mnogość funkcji znacznie zwiększa złożoność urządzeń i podnosi ich koszt. Co więcej, mechanizmy wyznaczania tras na podstawie zmiennych etykiet okazały się na tyle zawiłe, że ich nie stosowano. Standard ATM nie spełnił pokładanych w nim nadziei i zniknął z rynku.

#### 19.4.6. Wieloprotokołowe przełączanie etykiet (MPLS)

Rozwiązanie MPLS, mimo że nie jest standardem budowy sieci, bazuje na wynikach prac grupy zajmującej się systemami ATM — inżynierowie zaadaptowali mechanizm przełączania etykiet do budowy routerów internetowych<sup>53</sup>. Jednak zamiast całkowicie zastępować istniejące urządzenia (jak w przypadku ATM-u), wdrożenie technologii MPLS sprowadza się do uruchomienia dodatkowego oprogramowania. Routery MPLS odbierają pakiety internetowe, dodają do nich specjalne etykiety, a następnie przesyłają wzduż trasy MPLS, wykorzystując mechanizm przełączania na podstawie etykiet. W węzle docelowym etykiety są usuwane, a pakiet podlega standardowemu przetwarzaniu. Rozwiązanie

<sup>53</sup> Architektura internetu i zagadnienia związane z routingiem zostały opisane w rozdziale 20.

to jest szczególnie często wykorzystywane w rdzeniu internetu — dostawcy poziomu pierwszego stosują technikę MPLS do przesyłania pakietów określonymi trasami (klienci płacący więcej za usługę przekazywania danych mogą korzystać z krótszych tras, które są niedostępne dla odbiorców płacących mniejsze stawki abonamentowe).

#### 19.4.7. Sieć cyfrowa z integracją usług (ISDN)

Szczegółowy opis technologii ISDN znajduje się w rozdziale 12. (w tym punkcie wymienione zostały tylko najważniejsze cechy tego rozwiązania). Firmy telekomunikacyjne opracowały specyfikację ISDN z myślą o dostarczaniu usług sieciowych z większą przepływnością, niż było to możliwe w przypadku wykorzystania połączeń modemowych. Gdy standard wchodził na rynek, szybkość transmisji 128 kb/s wydawała się wystarczająca. Jednak z czasem okazało się, że za tę samą cenę można uzyskać znacznie większą szerokość pasma. Z tego powodu w większości regionów świata linie ISDN zostały zastąpione przez połączenia DSL, modemy kablowe oraz systemy komórkowe 3G. Wszystkie one oferują znacznie większą przepustowość.

### 19.5. Podsumowanie

Przez lata opracowano wiele technologii komunikacji sieciowej. Niektóre z nich okazały się zbyt skomplikowane, inne zbyt kosztowne, a jeszcze innym brakowało istotnych funkcji. Część została wycofana mimo początkowego sukcesu rynkowego. Nawet w przypadku Ethernetu, który przetrwał ponad trzydzieści lat, zachowana została jedynie nazwa i format ramki — mechanizmy transmisji zostały całkowicie zmienione.

## ZADANIA

- 19.1. Co to jest SONET?
- 19.2. Pod jaką nazwą jest znana użytkownikom technologia DOCSIS?
- 19.3. W którym rozwiązaniu należy się spodziewać mniejszych opóźnień: w VSAT czy w WiMAX? Uzasadnij odpowiedź.
- 19.4. Która firma opracowała technologię Token Ring?
- 19.5. Która technologia wyparła rozwiązania FDDI?
- 19.6. W której technologii wyeliminowano koncentratory ethernetowe?
- 19.7. Podaj nazwę technologii WAN, w której w 1983 roku zastosowano protokoły internetowe.
- 19.8. Jaka technologia WAN była stosowana w latach 80. w połączeniach bankowych?
- 19.9. Za co odpowiada standard ATM w komunikacji sieciowej?
- 19.10. Podaj nazwę technologii, która powstała na bazie ATM.
- 19.11. Dlaczego rozwiązania ISDN nie upowszechniły się na rynku telekomunikacyjnym?

# CZĘŚĆ IV

## Sieci TCP/IP

### Architektura internetu, adresacja, odwzorowanie adresów, enkapsulacja i protokoły

#### Rozdziały:

Rozdział 20. Internet — koncepcje, architektura i protokoły	355
Rozdział 21. IP — adresowanie w internecie	365
Rozdział 22. Przekazywanie datagramów	383
Rozdział 23. Protokoły i technologie uzupełniające	401
Rozdział 24. Przyszłość protokołu IP (IPv6)	425
Rozdział 25. UDP — usługa transportu datagramów	439
Rozdział 26. TCP — usługa niezawodnego transportu danych	449
Rozdział 27. Routing internetowy i protokoły routingu	469

# Zawartość rozdziału

- 20.1. Wprowadzenie 355
- 20.2. Przyczyny powstania internetu 355
- 20.3. Idea jednolitych usług 356
- 20.4. Jednolite usługi w heterogenicznym świecie 356
- 20.5. Internet 357
- 20.6. Fizyczne łączenie sieci za pomocą routerów 357
- 20.7. Architektura internetu 358
- 20.8. Wdrażanie jednolitych usług 359
- 20.9. Wirtualna sieć 359
- 20.10. Protokoły internetowe 361
- 20.11. Warstwy stosu TCP/IP 361
- 20.12. Stacje sieciowe, routery i warstwy protokołów 362
- 20.13. Podsumowanie 362

# 20

## *Internet — koncepcje, architektura i protokoły*

### **20.1. Wprowadzenie**

W poprzednich rozdziałach opisane zostały podstawy funkcjonowania sieci, w tym komponenty sprzętowe stosowane w systemach LAN i WAN oraz ogólne mechanizmy takie jak adresowanie i routing. Tematem tego rozdziału jest kolejna fundamentalna koncepcja w dziedzinie komunikacji z użyciem komputerów — internet, czyli technologia umożliwiająca łączenie wielu fizycznych sieci w jeden wspólny system komunikacji. Opisane zostały tutaj przyczyny powstania takiego rozwiązania, sposób łączenia komponentów składowych sieci oraz konsekwencje powstania takiego rozwiązania. Pozostałe rozdziały tej części dostarczają szczegółowych informacji na temat każdego wzmiankowanego elementu technologii. Zawierają omówienie poszczególnych protokołów z uwzględnieniem sposobu wykorzystania opisanych wcześniej technik do zagwarantowania niezakłóconej i pozbawionej błędów wymiany informacji.

### **20.2. Przyczyny powstania internetu**

Każda technologia sieciowa spełnia pewne określone założenia. Na przykład rozwiązania LAN są przeznaczone do szybkiej wymiany danych na krótkich odległościach. Natomiast technologie WAN służą do zapewnienia komunikacji na dużych obszarach. W rezultacie

żadna technologia sieciowa nie spełnia wszystkich potrzeb.

Duże organizacje (o różnych potrzebach w zakresie komunikacji sieciowej) muszą utrzymywać wiele sieci fizycznych. Ponadto, jeśli zdecydują się na wdrożenie rozwiązań właściwych do realizacji określonych zadań, może się okazać, że będą zarządzać wieloma sieciami wykonanymi w różnych technologiach. Na przykład Ethernet może się okazać najlepszym sposobem na połączenie komputerów w danym ośrodku, ale samo połączenie ośrodka z budynkiem firmowym zlokalizowanym w innej części miasta może wymagać zastosowania łączy dzierżawionych.

{}

### 20.3. Idea jednolitych usług

Najważniejszy problem w komunikacji sieciowej wydaje się oczywisty — komputery przyłączone do jednej sieci mogą wymieniać informacje tylko z innymi komputerami tej samej sieci. Problem ten uwidocznił się w latach 70. ubiegłego wieku, gdy duże organizacje rozpoczęły budowę osobnych sieci. Każdy z systemów korporacyjnych był wówczas odizolowany od pozostałych. Komputery często były przyłączane do pojedynczej sieci, co oznaczało, że pracownicy musieli wybierać stacje robocze do wykonania poszczególnych zadań. Otrzymywali dostęp do kilku terminali i zmieniali komputery za każdym razem, gdy trzeba było przesyłać informację w ramach określonej sieci.

Trudno się dziwić, że użytkownicy opisanych systemów nie byli zadowoleni z takiego sposobu działania ani dostatecznie efektywni w pracy. Dlatego większość nowoczesnych systemów komunikacji umożliwia wymianę informacji między dowolnymi komputerami, analogicznie do systemu telefonii, który zapewnia połączenia między dowolnymi aparatami telefonicznymi. Idea ta, nazywana **jednolitą usługą**, jest fundamentem współczesnych sieci komputerowych. Dzięki jednolitym usługom użytkownik dowolnego komputera firmowego może przesyłać wiadomość lub zbiór danych do każdego innego użytkownika sieci. Co więcej, nie musi zmieniać systemu, wykonując inne zadanie — wszystkie informacje są dostępne na wszystkich jednostkach. Takie rozwiązanie z pewnością zwiększa produktywność przedsiębiorstwa. Podsumowując:

*Systemy komunikacyjne udostępniające jednolite usługi umożliwiają komunikację między dowolną parą komputerów.*

### 20.4. Jednolite usługi w heterogenicznym świecie

Czy dostępność jednolitych usług oznacza, że wszyscy muszą korzystać z jednego rozwiązania sieciowego? Czy jednolite usługi mogą być udostępniane w różnych sieciach, o różnych technologiach? Zbudowanie jednej sieci przez połączenie przewodami wielu sieci składowych jest niemożliwe ze względu na brak zgodności mechanizmów komunikacyjnych. Nie można również zastosować standardowych technik rozszerzania sieci,

takich jak instalowanie mostów, ponieważ w każdej technologii obowiązuje inny format pakietu oraz inny sposób adresowania stacji. Ramki sformowanej w jednej sieci nie należy więc przekazywać do sieci zrealizowanej na bazie innej technologii.

*Choć wdrożenie jednolitych usług byłoby bardzo pożądane, niezgodności między urządzeniami, ramkami i mechanizmami adresowania uniemożliwiają dołączenie za pomocą mostów segmentów wykonanych w różnych technologiach.*

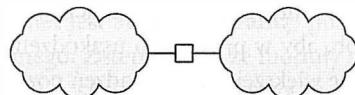
## 20.5. Internet

Pomimo niezgodności między poszczególnymi rozwiązaniami inżynierowie opracowali system, który zapewnia jednolity mechanizm wymiany informacji w sieciach heterogenicznych. Rozwiązanie odnosi się zarówno do sprzętu, jak i oprogramowania. Połączenia między poszczególnymi sieciami fizycznymi są realizowane za pomocą dodatkowych komponentów sprzętowych. Z kolei zainstalowane na stacjach sieciowych oprogramowanie zapewnia dostęp do wspomnianych wcześniej jednolitych usług. Wynikowy system, składający się z połączonych ze sobą sieci fizycznych, nazywa się **internetem**.

Pojęcie pracy internetowej ma dość ogólne znaczenie. Internet nie jest bowiem określonego rozmiaru — internet może się składać z kilku połączonych ze sobą sieci, może również być postrzegany jako ogólnoswiatowy Internet skupiający dziesiątki tysięcy pojedynczych systemów. Różna jest także liczba komputerów przyłączanych do każdej sieci składowej — w niektórych sieciach nie ma żadnych jednostek, w innych są ich setki.

## 20.6. Fizyczne łączenie sieci za pomocą routerów

Podstawowym komponentem sprzętowym służącym do łączenia heterogenicznych sieci jest **router**. Router jest odrębnym urządzeniem, którego jedyne zadanie polega pośredniczeniu w wymianie danych między sieciami. Podobnie jak most, składa się z procesora, pamięci operacyjnej oraz oddzielnego interfejsu wejścia-wyjścia w każdej sieci, do której jest przyłączony. Od strony sieci przyłączenie routera nie różni się niczym od przyłączenia komputera. Nie jest więc szczególnie skomplikowane. Obrazowo połączenie dwóch sieci przedstawiono na rysunku 20.1.



Rysunek 20.1. Połączenie dwóch sieci fizycznych za pomocą routera, który dysponuje dwoma niezależnymi interfejsami.

Komputery są przyłączane w tradycyjny sposób do każdej z sieci składowych

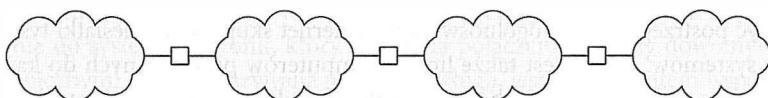
Same sieci zostały przedstawione w formie chmur, ponieważ połączenia routera nie zależą od technologii określonego segmentu. Router może łączyć ze sobą dwie sieci LAN, sieć LAN z siecią WAN lub dwie sieci WAN. Sieci z danej kategorii nie muszą być wykonane w tej samej technologii. Na przykład użycie routera pozwala na połączenie sieci Ethernet i Wi-Fi. Każda chmura obrazuje więc sieć o dowolnej technologii wykonania.

Podsumowując:

*Router internetowy jest urządzeniem specjalnego przeznaczenia, które zapewnia łączenie sieci. Może pośredniczyć w wymianie danych między systemami wykonanymi w różnych technologiach, o różnych mediach, schematach adresowania i formatach ramek.*

## 20.7. Architektura internetu

Dzięki routерom organizacje mogą dobierać technologie sieciowe do swoich potrzeb, a jednocześnie mają pewność, że wszystkie ich systemy zostaną połączone za pomocą internetu. Na rysunku 20.2 przedstawiono przykład zastosowania trzech routerów do połączenia czterech sieci fizycznych w internet.



Rysunek 20.2. Internet zbudowany z trzech routerek łączących cztery sieci fizyczne

Choć na rysunku przedstawiono każdy z routerek jako element o dwóch połączeniach, faktycznie produkowane urządzenia mogą łączyć znacznie więcej sieci. Nic nie stoi na przeszkodzie, aby do połączenia czterech przykładowych sieci został wykorzystany tylko jeden router. Mimo łatwości wdrożenia firmy starają się unikać konfiguracji bazujących na pojedynczych routeraх z dwóch powodów:

- Ponieważ działanie routera polega na przekazywaniu pakietów, procesor pojedynczego urządzenia może się okazać dość wydajny, aby zagwarantować obsługę większej liczby sieci.
- Nadmiarowość zwiększa niezawodność sieci. Oprogramowanie obsługujące protokoły routingu nieustannie monitoruje połączenia internetowe i steruje pracą routerek w taki sposób, aby w przypadku uszkodzenia sieci wybierały trasy alternatywne. Zainstalowanie większej liczby urządzeń pozwala uniknąć sytuacji, w której działanie sieci może zostać zakłócone po awarii pojedynczego elementu.

Planując sieć internetową, organizacja musi więc wybrać rozwiązanie, które zapewni odpowiednią niezawodność systemu, właściwą wydajność i akceptowalny koszt. W praktyce oznacza to dostosowanie topologii połączeń do przepustowości sieci składowych,

spodziewanego natężenia ruchu, zakładanego poziomu niezawodności oraz kosztu i wydajności samego routera. Podsumowując:

*Internet składa się z wielu sieci połączonych ze sobą routerami. Zasada działania internetu nie zależy od liczby oraz rodzaju sieci, liczby routerów łączących sieci oraz topologii połączeń międzysieciowych.*

## 20.8. Wdrażanie jednolitych usług

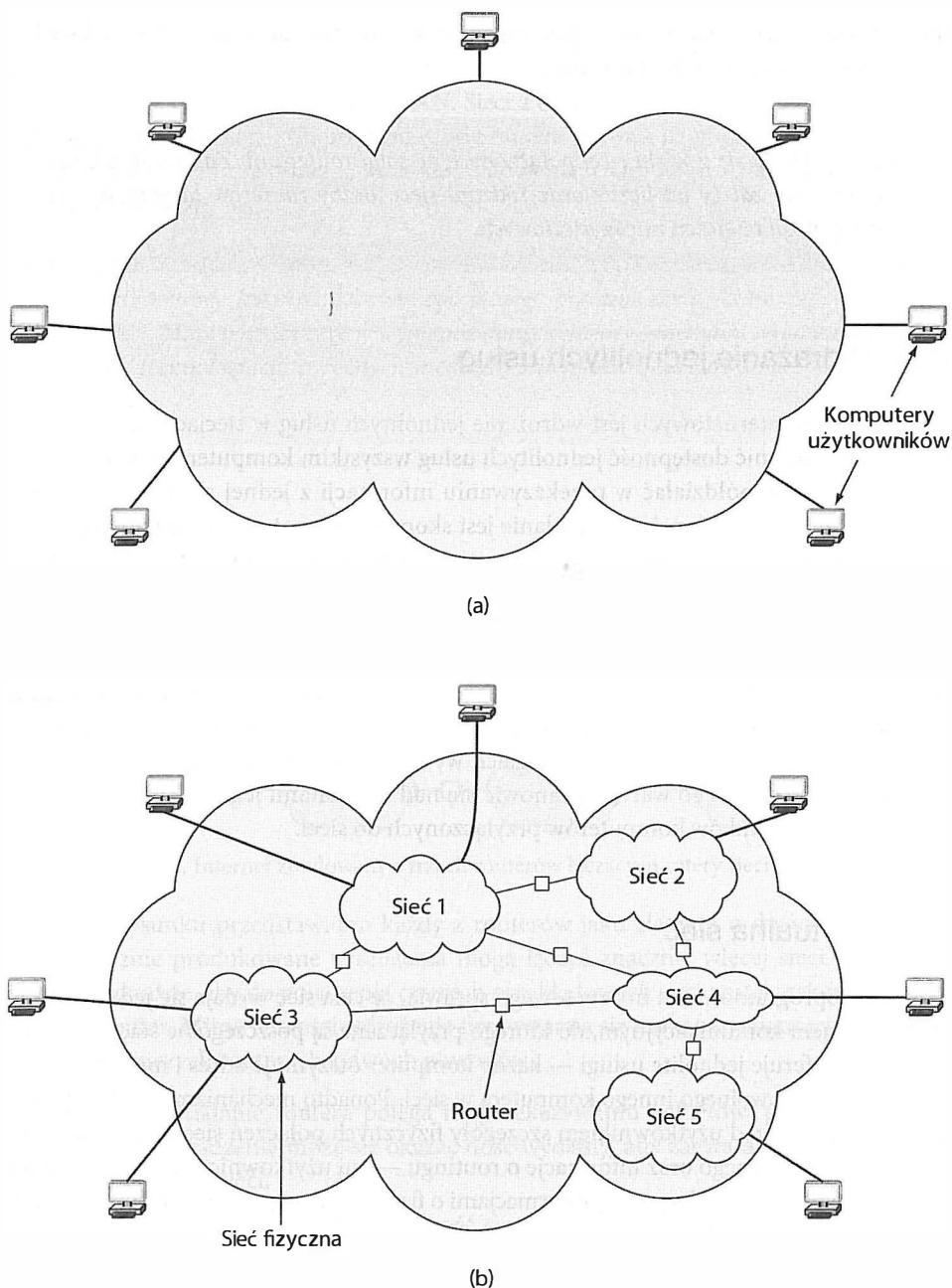
Celem działań internetowych jest wdrożenie jednolitych usług w sieciach heterogenicznych. Aby zapewnić dostępność jednolitych usług wszystkim komputerom w internecie, routery muszą współdziałać w przekazywaniu informacji z jednej sieci do określonej jednostki docelowej w innej sieci. Zadanie jest skomplikowane, ponieważ w różnych sieciach stosowane są różne formaty ramek i systemy adresowania. Konieczne jest więc instalowanie w komputerach i routerach oprogramowania, które obsługuje odpowiednie protokoły.

Oprogramowanie protokołu internetowego zostało szczegółowo opisane w dalszych rozdziałach książki. Omówienie zawiera informacje na temat rozwiązywania problemów wynikających z różnic w formatach ramek i mechanizmach adresowania (charakterystycznych dla sieci o różnych technologiach wykonania). Jednak przed analizą samego protokołu internetowego warto zastanowić się nad rezultatami jego działania z punktu widzenia użytkowników komputerów przyłączonych do sieci.

## 20.9. Wirtualna sieć

Działanie oprogramowania internetowego sprawia, że cała sieć wydaje się jednym, spójnym systemem komunikacyjnym, do którego przyłączane są poszczególne stacje robocze. System ten oferuje jednolite usługi — każdy komputer otrzymuje adres i może przesyłać informacje do dowolnego innego komputera w sieci. Ponadto mechanizmy obsługi protokołu ukrywają przed użytkownikiem szczegóły fizycznych połączeń sieciowych, schemat adresowania fizycznego oraz informacje o routingu — ani użytkownicy, ani uruchamiane przez nich aplikacje nie dysponują informacjami o fizycznym połączeniu sieci lub routerach pośredniczących w wymianie danych.

Często mówi się, że internet jest **siecią wirtualną**, ponieważ stanowi pewne uogólnienie systemu komunikacyjnego. Zatem mimo że moduły sprzętowe i oprogramowanie dostarczają iluzji ujednoliconego systemu sieciowego, w rzeczywistości taki system nie istnieje. Idea sieci wirtualnej oraz odpowiadająca jej struktura fizyczna zostały przedstawione na rysunku 20.3.



Rysunek 20.3. Internet. Złudzenie pracy użytkowników i aplikacji w jednej sieci (a) oraz fizyczna struktura połączonych przez routery sieci (b)

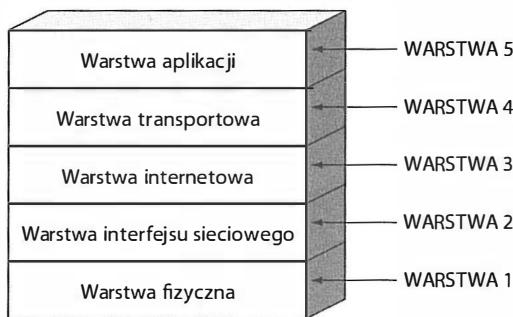
## 20.10. Protokoły internetowe

Choć zaproponowano wiele protokołów przeznaczonych do obsługi rozwiązań internetowych, obecnie powszechnie wykorzystuje się tylko jeden ich zestaw. Zestaw ten nazywa się formalnie **protokolami internetowymi TCP/IP**, ale większość osób zajmujących się sieciami nazywa go **stosem TCP/IP**<sup>54</sup>.

Stos TCP/IP został opracowany w tym samym czasie, w którym powstała globalna sieć Internet. Tak naprawdę, propozycja rozwiązań TCP/IP została przedstawiona przez tę samą grupę osób, które pracowały nad architekturą sieci Internet. Prace nad nimi rozpoczęły się w latach 70., czyli w czasie powstawania sieci lokalnych, i były kontynuowane do początku lat 90., gdy Internet stał się produktem komercyjnym.

## 20.11. Warstwy stosu TCP/IP

Z rozdziału 1. wiadomo, że protokoły internetowe wyznaczają pięciowarstwowy model odniesienia (przedstawiony na rysunku 20.4).



Rysunek 20.4. Pięciowarstwowy model odniesienia TCP/IP

Trzy spośród wymienionych warstw zostały opisane we wcześniejszych rozdziałach — omówienie warstwy aplikacji znajduje się w części I książki, natomiast części II i III odnoszą się do protokołów warstw 1. i 2. Dwie pozostałe warstwy są opisane w rozdziałach tej części.

### Warstwa 3. — internetowa

Warstwa 3. (IP) definiuje format pakietów przekazywanych w internecie oraz mechanizmy służące do transportu pakietów z określonej stacji przez jeden router lub kilka routerów do jednostki docelowej.

### Warstwa 4. — transportowa

Warstwa 4. (TCP) określa komunikaty i procedury gwarantujące niezawodność transferu danych.

<sup>54</sup> Nazwa pochodzi od akronimów TCP i IP, które odpowiadają dwóm najważniejszym protokołom ze stosu.

Podsumowując:

*Protokoły internetowe są podzielone na pięć teoretycznych warstw, z których trzecia należy do protokołu IP, a czwarta do protokołu TCP.*

## 20.12. Stacje sieciowe, routery i warstwy protokołów

Urządzenia przyłączane do internetu (w których działają aplikacje) są często nazywane **stacjami sieciowymi**. Jednostką taką może być zarówno telefon komórkowy, jak i komputer typu mainframe. Procesor stacji sieciowej może mieć małą lub dużą moc obliczeniową, pamięć może mieć mały lub duży rozmiar, a interfejsy umożliwiające przyłączenie urządzenia do sieci mogą pracować z małą przepustowością lub dużą. Protokoły TCP/IP zapewniają komunikację między jednostkami w sieci niezależnie od różnic sprzętowych.

Zarówno stacje sieciowe, jak i routery wymagają oprogramowania stosu protokołów TCP/IP. Routery nie operują jednak protokołami wszystkich warstw. Szczególnie dotyczy to protokołów warstwy 5., ponieważ routery nie wykonują standardowych aplikacji, takich jak transfer plików *ftp*<sup>55</sup>. Więcej informacji na temat oprogramowania protokołów TCP/IP oraz przeznaczenia poszczególnych warstw stosu internetowego zostało zamieszczonych w kolejnych rozdziałach.

## 20.13. Podsumowanie

Pozornie internet wydaje się jednym, spójnym systemem komunikacyjnym. Dowolna para komputerów przyłączonych do internetu może wymieniać dane tak, jakby pracowały w jednej sieci. Oznacza to, że każda stacja robocza może wysyłać informacje do innej jednostki przyłączonej do internetu. W ujęciu fizycznym internet jest zbiorem sieci połączonych ze sobą za pomocą **ruterów**. Każdy router jest urządzeniem specjalnego przeznaczenia, które pośredniczy w przekazywaniu pakietów między dwoma sieciami lub większą liczbą sieci.

Komputery przyłączone do internetu są nazywane **stacjami sieciowymi**. Mogą to być wydajne jednostki (na przykład superkomputery) lub urządzenia o małej mocy obliczeniowej (na przykład telefony komórkowe). Każda stacja jest przyłączona do jednej z fizycznych sieci internetu.

Iluzję pracy w jednym systemie komunikacyjnym zapewnia oprogramowanie protokołów internetowych, które musi być uruchomione w każdym komputerze i routerze działającym w sieci. Oprogramowanie to ukrywa przed użytkownikiem działanie urządzeń warstwy fizycznej i zapewnia dostarczanie wszystkich pakietów do ich jednostek docelowych.

<sup>55</sup> Choć w niektórych routeraх jest uruchamiane specjalne oprogramowanie, które umożliwia zdalne zarządzanie jego pracą.

Najważniejsze protokoły zapewniające komunikację internetową są znane pod nazwą **stosu protokołów TCP/IP**. Rozwiązania z tej grupy są od wielu lat wykorzystywane w sieciach prywatnych, a także w globalnej sieci Internet.

## ZADANIA

- 20.1. Czy internet zostanie zastąpiony przez pojedynczą technologię sieciową? Uzasadnij odpowiedź.
- 20.2. Jaka jest główna trudność we wdrożeniu jednolitych usług?
- 20.3. Wymień dwa powody, dla których firmy nie wykorzystują pojedynczych routerów do łączenia wszystkich sieci korporacyjnych.
- 20.4. Jeśli dany router może zostać przyłączony maksymalnie do  $K$  sieci, ile routerów ( $R$ ) jest potrzebnych do połączenia  $N$  sieci? Zapisz równanie, które wyrazi  $R$  względem  $N$  i  $K$ .
- 20.5. Użytkownicy postrzegają internet jako pojedynczą sieć. Jaka jest rzeczywistość? Do czego przyłączony jest komputer użytkownika?
- 20.6. Jakie jest przeznaczenie poszczególnych warstw pięciowarstwowego modelu odniesienia, wykorzystawanego w opisie protokołów TCP/IP?

# Zawartość rozdziału

- 21.1. Wprowadzenie 365
- 21.2. Adresy wirtualnego internetu 365
- 21.3. Schemat adresowania IP 366
- 21.4. Hierarchia adresów IP 367
- 21.5. Klasy adresów IP 367
- 21.6. Notacja dziesiętna z kropkami 368
- 21.7. Podział przestrzeni adresowej 369
- 21.8. Organizacje zarządzające przydziałem adresów 370
- 21.9. Adresowanie bezklasowe i podsieci 370
- 21.10. Maski adresów 371
- 21.11. Notacja CIDR 373
- 21.12. Przykład notacji CIDR 374
- 21.13. Adresy stacji w notacji CIDR 375
- 21.14. Adresy IP o specjalnym znaczeniu 375
- 21.15. Zestawienie adresów IP o specjalnym znaczeniu 378
- 21.16. Adres rozgłoszeniowy w formacie Berkeley 378
- 21.17. Routery i zasady adresowania IP 379
- 21.18. Stacje o wielu interfejsach sieciowych 380
- 21.19. Podsumowanie 380

# 21

## IP — adresowanie w internecie

### 21.1. Wprowadzenie

Tematem poprzedniego rozdziału była fizyczna architektura internetu, uwzględniająca routery pośredniczące w wymianie danych między sieciami. Ten rozdział jest pierwszym z grupy rozdziałów poświęconych oprogramowaniu protokołów, dzięki którym internet sprawia wrażenie jednolitego systemu komunikacyjnego. Opisano tutaj mechanizmy adresowania **protookołu internetowego** (IPv4 — ang. *Internet Protocol* wersja 4) oraz przeznaczenie masek podsieci<sup>56</sup>.

Omówienie protokołu IP jest kontynuowane w kolejnych rozdziałach. Każdy z nich opisuje jeden aspekt funkcjonowania protokołu. Razem rozdziały te dostarczają pełnych informacji na temat protokołu IP oraz zasad wykorzystywania go do przesyłania pakietów między komputerami w internecie.

### 21.2. Adresy wirtualnego internetu

Z rozdziału 20. wiadomo, że celem internetu jest zapewnienie jednolitego systemu komunikacji. Aby do tego doprowadzić, oprogramowanie protokołów musi przesyłać szczelesty implementacyjne sieci i zapewniać wrażenie pracy w jednej dużej sieci. Od strony aplikacji internet działa jak każda inna sieć — umożliwia komputerom wysyłanie i odbieranie pakietów. Główną różnicą między internetem a fizyczną siecią jest to, że jest on pewnym abstrakcyjnym systemem, wymyślonym przez jego projektantów i wytworzonym całkowicie przez oprogramowanie obsługujące protokoły internetowe. Projektanci rozwiązania wymyślili adresy, formaty pakietów oraz techniki przekazywania danych całkowicie niezależne od mechanizmów obowiązujących w samym systemie transmisyjnym.

---

<sup>56</sup> Jeśli nie zaznaczono inaczej, IP oznacza czwartą wersję protokołu internetowego.

Jednym z najważniejszych elementów tak zdefiniowanego internetu jest adres. Złudzenie pracy we wspólnej sieci jest możliwe tylko wtedy, gdy wszystkie stacje będą korzystały z jednego schematu adresowania oraz gdy każda jednostka będzie dysponować adresem o niepowtarzalnej wartości. Każdy komputer ma co prawda własny adres MAC, ale nie można na nim polegać, ponieważ w ramach internetu skupione są sieci wykonane w różnych technologiach, a każda z nich ma własną definicję adresu MAC.

Aby zapewnić jednolity system adresowania, w protokole IP zdefiniowano mechanizm, który jest niezależny od wykorzystawanego na niższych poziomach systemu komunikacyjnego. Adresy IP wyznaczają stacje docelowe w internecie podobnie, jak adresy MAC wskazują jednostki w sieciach LAN. Chcąc przesyłać pakiet przez internet, nadawca musi zapisać w nim adres IP stacji docelowej i przekazać do oprogramowania IP. W czasie przekazywania danych do komputera zdalnego docelowy adres IP jest wielokrotnie sprawdzany przez oprogramowanie jednostek pośredniczących w dostarczaniu informacji.

Zaletą adresowania IP jest jego jednorodność. Dzięki niemu dowolna para aplikacji może się ze sobą komunikować niezależnie od rodzaju urządzeń, z których sieć jest zbudowana, lub stosowanych w niej adresów MAC. Złudzenie pracy w jednolitym systemie jest tak przekonujące, że wiele osób zdziwieniem reaguje na informację o tym, że adresy IP nie są związane z fizyczną siecią, a za ich przydział odpowiada oprogramowanie protokołu IP.

Podsumowując:

*W celu zapewnienia jednolitego systemu adresowania w internecie opracowano mechanizm, który zapewnia przypisanie każdej jednostce niepowtarzalnego adresu IP służącego do komunikacji z innymi stacjami w sieci.*

### 21.3. Schemat adresowania IP

Zgodnie ze standardem IP każda jednostka otrzymuje niepowtarzalny 32-bitowy numer nazywany **adresem IP** (lub **adresem internetowym**<sup>57)</sup> stacji. Wysyłając pakiet, nadawca musi zawrzeć w nim własny adres IP (adres źródłowy) oraz adres jednostki odbiorczej (docelowy adres IP).

Podsumowując:

*Adres internetowy (adres IP) jest niepowtarzalną 32-bitową wartością binarną, przypisywaną stacji i wykorzystywaną w komunikacji z tą stacją.*

<sup>57</sup> Obydwu określeń można używać zamiennie.

## 21.4. Hierarchia adresów IP

Analogicznie do hierarchii adresów sieci WAN, każdy 32-bitowy adres IP jest podzielony na dwie części — prefiks i sufiks. Jednak zamiast identyfikowania przełącznika pakietów prefiks adresu IP wskazuje sieć składową, do której komputer został przyłączony. Sufiks z kolei reprezentuje określony komputer w danej sieci. Oznacza to, że każda sieć fizyczna dołączona do internetu dysponuje niepowtarzalnym **adresem sieci**, który jest zapisywany w formie prefiksu w adresach IP wszystkich komputerów należących do tej sieci. Ponadto każda stacja pracująca w danej sieci posiada własny niepowtarzalny sufiks.

Aby zagwarantować niepowtarzalność adresów, wystarczy zapewnić, że nie ma w internecie dwóch sieci o takich samych identyfikatorach oraz że w danej sieci nie występują dwie jednostki o identycznych sufiksach. Na przykład jeśli system składa się z trzech sieci fizycznych, poszczególnym sieciom składowym można przypisać identyfikatory 1, 2 i 3. Komputery przyłączone do pierwszej sieci mogą otrzymać sufiksy 1, 3 i 5, a jednostki z drugiej sieci mogą być opisane za pomocą wartości 1, 2 i 3. Identyfikatory stacji nie muszą być kolejnymi liczbami.

Opisany schemat adresowania ma dwie bardzo istotne cechy:

- Każdy komputer dysponuje niepowtarzalnym adresem (tj. dany adres nie może zostać przypisany do więcej niż jednej stacji).
- Choć nadawanie adresów sieci musi być koordynowane globalnie, wartości sufiksów można dobierać lokalnie (bez konieczności uzgadniania z innymi organizacjami).

Pierwsza własność wynika z tego, że w każdym adresie IP zawarta jest odpowiednia wartość prefiksu i sufiksu. Jeśli komputery należą do różnych sieci, prefiksy ich adresów muszą być różne. Natomiast jeśli pracują w tej samej sieci, odróżnia je wartości sufiksu. Dzięki temu adres przypisany do komputera jest niepowtarzalny.

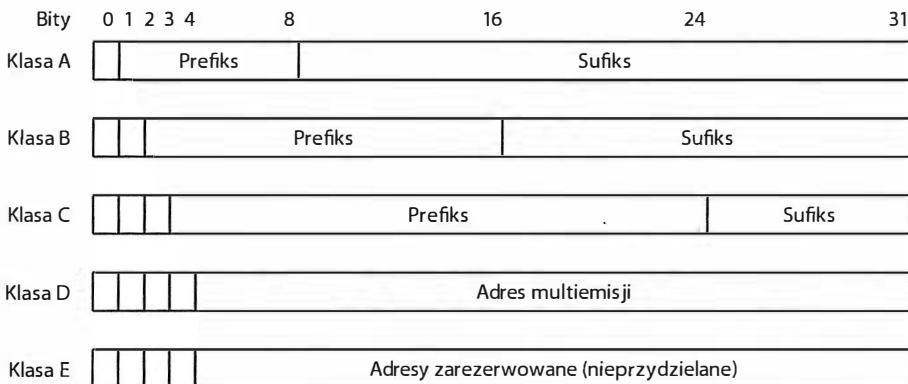
## 21.5. Klasy adresów IP

Po określeniu rozmiaru adresów IP i zdecydowaniu o podziale na dwie części projektanci protokołu IP musieli określić, ile bitów zostanie przeznaczonych na każdą z części. Prefiks musi się składać z takiej liczby bitów, która zagwarantuje niepowtarzalność identyfikatora sieci w internecie. Sufiks z kolei musi jednoznacznie wyróżniać jednostkę w danej sieci składowej. Zadanie okazuje się niezwykle trudne, ponieważ dodanie określonej liczby bitów do jednej części adresu powoduje odjęcie takiej samej liczby bitów z drugiej części. Wybranie prefiksów o dużym rozmiarze pozwala na utworzenie wielu sieci, ale ogranicza rozmiar każdej z nich. Zwiększenie rozmiaru sufiksu oznacza możliwość przyłączenia większej liczby komputerów, ale jednocześnie zmniejsza całkowitą liczbę sieci.

Dzięki uniezależnieniu internetu od technologii wykonania sieci składowych można w nim wyróżnić systemy o dużym zasięgu (których jest niewiele) oraz niewielkie sieci (występujące w dużej liczbie). Projektanci protokołu opracowali więc schemat adresowania, który uwzględnia zarówno małe, jak i duże systemy składowe. W pierwotnym mechanizmie,

nazywanym **klasowym adresowaniem IP** (ang. *classfull IP addressing*), przestrzeń adresowa została podzielona na trzy podstawowe **klasy**, zależnie od długości prefiksów i sufiksów.

O przynależności adresu do określonej klasy decydują cztery pierwsze bity. Na ich podstawie można również określić, jaki jest rozmiar prefiku oraz sufiksu. Na rysunku 21.1 przedstawiono pięć klas adresów, kombinacje bitowe odpowiadające poszczególnym klasom oraz rozmiary prefiku i sufiksu. Numeracja bitów jest zgodna ze specyfikacją protokołów TCP/IP — bity są ponumerowane kolejno od lewej do prawej strony, a bit o numerze zero występuje na pierwszej pozycji.



Rysunek 21.1. Pięć klas adresów IP pierwotnego, klasowego schematu adresowania

Choć klasowy system adresowania został wyparty przez inne rozwiązanie, adresy klasy D nadal są wykorzystywane w multiemisji (w technice dostarczania danych do grupy komputerów). Każdy adres multiemisji wyznacza grupę jednostek sieciowych. Jeśli taka grupa zostanie sformowana, wysłanie pojedynczego pakietu z adresem tej grupy powoduje przekazanie danych do wszystkich stacji należących do grupy. W praktyce globalne mechanizmy multiemisji nie zostały nigdy wdrożone. Niemniej rozwiązanie to można z powodzeniem stosować w ramach własnych sieci.

Podsumowując:

*Pierwotny system adresowania IP zakładał podział adresów na trzy klasy. Klasa D jest wciąż wykorzystywana w multiemisji, choć rozwiązanie to nie jest stosowane na skalę globalną.*

## 21.6. Notacja dziesiętna z kropkami

Jak wiadomo, adresy IP są 32-bitowymi identyfikatorami. Użytkownicy sieci nie muszą ich jednak wprowadzać lub przetwarzać w formacie binarnym. Oprogramowanie stacji sieciowych prezentuje je w formie, która jest znacznie łatwiejsza do zapamiętania przez

człowieka. Sposób zapisu adresów jest nazywany **notacją dziesiętną z kropkami** (ang. *dotted decimal notation*) i polega na przedstawieniu każdej 8-bitowej sekcji 32-bitowego adresu w formie liczby całkowitej, która jest oddzielona kropką od kolejnej liczby całkowitej. W tabeli 21.1 zaprezentowano kilka przykładów binarnych wartości adresu wraz z odpowiadającą im notacją dziesiętną.

**Tabela 21.1.** Przykłady 32-bitowych adresów zapisanych w formacie binarnym oraz w notacji dziesiętnej z kropkami

32-bitowa wartość binarna	Odpowiednik w notacji dziesiętnej z kropkami
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

W zapisie dziesiętnym z kropkami każdy **oktet** (8-bitowa wartość) jest interpretowany jako binarna liczba całkowita bez znaku<sup>58</sup>. W ostatnim z przedstawionych przykładów można zauważyć, że najmniejsza dopuszczalna wartość (0) składa się z samych zer logicznych. Natomiast największa wartość (255) odpowiada oktetowi złożonemu z samych jedynek logicznych. Zatem adresy w notacji dziesiętnej muszą zawierać się w przedziale od 0.0.0.0 do 255.255.255.255. Adresy multiemisji (pochodzące z klasy D) mają wartości z przedziału od 224.0.0.0 do 239.255.255.255.

Podsumowując:

*Notacja dziesiętna z kropkami jest sposobem prezentacji 32-bitowych wartości binarnych wykorzystywanym przez oprogramowanie w interakcjach z użytkownikiem. Zgodnie z założeniami każdy oktet jest przedstawiany jako liczba całkowita oddzielona kropką od kolejnego oktetu.*

## 21.7. Podział przestrzeni adresowej

Pierwotny schemat adresowania klasowego został opracowany przed pojawiением się na rynku komputerów PC, przed upowszechnieniem się sieci LAN oraz przed wdrożeniem sieci komputerowych w większości firm. Zgodnie z nim przestrzeń adresowa została podzielona na kilka zakresów o różnych rozmiarach. Nierównomierny podział miał zapew-

<sup>58</sup> W specyfikacji IP wykorzystywany jest termin **oktet**, a nie **bajt**, ponieważ rozmiar bajtu jest zależny od konkretnego komputera. Dlatego, mimo że 8-bitowe bajty są de-facto standardem, określenie „oktet” eliminuje wszelkie niejednoznaczności.

nić obsługę różnych scenariuszy implementacyjnych. Na przykład klasa A zajmuje połowę całej przestrzeni adresowej, mimo że umożliwia wydzielenie tylko 128 sieci. Celem takiego doboru zakresu wartości było umożliwienie kluczowym dostawcom usług internetowych tworzenia sieci złożonych z milionów komputerów. Z kolei powodem utworzenia klasa C była chęć zapewnienia adresów dla sieci firmowych, składających się z kilku komputerów. Maksymalna liczba sieci w każdej z klas oraz maksymalna liczba stacji w ramach każdej z sieci zostały przedstawione w tabeli 21.2.

Tabela 21.2. Liczba sieci oraz liczba komputerów w każdej z sieci wynikające z klas adresów IP

Klasa adresów	Liczba bitów w prefiksie	Maksymalna liczba sieci	Liczba bitów w sufiksie	Maksymalna liczba stacji w sieci
A	7	128	24	16 777 216
B	14	16 384	16	65 536
C	21	2 097 152	8	256

## 21.8. Organizacje zarządzające przydziałem adresów

Prefiksy przypisywane poszczególnym sieciom w internecie nie mogą się powtarzać. Powołano więc centralną organizację ICANN (ang. *Internet Corporation for Assigned Names and Numbers*), która zarządza przydziałem adresów i rozwiązywaniem sporów w tym zakresie. Nie zajmuje się jednak przydzielaniem poszczególnych prefiksów sieciowych. Upoważnia do tego specjalne **urzędy rejestracyjne** (ang. *registrar*), które z kolei udostępniają bloki adresów dostawcom usług internetowych. Aby uzyskać odpowiedni prefiks sieci, przedstawiciele firm zazwyczaj kontaktują się bezpośrednio z dostawcami usług internetowych<sup>59</sup>.

## 21.9. Adresowanie bezklasowe i podsieci

Rozwój internetu spowodował, że klasyczne adresowanie klasowe stało się nieefektywne. Wiele organizacji składało wnioski o przydział adresów klasy A lub B, chcąc zagwarantować sobie możliwość późniejszego rozwoju. W praktyce jednak znaczna część z przyznanych adresów nie była wykorzystywana. I mimo tego, że wiele adresów klasy C pozostało wolnych, tylko nieliczni chcieli z nich korzystać.

Chcąc wyeliminować opisane ograniczenia, opracowano dwa nowe mechanizmy przydziału adresów:

- dzielenie na podsieci,
- adresowanie bezklasowe.

<sup>59</sup> Procedura pozyskiwania przez komputer sufiksu adresu została opisana w rozdziale 23.

Są one jednak tak ściśle ze sobą powiązane, że można je rozpatrywać jako jedno rozwiązanie. Zamiast podziału na trzy klasy zaproponowano możliwość wskazania dowolnego bitu, który będzie stanowił granicę między prefiksem sieci i sufiksem stacji. Mechanizm wydzielania podsieci był początkowo stosowany w dużych korporacjach przyłączonych do globalnego internetu. Natomiast metoda adresowania bezklasowego wprowadziła tę technikę zarządzania adresami do samego internetu.

Aby zrozumieć, dlaczego możliwość dowolnego podziału adresu jest tak istotna, przeanalizujmy przykład dostawcy usług internetowych, który dysponuje pewną pulą adresów. Założymy, że jeden z klientów dostawcy żąda prefiksu dla sieci złożonej z trzydziestu pięciu jednostek. W przypadku zastosowania adresowania klasowego dostawca musiałby przydzielić odbiorcy prefiks klasy C. Do zapisania wszystkich możliwych kombinacji sufiksów stacji klienta wystarczą cztery bity. Oznacza to, że 219 spośród 254 dozwolonych wartości nigdy nie zostały przypisanych do żadnej stacji<sup>60</sup>. Innymi słowy, większa część przestrzeni adresowej klasy C została zmarnowana. Adresowanie bezklasowe okazuje się w tym względzie znacznie efektywniejsze. Umożliwiłoby bowiem dostawcy wykorzystanie prefiksu złożonego z 26 bitów. Sufiks składałby się wówczas z 6 bitów, a to oznacza zmarnowanie jedynie 27 adresów.

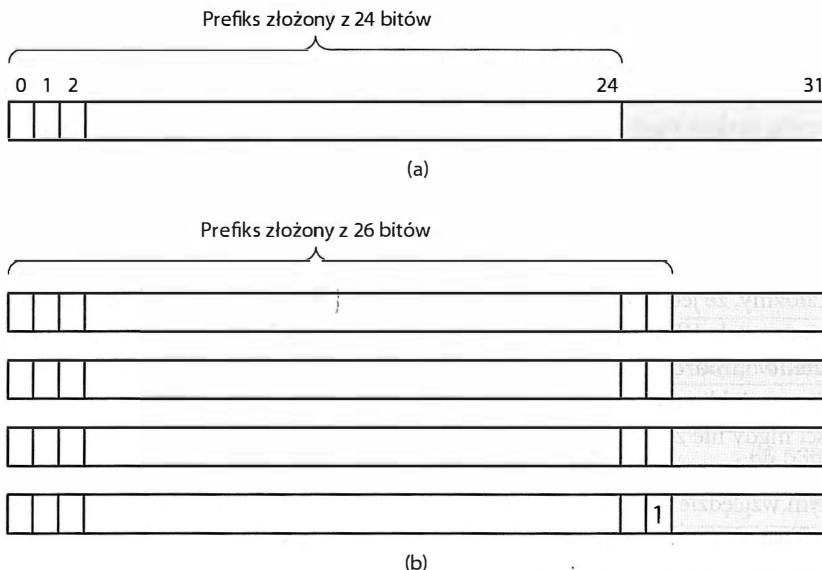
Innym spojrzeniem na problem może być rozważenie sytuacji, w której dostawca usług internetowych dysponuje jednym prefiksem klasy C. Technika adresowania klasowego spowodowałaby przydzielenie całej przestrzeni adresowej jednemu odbiorcy. Rozwiązań bezklasowe, dzięki wydłużeniu poszczególnych prefiksów, umożliwia natomiast podzielenie zakresu między większą liczbą klientów. Na rysunku 21.2 przedstawiono sposób postępowania, dzięki któremu dostawca usług może podzielić prefiks klasy C na cztery prefiksy o większej liczbie bitów, odpowiadające sieciom o maksymalnie sześćdziesięciu dwóch stacjach.

Bitы stacji w ramach każdego z prefiksów zostały zaznaczone na rysunku szarym kolorem. W klasie C sufiks ma rozmiar ośmiu bitów. Natomiast w zaproponowanym podziale bezklasowym składa się z sześciu bitów. Jeśli pierwotny prefiks był niepowtarzalny, każdy z prefiksów powstały z przesunięcia granicy bitowej również będzie niepowtarzalny. Dostawca usług internetowych może więc przydzielić powstałe prefiksy bezklasowe czterem klientom bez obaw o marnowanie adresów.

## 21.10. Maski adresów

Systemy adresowania bezklasowego oraz adresowania uwzględniającego podsieci wymagają od komputerów przetwarzających adresy przechowywania pewnej dodatkowej informacji — wartości, która określa granicę podziału między bitami sieci a bitami stacji. Aby wyznaczyć wspomnianą granicę, w specyfikacji IP przewidziano użycie 32-bitowego ciągu nazywanego **maską adresu** lub **maską podsieci**. Występujące w masce jedynki oznaczają prefiks sieci, natomiast zera wskazują część adresu stacji.

<sup>60</sup> Liczba 254 wynika z tego, że wśród 256 adresów klasy C dwie kombinacje (jedna złożona z samych zer i jedna zawierająca samej jedynki) są zarezerwowane na adres sieci i adres rozgłoszeniowy, co zostało opisane w dalszej części rozdziału.



Rysunek 21.2. Prefiks klasy C (a) oraz ten sam prefiks podzielony na cztery prefiksy bezklasowe (b)

Dlaczego informacja o granicy podziału jest przechowywana w formie maski bitowej? Użycie maski zwiększa bowiem wydajność przetwarzania. W dalszej części książki została opisana praca routera, która polega na nieustannym porównywaniu prefiksów sieci nadchodzących pakietów z wartościami zapisanymi w tablicy routingu. Dzięki zastosowaniu maski podsieci porównanie takie jest bardzo efektywne. Założymy na przykład, że router dysponuje adresem docelowym ( $D$ ), prefiksem sieci przedstawionym w formie 32-bitowej wartości binarnej ( $N$ ) oraz 32-bitową maską podsieci ( $M$ ). Założymy dalej, że wartość  $N$  przechowuje prefiks sieci na początkowych bitach, a pozostałe bity są zerami. Aby sprawdzić, czy adres docelowy wchodzi w skład określonej sieci, router wykonuje następujące porównanie:

$$N == (D \And M)$$

Zadanie sprowadza się do wykonania iloczynu logicznego maski z adresem  $D$  (co powoduje ustawienie bitów stacji na zera), a następnie porównania wyniku z prefiksem sieci  $N$ .

Oto konkretny przykład 32-bitowego prefiksu sieci:

10000000 00001010 00000000 00000000

który w zapisie dziesiętnym ma wartość 128.10.0.0. Przymijmy, że operacja jest wykonywana z użyciem 32-bitowej maski, w której 16 bardziej znaczących bitów ma wartość 1, a pozostałe są zerami (w zapisie dziesiętnym wartość maski to 255.255.0.0):

11111111 11111111 00000000 00000000

Założymy, że adres docelowy ma wartość 128.10.2.3, która odpowiada binarnej wartości:

10000000 00001010 00000010 00000011

Wykonanie iloczynu logicznego adresu docelowego i maski spowoduje wyodrębnienie sześciastu bardziej znaczących bitów. Wynikiem jest wówczas ciąg:

10000000 00001010 00000000 00000000

odpowiadający prefiksowi sieci 128.10.0.0.

## 21.11. Notacja CIDR

Adresowanie bezklasowe jest formalnie nazywane **bezklasowym routingiem międzymomenowym** (CIDR — ang. *Classless Interdomain Routing*). Nazwa jest jednak nieco myląca, ponieważ w praktyce mechanizm CIDR opisuje jedynie techniki adresowania i przekazywania pakietów. Celem opracowania techniki CIDR było ułatwienie użytkownikom definiowania masek podsieci. Przeanalizujmy więc działanie rozwiązania na przykładzie maski niezbędnej do wykonania zadania z rysunku 21.2b. Wymagałoby to zapisać masek dwudziestu sześciu jedynek oraz sześciu zer, a dziesiętna postać maski powinna być zgodna z poniższą wartością:

255.255.255.192

Aby ułatwić użytkownikom sieci interpretowanie wartości masek, wprowadzono zmiany w formacie zapisu dziesiętnego z kropkami. Nowa forma reprezentacji adresu jest znana jako **notacja CIDR**. Zgodnie z jej założeniami adres i maska mogą zostać zapisane w jednym ciągu. Za wartością adresu należy wówczas dopisać znak ukośnika oraz liczbę, która odpowiada sekwencji jedynek logicznych w masce. Bity jedynek logicznych maski muszą następować po sobie (nie mogą być przeplatane zerami) i muszą zajmować najstarsze pozycje bitowe. Ogólny format zapisu jest następujący:

ddd.ddd.ddd.ddd/m

Bloki *ddd* reprezentują w nim dziesiętne wartości kolejnych oktetów adresu, a *m* jest liczbą jedynek w masce. Zatem zapis:

192.5.48.69/26

oznacza, że maska składa się z 26 bitów o wartości 1. W tabeli 21.3 przedstawiono maski adresów w notacji CIDR oraz odpowiadające im wartości dziesiętne. Warto zauważyć, że niektóre z masek odpowiadają pierwotnemu klasowemu podziałowi adresów.

Tabela 21.3. Lista masek podsieci w notacji CIDR oraz w zapisie dziesiętnym z kropkami

Długość (CIDR)	Maska adresu	Uwagi
/0	0.0.0.0	Same zera (odpowiednik braku maski)
/1	128.0.0.0	
/2	192.0.0.0	
/3	224.0.0.0	
/4	240.0.0.0	
/5	248.0.0.0	
/6	252.0.0.0	

Tabela 21.3. Lista masek podsieci w notacji CIDR oraz w zapisie dziesiętnym z kropkami — ciąg dalszy

Długość (CIDR)	Maska adresu	Uwagi
/7	254.0.0.0	
/8	255.0.0.0	Maska klasy A
/9	255.128.0.0	
/10	255.192.0.0	
/11	255.224.0.0	
/12	255.240.0.0	
/13	255.248.0.0	
/14	255.252.0.0	
/15	255.254.0.0	
/16	255.255.0.0	Maska klasy B
/17	255.255.128.0	
/18	255.255.192.0	
/19	255.255.224.0	
/20	255.255.240.0	
/21	255.255.248.0	
/22	255.255.252.0	
/23	255.255.254.0	
/24	255.255.255.0	Maska klasy C
/25	255.255.255.128	
/26	255.255.255.192	
/27	255.255.255.224	
/28	255.255.255.240	
/29	255.255.255.248	
/30	255.255.255.252	
/31	255.255.255.254	
/32	255.255.255.255	Same jedynki (maska typowa dla stacji sieciowej)

## 21.12. Przykład notacji CIDR

Przeanalizujmy przykład notacji CIDR wykorzystywany przez dostawcę usług internetowych, który dysponuje następującą przestrzenią adresową:

128.211.0.0/16

Przyjmijmy, że dostawca ma dwóch klientów, z których jeden potrzebuje dwunastu adresów, a drugi dziewięciu. Prefiks pierwszego klienta może mieć wówczas wartość:

128.211.0.16/28

a drugiego:

128.211.0.32/28

Maski obydwu klientów mają takie same rozmiary (28 bitów), ale prefiksy są różne.

Adres przydzielony pierwszemu odbiorcy ma postać binarną:

10000000 11010011 00000000 0001 0000

Natomiast wartość przydzielona drugiemu klientowi to:

10000000 11010011 00000000 0010 0000

Nie ma między nimi niejednoznaczności. Każdy odbiorca otrzymał niepowtarzalny prefiks. Dodatkowo dostawca usług internetowych zachował większą część pierwotnej przestrzeni adresowej, którą może wykorzystać, przydzielając adresy innym klientom.

## 21.13. Adresy stacji w notacji CIDR

Zastanówmy się nad sposobem obliczenia zakresu adresów w określonym bloku CIDR. Gdy klient otrzyma prefiks CIDR od dostawcy usług internetowych, może go wykorzystać do nadania adresów stacjom we własnej sieci. Założymy na przykład, że dana organizacja otrzymała prefiks 128.211.0.16/28 (zgodnie z wcześniejszym opisem). Z rysunku 21.3 wynika, że może wykorzystać cztery bity do wyznaczenia adresów stacji. Na rysunku przedstawiono również najmniejszą i największą wartość bitową z przyznanego zakresu. Wykluczono jednak z przedziału adresy składające się z samych jedynek i z samych zer na bitach stacji.

Z rysunku 21.3 wynika pewna niedogodność stosowania adresowania bezklasowego — przez to, że granice przedziału identyfikatorów stacji mogą być różnie dobierane, dziesiątne wartości zakresu adresów nie są łatwe do zapamiętania. W przedstawionym przykładzie zestawienie prefiksu sieci z czternastoma dozwolonymi wartościami sufiku doprowadziło do wyznaczenia bloku adresów z przedziału od 128.211.0.17 do 128.211.0.30.

## 21.14. Adresy IP o specjalnym znaczeniu

Poza adresami przeznaczonymi dla komputerów wyznacza się pewne adresy, które opisują sieci lub grupy stacji. W specyfikacji IP adresy o specjalnym znaczeniu są adresami **zarezerwowanymi**. Oznacza to, że nie mogą zostać przypisane zwykłej stacji sieciowej. Składnia i znaczenie poszczególnych adresów specjalnych zostały opisane w kolejnych punktach podrozdziału.

0	Prefiks sieci 128.211.0.16/28														28	31
	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1
:																
0	Maska podsieci 255.255.255.240														28	31
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	Najmniejsza wartość adresu stacji 128.211.0.17														28	31
	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0
0	Największa wartość adresu stacji 128.211.0.30														28	31
	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1

Rysunek 21.3. Adresacja CIDR z wykorzystaniem 28-bitowego prefiksu

### 21.14.1. Adres sieci

Jedna z przyczyn zarezerwowania specjalnej kombinacji bitowej na adres sieci wynika z rysunku 21.3 — wyróżnienie szczególnej wartości, odpowiadającej prefiksowi skojarzonemu z daną siecią, ułatwia posługiwanie się adresami. Zgodnie ze standardem IP do oznaczenia sieci stosuje się adresy, w których bity stacji są ustalone na zero. Zatem adres 128.211.0.16/28 oznacza sieć, ponieważ wszystkie bity od 29. włącznie mają wartość zero. Adres sieci nigdy nie może występować w pakiecie jako adres docelowy<sup>61</sup>.

### 21.14.2. Adresy rozgłoszenia kierowanego

Często użyteczne okazuje się dostarczenie kopii pakietu do wszystkich stacji w danej sieci fizycznej. Aby ułatwić rozgłoszanie informacji, w specyfikacji IP wydzielono specjalny **adres rozgłoszenia kierowanego** do wybranej sieci. Zapisanie w pakiecie adresu docelowego odpowiadającego adresowi rozgłoszenia kierowanego powoduje, że pakiet jest przekazywany przez internet aż do routera obsługującego wskazaną sieć, a następnie zostaje rozesłany do wszystkich stacji przyłączonych do tej sieci.

Adres rozgłoszenia kierowanego jest tworzony przez dodanie sufiksu złożonego z samych jedynek do prefiksu opisującego sieć. Identyfikator stacji złożony z samych jedynek bitowych jest więc zarezerwowany do szczególnego wykorzystania. Jeśli administrator przez pomyłkę przypisze komputerowi adres z samymi jedynkami na bitach stacji, zainstalowane w komputerze oprogramowanie sieciowe może działać niepoprawnie.

<sup>61</sup> W podrozdziale 21.16 opisana została notacja adresów rozgłoszeniowych w formacie Berkeley, która jest wyjątkiem od tej reguły.

W jaki sposób wykonywane jest rozgłoszenie? Jeśli moduł sieciowy ma zaimplementowaną funkcję rozgłaszenia, wykorzystuje wewnętrzny mechanizm dostarczania rozgłoszeń. Jeśli jednak taki sposób działania nie jest obsługiwany sprzętowo, oprogramowanie musi rozesłać kopie każdego pakietu do wszystkich stacji w sieci.

### 21.14.3. Adres ograniczonego rozgłaszenia

Pojęcie **ograniczonego rozgłaszenia** odnosi się do rozgłoszenia wykonywanego w ramach bezpośrednio przyłączonej sieci. Taka forma emisji pakietów jest na przykład stosowana w chwili uruchamiania komputera, gdy nie ma on jeszcze informacji o adresie sieci.

W specyfikacji IP zarezerwowano na ten cel wartości składające się z trzydziestu dwóch jedynek. Wszystkie pakiety dostarczane pod adres złożony z samych jedynek są więc rozglaszane w sieci lokalnej.

### 21.14.4. Adres własny komputera

Jak wiadomo, każdy pakiet internetowy zawiera adres nadawcy oraz adres odbiorcy. Jednostka sieciowa musi więc znać własny adres IP, zanim rozpoczęcie wysyłanie lub odbieranie danych. W rozdziale 23. został opisany protokół (z rodziny TCP/IP), który umożliwia automatyczne pozyskanie adresu IP w czasie uruchamiania jednostki. Niestety, do wykonania tego zadania wykorzystywany jest protokół IP. Uruchamiający się komputer nie może więc użyć poprawnego adresu źródłowego. W takich przypadkach rozwiązaniem jest zastosowanie wartości złożonej z samych zer logicznych, która oznacza **niniejszą stację**.

### 21.14.5. Adres pętli zwrotnej

W specyfikacji IP zdefiniowano **adres pętli zwrotnej**, który służy do testowania aplikacji sieciowych. Programiści często korzystają z tego adresu we wstępnej fazie sprawdzania nowo utworzonego programu. Aby wykonać wspomniany test, trzeba dysponować dwiema aplikacjami, które docelowo będą komunikować się za pośrednictwem sieci. Każda z nich musi zawierać kod współdziałający z oprogramowaniem stosu protokołów TCP/IP. Programów nie należy jednak uruchamiać w oddzielnych komputerach. Powinny one działać w jednym systemie i wykorzystywać w komunikacji adres pętli zwrotnej. Przesyłanie danych z jednej aplikacji do drugiej sprowadza się wówczas do przekazania ich w ramach stosu protokołów do warstwy IP, która zwraca strumień i dostarcza informacje do stosu protokołów drugiego programu. Programista może w ten sposób szybko przetestować działanie własnego kodu i nie musi przesyłać pakietów przez sieć.

Na potrzeby odwołań z użyciem pętli zwrotnej zarezerwowano prefiks 127.0.0.0/8. Identyfikator stacji nie jest istotny — wszystkie adresy stacji są traktowane jednakowo. Zazwyczaj jednak programiści posługują się adresem 127/8 (z identyfikatorem stacji o wartości 1), co sprawia, że jest to najpopularniejszy adres pętli zwrotnej.

Żaden z pakietów wygenerowanych w czasie testu nie opuszcza komputera. Oprogramowanie IP przekazuje jedynie dane z jednej aplikacji do drugiej. Oznacza to również, że adres pętli zwrotnej nigdy nie występuje w pakietach przekazywanych przez sieć.

## 21.15. Zestawienie adresów IP o specjalnym znaczeniu

Wszystkie zarezerwowane wartości adresów IP zostały przedstawione w tabeli 21.4.

Tabela 21.4. Zestawienie adresów IP o specjalnym przeznaczeniu

Prefiks	Sufiks	Rodzaj adresu	Przeznaczenie
Same zera	Same zera	Niniejsza stacja { }	Wykorzystywany podczas uruchamiania komputera
Sieć	Same zera	Adres sieci	Identyfikuje sieć
Sieć	Same jedynki	Rozgłoszenie skierowane	Rozgłoszenie pakietu w wybranej sieci
Same jedynki	Same jedynki	Rozgłoszenie lokalne	Rozgłoszenie pakietu w sieci lokalnej
127/8	Dowolny	Pętla zwrotna	Testowanie

Zgodnie z wcześniejszymi informacjami adresy specjalne są zarezerwowane i nie powinny być przypisywane komputerom. Ponadto ich zastosowanie ogranicza się do pewnych określonych przypadków. Na przykład adres rozgłoszeniowy nigdy nie może występować w pakiecie jako adres źródłowy. Z kolei adres złożony z samych zer logicznych nie powinien być wykorzystywany, jeśli procedura uruchamiania systemu została zakończona i jednostka dysponuje właściwym adresem IP.

## 21.16. Adres rozgłoszeniowy w formacie Berkeley

Uniwersytet Kalifornijski z Berkeley opracował jedną z pierwszych implementacji stosu TCP/IP i zaważył ją w systemie BSD UNIX<sup>62</sup>. Kod BSD zawierał niestandardową funkcję, która została powielona w wielu późniejszych implementacjach stosu. Zamiast identyfikatora stacji złożonego z samych jedynek do reprezentowania adresu rozgłoszeniowego wykorzystano w niej sufiks obejmujący same zera (identyczny z adresem sieci). Taki zapis adresu rozgłoszeniowego jest znany jako **rozgłoszenie w formacie Berkeley**.

Niestety, w początkowej fazie stosowania protokołu IP na bazie oprogramowania z Berkeley opracowano wiele aplikacji TCP/IP. Niektóre z systemów nadal wykorzystują tę formę rozgłoszenia. Dlatego w oprogramowaniu TCP/IP często udostępniana jest opcja, która umożliwia wybranie odpowiedniego formatu adresu rozgłoszeniowego. Wiele rozwiązań pozwala również na stosowanie obydwu form zapisu. Administrator sieci musi więc sam zdecydować, który format będzie stosowany w jego sieci (jeśli rozgłoszenia są w ogóle dozwolone).

<sup>62</sup> Skrót BSD pochodzi od angielskich słów *Berkeley Software Distribution*, oznaczających sposób dystrybucji oprogramowania na zasadach opracowanych w Berkeley.

## 21.17. Routery i zasady adresowania IP

Specyfikacja protokołu internetowego stanowi, że oprócz przydzielania adresów IP komputerom przyłączonym do sieci, konieczne jest zagwarantowanie oddzielnego adresu dla routerów. W praktyce każdy router musi dysponować co najmniej dwoma adresami IP, po jednym na każdą sieć, do której jest przyłączony. Wynika to z tego, że:

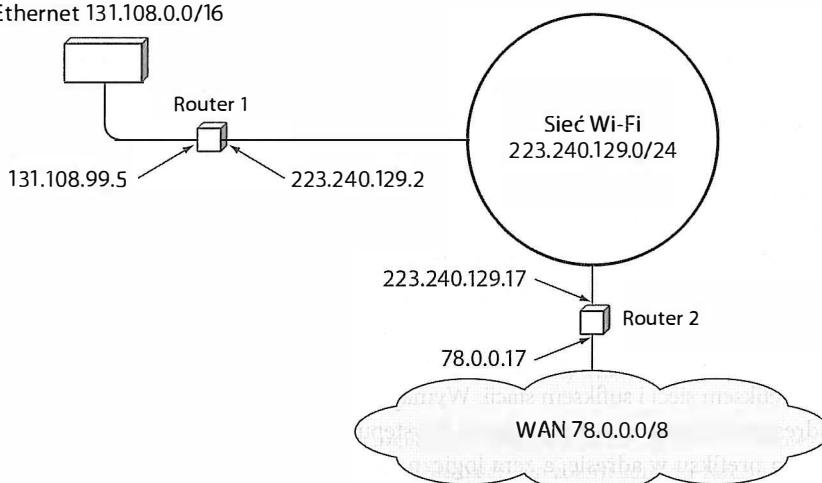
- Router jest przyłączony do większej liczby sieci.
- Każdy adres IP zawiera prefiks identyfikujący określoną sieć fizyczną.

Pojedynczy adres IP nie jest wystarczający dla routera, ponieważ musi on być przyłączony do kilku sieci, a każdej z nich obowiązuje inny prefiks. Na tej podstawie można sformułować jedną z fundamentalnych zasad adresowania IP:

*Adres IP nie identyfikuje komputera, ale połączenie między komputerem a siecią. Komputer wyposażony w większą liczbę kart sieciowych (na przykład router) musi mieć po jednym adresie IP na każde połączenie.*

Przykład praktycznego zastosowania powyższej reguły został przedstawiony na rysunku 21.4. Zaprezentowano na nim dwa routery przyłączone do trzech sieci.

Ethernet 131.108.0.0/16



Rysunek 21.4. Przydział adresów IP w sieci złożonej z dwóch routerów

Do poszczególnych interfejsów routera można przypisywać sufiksy o różnych wartościach. W przykładzie przedstawionym na rysunku router łączący sieć Ethernet z siecią Wi-Fi ma sufiksy 99.5 (po stronie Ethernetu) oraz 2 (po stronie sieci Wi-Fi).

Nic jednak nie stoi na przeszkodzie, aby ten sam sufiks był przypisany do wszystkich interfejsów. W rozwiążaniu zaprezentowanym na rysunku sufiks o wartości 17 został

przypisany dwóm interfejsom routera łączącego sieć Wi-Fi z siecią WAN. W praktyce stosowanie jednakowych sufiksów ułatwia zarządzanie siecią, ponieważ wymaga zapamiętania tylko jednej wartości.

## 21.18. Stacje o wielu interfejsach sieciowych

Czy komputery mogą być przyłączane do wielu sieci? Tak. Przyłączanie jednostek do większej liczby sieci jest często stosowane w celu zwiększenia niezawodności systemu. Jeśli jedna z sieci ulegnie awarii, dany komputer nadal będzie się mógł komunikować ze stacjami internetowymi za pośrednictwem drugiego interfejsu. To samo rozwiązanie bywa również wykorzystywane w celu zwiększenia wydajności sieci. Połączenia z różnymi sieciami umożliwiają niekiedy pominięcie routerów, które mogą być przeciążone. Podobnie jak routery, komputery o wielu interfejsach sieciowych muszą mieć wiele adresów IP (po jednym na każdy interfejs).

## 21.19. Podsumowanie

Aby zapewnić wrażenie pracy w jednej spójnej sieci, rozwiązania internetowe bazują na jednolitym mechanizmie adresowania. Każdy komputer otrzymuje niepowtarzalny adres IP, który jest wykorzystywany przez uruchomione w nim aplikacje do komunikacji z innymi jednostkami.

Specyfikacja protokołu IP opisuje podział adresu internetowego na dwie części, wyznaczające dwa poziomy hierarchii — część identyfikującą sieć oraz część odpowiadającą komputerowi przyłączonemu do danej sieci. W celu zagwarantowania niepowtarzalności adresów w ramach internetu prefiksy sieciowe są przyznawane jedynie przez organizacje specjalnie powołane do tego celu. Po uzyskaniu prefiksu administrator sieci firmowej może dowolnie przydzielać identyfikatory stacji poszczególnym komputerom.

Adres IP jest 32-bitową liczbą. Pierwotnie przestrzeń adresowa została podzielona na klasy, z których obecnie wykorzystywana jest jedynie klasa multiemisji. Technika adresowania klasowego została zastąpiona adresowaniem bezklasowym oraz wydzielaniem podsieci. W obydwu przypadkach istnieje możliwość dowolnego wyznaczania granicy między prefiksem sieci i sufiksem stacji. Wymagane jest jednak przechowywanie wraz z każdym adresem 32-bitowej maski podsieci. Występujące w masce jedynki logiczne odpowiadają bitom prefiksu w adresie, a zera logiczne są zapisywane na pozycjach odpowiadających identyfikatorowi stacji.

W standardzie IP zdefiniowano pewien zbiór adresów o specjalnym znaczeniu. Adresy specjalne opisują pętlę zwrotną (interfejs testowy), adres sieci, rozgłoszenia w sieci lokalnej oraz rozgłoszenia w sieci zdalnej.

Choć zazwyczaj adres IP kojarzy się z komputerem, tak naprawdę adresy IP identyfikują połączenia między komputerem a siecią. Routery i jednostki o większej liczbie interfejsów sieciowych dysponują więc wieloma adresami IP.

## ZADANIA

- 21.1. Czy protokół IP można zaprojektować ponownie w taki sposób, aby wykorzystywał adresy sprzętowe zamiast 32-bitowych wartości stosowanych obecnie? Uzasadnij odpowiedź.
- 21.2. Jaka jest zaleta hierarchicznego adresowania z punktu widzenia lokalnego administratora?
- 21.3. Czy w przypadku zastosowania pierwotnego podziału klasowego można ustalić klasę adresu jedynie na podstawie wartości samego adresu? Uzasadnij odpowiedź.
- 21.4. Napisz program komputerowy, który pobierze adres IP w formacie dziesiętnym i wyświetli na ekranie jego reprezentację binarną.
- 21.5. Napisz program komputerowy, który pobierze adres IP w formacie dziesiętnym i sprawdzi, czy jest to adres multiemisji.
- 21.6. Napisz program komputerowy, który zamieni adres w notacji CIDR (z ukośnikiem) na odpowiadającą mu wartość dziesiętną z kropkami.
- 21.7. Ile komputerom można przypisać adresy IP, jeśli dostawca usługi internetowych przydzielił danej sieci blok /28?
- 21.8. Dostawca usług internetowych oferuje blok adresów /17 za  $N$  złotych oraz blok adresów /16 za  $1,5N$  złotych miesięcznie. Który wariant gwarantuje niższy koszt w przeliczeniu na jeden komputer?
- 21.9. Czy prefiks 1.2.3.4/29 jest poprawny? Uzasadnij odpowiedź.
- 21.10. Czy dostawca usług internetowych dysponujący blokiem adresów /24 może zrealizować prośbę klienta o przydział adresów dla 255 komputerów (podpowiedź: należy uwzględnić adresy specjalne)?
- 21.11. W jaki sposób dostawca usług internetowych powinien podzielić przestrzeń adresową /22, aby przydzielić adresy czterem klientom, z których każdy ma 60 komputerów?
- 21.12. Czy dostawca usług internetowych dysponujący przestrzenią adresową /22 może zrealizować prośby sześciu klientów o przydział adresów sieciom, w których pracuje 9, 15, 20, 41, 128 i 260 komputerów? Jeśli tak, zaproponuj schemat podziału. Jeśli nie, wyjaśnij, dlaczego nie jest to możliwe.
- 21.13. Napisz program komputerowy, który pobierze adres w notacji CIDR, a następnie wyświetli wartość adresu i maski w formacie binarnym.
- 21.14. Napisz program komputerowy, który pobierze prefiks sieci w notacji CIDR oraz liczbę stacji sieciowych. Wynikiem działania aplikacji powinien być prefiks w notacji CIDR powstały po zrealizowaniu prośby o przydział adresów dla podanej liczby stacji, ale przy założeniu oszczędnego gospodarowania adresami.
- 21.15. Napisz program komputerowy, który pobierze 32-bitowy adres stacji oraz 32-bitową maskę w notacji CIDR i sprawdzi, czy podana wartość należy do zbioru adresów zarezerwowanych.
- 21.16. Jaką wartość ma adres rozgłoszeniowy w formacie Berkeley?
- 21.17. Ile adresów IP jest przypisywanych do routera? Wyjaśnij zagadnienie.
- 21.18. Czy komputer może mieć więcej niż jeden adres IP? Wyjaśnij zagadnienie.

# Zawartość rozdziału

- 22.1. Wprowadzenie 383
- 22.2. Usługa transmisji bezpołączeniowej 383
- 22.3. Wirtualne pakiety 384
- 22.4. Datagram IP 384
- 22.5. Format nagłówka datagramu IP 385
- 22.6. Przekazywanie datagramu IP 387
- 22.7. Odczytywanie prefiksów sieci i przekazywanie datagramów 388
- 22.8. Dopasowanie o najdłuższym prefiksie 389
- 22.9. Adresy docelowe i adresy następnego skoku 389
- 22.10. Brak gwarancji dostarczenia datagramu 390
- 22.11. Enkapsulacja IP 391
- 22.12. Transmisja datagramu w internecie 391
- 22.13. MTU i fragmentowanie datagramu 393
- 22.14. Odtwarzanie datagramu z fragmentów 394
- 22.15. Rejestrowanie fragmentów datagramu 395
- 22.16. Konsekwencje utraty pakietu 395
- 22.17. Fragmentowanie fragmentów 396
- 22.18. Podsumowanie 397

# 22

## Przekazywanie datagramów

### 22.1. Wprowadzenie

W poprzednich rozdziałach przedstawiona została architektura internetu oraz zasady doboru adresów internetowych. Tematem tego rozdziału jest natomiast podstawowa usługa komunikacyjna internetu. Omówione zostały tutaj takie zagadnienia jak format przesyłanych pakietów, najważniejsze rozwiązania w zakresie enkapsulacji datagramów, przekazywanie pakietów przez węzły sieci oraz fragmentacja i odtwarzanie pakietów. W kolejnych rozdziałach opisano inne protokoły, które uzupełniają tę usługę.

### 22.2. Usługa transmisji bezpołączeniowej

Celem projektantów internetu było opracowanie takiego pakietowego systemu komunikacyjnego, który umożliwi programowi działającemu w jednym komputerze dostarczenie danych do programu uruchomionego w innym systemie. W poprawnie zaprojektowanym internecie użytkownicy aplikacji są nieświadomi różnic w budowie sieci fizycznych — mogą wysyłać i odbierać dane bez wnikania w szczegóły połączenia między komputerem a siecią lokalną, budowy sieci zdalnej (do której przyłączony jest odbiorca) czy połączenia między obydwoma sieciami.

Jednym z najważniejszych założeń projektowych, które trzeba zdefiniować podczas prac nad tego typu systemem, jest określenie zakresu oferowanych usług. Sprowadza się to do ustalenia odpowiedzi na pytanie, czy usługa powinna mieć charakter **połączeniowy** (ang. *connection-oriented*), **beopołączeniowy** (ang. *connectionless*), czy powinna obsługiwać obydwa wymienione tryby pracy.

Twórcy stoso TCP/IP opracowali protokoły realizujące zarówno usługi połączeniowe, jak i bezpołączeniowe. Jednak zgodnie z ich założeniem podstawowa usługa dostarczania danych ma charakter bezpołączeniowy, a połączeniowe mechanizmy gwarantujące bez-

błędne przekazywanie informacji są zbudowane na bazie usługi bezpołączeniowej. Propozycja została pozytywnie oceniona i jest obecnie podstawą wszelkich komunikacji internetowych.

### 22.3. Wirtualne pakiety

Usługa bezpołączeniowa jest nieskomplikowanym rozwinięciem mechanizmu przełączania pakietów. Umożliwia bowiem nadawcy przesyłanie przez internet pojedynczych pakietów. Każdy pakiet jest przekazywany w sposób niezależny od pozostałych, na podstawie zawartego w nim identyfikatora odbiorcy.

W jaki sposób pakiety są przesyłane w internecie? Zasadnicza część pracy związanego z przekazywaniem danych należy do routerów. Jednostka źródłowa formuje pakiet, zapisuje w jego nagłówku adres stacji docelowej, a następnie dostarcza do najbliższego routera. Gdy router odbierze pakiet, analizuje zawarty w nim adres docelowy i na jego podstawie wybiera kolejny router na trasie do odbiorcy. Ostatecznie dane docierają do routera, który może przekazać je do ostatecznego adresata.

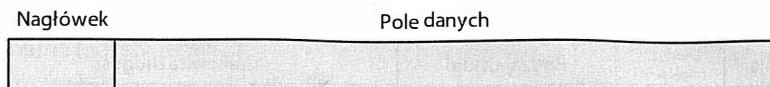
Jaki jest format pakietu internetowego? Z uwagi na to, że internet składa się z wielu różnych sieci o różnych formatach ramek, nie można było wykorzystać żadnego z istniejących wcześniej sposobów zapisu danych. Niemożliwa jest również zwykła zmiana formatu z zachowaniem istniejących wcześniej adresów (różne sieci bazują na różnych systemach adresowania).

Rozwiązaniem problemu heterogeniczności sieci okazało się opracowanie nowego formatu pakietów IP, który jest niezależny od warstwy sprzętowej. W wyniku podjętych prac powstał **uniwersalny pakiet wirtualny**, który umożliwia przekazywanie danych niezależnie od urządzeń sieciowych. Określenie **wirtualny** ma na celu podkreślenie niezależności formatu pakietu od rozwiązań sprzętowych. Z kolei **uniwersalność** wynika z tego, że każdy komputer lub router internetowy może wykorzystywać oprogramowanie, które poprawnie zinterpretuje wymieniane w sieci pakiety.

*Z uwagi na różnorodność sieci nie jest możliwe wykorzystanie w internecie jednego z formatów pakietów, które zostały zdefiniowane na poziomie warstwy sprzętowej. Uwzględnienie heterogeniczności systemu wymagało zdefiniowania odrębnego formatu pakietów IP, niezależnego od samych urządzeń transmisyjnych.*

### 22.4. Datagram IP

W specyfikacji protokołów TCP/IP pakiety internetowe są nazywane **datagramami IP**. Ogólny format datagramów IP jest zbliżony do formatu ramek sprzętowych — w ich początkowej części znajduje się nagłówek, po którym następuje pole danych, tak jak to zostało pokazane na rysunku 22.1.



**Rysunek 22.1.** Ogólna postać datagramu IP z nagłówkiem i polem danych

Podsumowując:

*Pakiet przesyłany w internecie TCP/IP jest nazywany datagramem IP. Każdy datagram składa się z nagłówka i pola danych.*

Ilość przenoszonych danych w datagramie nie jest stała. Zależy od bieżących potrzeb nadawcy. Na przykład aplikacje wysyłające informacje o naciskanych przez użytkownika klawiszach zapisują w datagramie jedynie kod danego klawisza. Z kolei programy przeznaczone do transferowania dużych plików zapisują w każdym pakiecie znaczne ilości danych.

*Rozmiar datagramu jest ustalany przez program wysyłający dane, a zmienność długość pakietu pozwala na adaptowanie mechanizmów IP do różnych potrzeb aplikacji.*

W bieżącej wersji protokołu IP (w wersji 4) datagram może przenosić od jednego oktetu danych do 64 000 oktetów (z uwzględnieniem nagłówka). W większości przypadków rozmiar nagłówka jest znacznie mniejszy od rozmiaru pola danych. Niemniej istnieje nagłówka wnosi pewien narzut transmisyjny — ponieważ rozmiar nagłówka jest stały, wysyłanie dużych datagramów gwarantuje przesłanie większej ilości danych w jednostce czasu (tj. zapewnia większą przepływność bitową).

## 22.5. Format nagłówka datagramu IP

Co zawiera nagłówek datagramu? Podobnie jak w przypadku ramki nagłówek datagramu przechowuje informacje potrzebne do przekazania pakietu. Występują w nim pola przeznaczone na adres źródłowy (adres nadawcy), adres docelowy (identyfikator adresata) oraz wartości odzwierciedlające rodzaj danych zawartych w polu danych. Każdy adres nagłówka jest adresem IP. Adresy MAC nadawcy i odbiorcy nie są zapisywane w datagramie.

Każde pole datagramu IP ma pewien ustalony rozmiar. Dzięki temu przetwarzanie pakietów jest znacznie efektywniejsze. Format nagłówka IP został przedstawiony na rysunku 22.2, a opis poszczególnych pól znajduje się pod rysunkiem.

**WERSJA.** Każdy datagram rozpoczyna się 4-bitowym polem numeru wersji protokołu (na rysunku przedstawiono format nagłówka wersji 4).

**DŁUGOŚĆ NAGŁÓWKA.** To 4-bitowe pole określa długość nagłówka wyrażoną w 32-bitowych wierszach. W przypadku braku opcji ma ono wartość 5.

0	4	8	16	19	24	31					
Wersja	Długość nagłówka	Rodzaj usługi	Całkowita długość								
Identyfikator		Znaczniki	Przesunięcie fragmentu								
TTL (czas życia)	Typ danych	Wartość kontrolna nagłówka									
Źródłowy adres IP											
Docelowy adres IP											
Opcje IP (mogą zostać pominięte)				Dopełnienie							
Początek pola danych (przesyłane dane)											
:											

Rysunek 22.2. Pola nagłówka datagramu IP w wersji 4

**RODZAJ USŁUGI.** To 8-bitowe pole przechowuje kod klasy usługi (rzadko wykorzystywany w praktyce). Interpretacja tej wartości została opisana w rozdziale 28.

**CAŁKOWITA DŁUGOŚĆ.** Jest to 16-bitowe pole, które określa całkowitą liczbę bajtów datagramu, włączając w to nagłówek i pole danych.

**IDENTYFIKATOR.** Jest to 16-bitowy numer (o przyrastających wartościach), który identyfikuje fragmenty pochodzące z jednego datagramu. Służy do odtworzenia datagramu w urządzeniu zdalnym.

**ZNACZNIKI.** To 3-bitowe pole składa się ze znaczników, które informują o tym, czy dany datagram jest fragmentem większego datagramu, a jeśli tak, to czy dany fragment jest ostatnim elementem większego datagramu.

**PRZESUNIĘCIE FRAGMENTU.** Jest to 13-bitowe pole, które wyznacza położenie danego fragmentu w pierwotnym datagramie. Pozycja fragmentu wynika z przemnożenia wartości przesunięcia przez osiem.

**TTL (CZAS ŻYCIA PAKIETU).** Jest to 8-bitowa liczba ustalana przez nadawcę, która w trakcie transmisji jest zmniejszana o jeden przez każdy router przekazujący datagram. Gdy wartość licznika osiągnie zero, pakiet jest usuwany, a nadawca datagramu otrzymuje powiadomienie o błędzie.

**TYP DANYCH.** To 8-bitowe pole zawiera informacje na temat rodzaju danych zapisanych w polu danych.

**WARTOŚĆ KONTROLNA NAGŁÓWKA.** Jest to 16-bitowa wartość kontrolna zapisana w kodzie uzupełnienia do jedności, wygenerowana zgodnie z algorytmem 8.1<sup>63</sup>.

**ŹRÓDŁOWY ADRES IP.** Jest to 32-bitowy adres internetowy pierwotnego nadawcy wiadomości (adresy routerów pośredniczących w dostarczaniu pakietów nie są rejestrowane w nagłówku).

<sup>63</sup> Algorytm 8.1 został przedstawiony na stronie 172.

**DOCELOWY ADRES IP.** Jest to 32-bitowy adres internetowy ostatecznego odbiorcy wiadomości (adresy routerów pośredniczących w dostarczaniu pakietów nie są rejestrowane w nagłówku).

**OPCJE IP.** W tej części zapisywane są opcjonalne wartości, które wpływają na proces routingu i przetwarzania datagramu. Większość pakietów nie zawiera żadnych opcji. Wówczas pole to można pominąć.

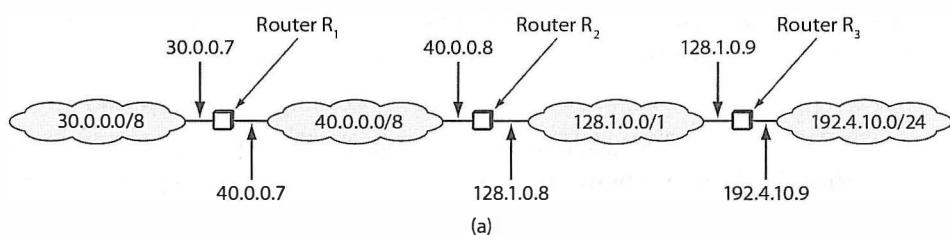
**DOPEŁNIENIE.** Jeśli opcje nie zajmują wielokrotności 32 bitów, dodawane są bity zerowe, które dopełniają nagłówek do wielokrotności 32 bitów.

## 22.6. Przekazywanie datagramu IP

Zgodnie z wcześniejszymi informacjami datagram jest przesyłany w internecie od stacji początkowej przez routery sieciowe do jednostki docelowej. Przekazywanie datagramów bazuje na idei obliczania następnego skoku. Każdy router, który odbierze pakiet, wyodrębnia z nagłówka adres docelowy i na podstawie odczytanej wartości określa następny skok datagramu. Następnie przesyła dane do kolejnego routera na trasie lub do jednostki docelowej.

Aby wybieranie kolejnych routerów na trasie było efektywne, routery IP korzystają z **tablicy routingu**. Tablica ta jest inicjowana w chwili uruchomienia urządzenia i musi być modyfikowana na bieżąco wraz ze zmianą topologii sieci lub w przypadku awarii jednostek sieciowych.

Tablicę routingu można postrzegać jako zbiór wierszy, które zawierają informację o sieciach docelowych oraz routerze następnego skoku (kolejnym routerze na trasie do celu). Na rysunku 22.3 przedstawiono przykładową sieć internetową oraz zawartość tablicy routingu jednego z węzłów (routera łączącego sieci składowe).



Cel	Maska	Następny skok
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	Bezpośrednie dostarczenie
128.1.0.0	255.255.0.0	Bezpośrednie dostarczenie
192.4.10.0	255.255.255.0	128.1.0.9

(b)

Rysunek 22.3. Przykład sieci internetowej złożonej z czterech sieci fizycznych (a) oraz tablica routingu routera R<sub>2</sub> (b)

W zaprezentowanym rozwiążaniu każdemu urządzeniu przydzielono dwa adresy IP (po jednym na każdy interfejs). Router R<sub>2</sub> jest bezpośrednio przyłączony do sieci 40.0.0.0/8 oraz 128.1.0.0/16. Adresy jego interfejsów to odpowiednio 40.0.0.8 i 128.1.0.8. Jak wiadomo z wcześniejszego rozdziału, sufiksy adresów nie muszą być jednakowe na wszystkich interfejsach. Ponieważ jednak zapamiętanie pojedynczej wartości jest znacznie łatwiejsze, administratorzy często wykorzystują ten sam sufiks na wszystkich interfejsach.

Szczególną uwagę warto zwrócić na rozmiar tablicy, który jest niezwykle istotny w przypadku globalnego internetu.

*Ponieważ wymieniane w tablicy routingu adresy docelowe odpowiadają sieciom, liczba wierszy tablicy jest proporcjonalna do liczby sieci w internecie, a nie do liczby stacji.*

## 22.7. Odczytywanie prefiksów sieci i przekazywanie datagramów

Proces wybierania z tablicy routingu kolejnego węzła na trasie pakietu jest nazywany **routingiem**. Z informacji zamieszczonych w rozdziale 21. wiadomo, że pole *maski* służy do wyodrębniania z adresu identyfikatora sieci. Gdy router otrzyma datagram z docelowym adresem o wartości *D*, funkcja routingu musi odnaleźć w tablicy routingu wpis, który wyznaczy kolejny węzeł na trasie do stacji *D*. W tym celu oprogramowanie analizuje każdy wpis w tablicy, wyodrębniając za pomocą maski prefiks sieci zawarty w adresie *D* i porównując go z wartością kolumny *cel* tego samego wiersza. Jeśli te dwie wartości są sobie równe, pakiet jest przekazywany dalej z zawartością kolumny *następny skok*.

Bitowa reprezentacja maski gwarantuje efektywne wyodrębnianie identyfikatora sieci — obliczenie sprowadza się do wykonania **iloczynu** logicznego maski i adresu docelowego *D*. Operację realizowaną na *i*-tym wierszu można więc zapisać w następujący sposób:

```
if ((Maska[i] & D) == Cel[i]) przekaż do NastępnySkok[i];
```

Jako przykład działania mechanizmu rozważmy datagram kierowany na adres 192.4.10.3, który w danej chwili jest przetwarzany w routerze R<sub>2</sub> należącym do sieci przedstawionej na rysunku 22.3. Dodatkowo przyjmijmy, że wiersze tablicy routingu są analizowane w kolejności występowania na rysunku. Pierwszy wpis nie pasuje do datagramu, ponieważ wynik iloczynu logicznego 255.0.0 & 192.4.10.3 jest różny od 30.0.0.0. Z tych samych powodów odrzucane są wiersze drugi i trzeci. Ostatecznie jako kolejny router na trasie wybrane zostanie urządzenie o adresie 128.1.0.9, ponieważ:

```
255.255.255.0 & 192.4.10.3 == 192.4.10.0
```

## 22.8. Dopasowanie o najdłuższym prefiksie

Na rysunku 22.3 przedstawiono niezbyt skomplikowany przypadek. Rzeczywiste tablice routingu są zazwyczaj bardzo duże, a algorytm routingu jest znacznie bardziej złożony. Internetowe tablice routingu mogą na przykład zawierać informacje o trasach **domyślnych** (podobnie jak opisane w rozdziale 18. mechanizmy przekazywania pakietów w sieciach WAN), które wyznaczają określoną trasę dostarczania datagramów do wszystkich jednostek docelowych niewymienionych wprost w tablicy routingu. Ponadto administrator routera ma możliwość zdefiniowania specjalnej **trasы do pojedynczej stacji**. Dzięki temu wszystkie pakiety kierowane do określonej jednostki mogą być dostarczane inną trasą niż pakiety do pozostałych stacji tej samej sieci (wymaga to zdefiniowania wpisu z 32-bitową maską, która oznacza, że wszystkie bity adresu muszą zostać dopasowane do wartości kolumny *cel*).

Routing internetowy ma pewną ciekawą cechę wynikającą z możliwości nakładania się wartości masek. Oto przykład dwóch prefiksów sieci zapisanych w dwóch wierszach jednej tablicy routingu:

```
128.10.0.0/16  
128.10.2.0/24
```

Co się stanie, gdy router odbierze datagram adresowany do jednostki 128.10.2.3? Procedura wyszukania pasujących wierszy powinna wskazać dwa dopasowania. Iloczyn logiczny adresu i 16-bitowej maski ma bowiem wartość 128.10.0.0, a wykonanie tej samej operacji z maską 24-bitową daje w wyniku wartość 128.10.2.0. Który wpis należy zastosować?

Niejednoznaczność dopasowań w przypadku nakładających się masek jest rozwiązywana przez wybieranie **dopasowania o najdłuższym prefiksie** (ang. *longest prefix match*). Dlatego zamiast porównywać wiersze w kolejności ich wprowadzania (w kolejności przypadkowej), oprogramowanie routingu sortuje wpisy tak, aby analizować w pierwszej kolejności te, które gwarantują najdłuższe dopasowanie. W przedstawionym przykładzie wybrany zostałby więc wpis 128.10.2.0/24. Należy zatem zapamiętać, że:

*Aby uniknąć niejednoznaczności wynikającej z dopasowania większej liczby wierszy do danego adresu docelowego, mechanizmy routingu internetowego sprawdzają w pierwszej kolejności wpisy o dłuższym prefiksie.*

## 22.9. Adresy docelowe i adresy następnego skoku

Jaka jest zależność między adresem docelowym zapisanym w nagłówku datagramu a adresem następnego skoku? Pole *docelowy adres IP* występujące w pakiecie odpowiada adresowi odbiorcy końcowego. Jego wartość nie ulega zmianie przez cały czas przesyłania datagramu w internecie. Gdy którykolwiek z routerów na trasie odbierze taki pakiet, analizuje adres odbiorcy końcowego *D* i na jego podstawie wyznacza następny router (*N*), do którego datagram powinien zostać wysłany. Mimo że pakiet jest dostarczany do routera *N*, zawarty w nagłówku adres docelowy pozostaje niezmieniony (ma wartość *D*). Innymi słowy:

Zawarty w nagłówku adres docelowy zawsze wskazuje ostatecznego odbiorcę pakietu. W każdym węźle sieci wyznaczany jest router następnego skoku, ale jego adres nie jest w żaden sposób odwzorowywany w nagłówku datagramu.

## 22.10. Brak gwarancji dostarczenia datagramu

Poza definicją formatu datagramów specyfikacja protokołu internetowego zawiera również opis przebiegu komunikacji. Zgodnie z nim realizowana przez protokół usługa jest rozwiązaniem typu **best-effort**. Oznacza to, że mimo iż protokół IP dokłada wszelkich starań, aby poprawnie dostarczyć datagram, nie można wykluczyć sytuacji wyjątkowej, w której pakiet nie dotrze do odbiorcy. W standardzie IP wymieniono nawet kilka potencjalnych problemów:

- zduplikowanie datagramu,
- opóźnione dostarczenie lub dostarczenie w niewłaściwej kolejności,
- uszkodzenie danych,
- utrata datagramu.

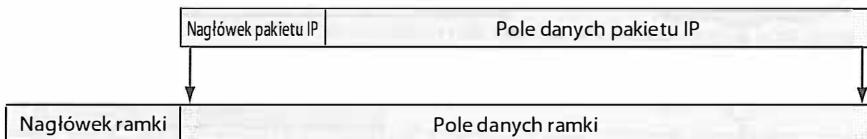
Fakt zamieszczenia w specyfikacji informacji o ewentualności wystąpienia błędów może się wydawać nieco dziwny. Niemniej jest ku temu ważny powód — protokół IP został zaprojektowany z myślą o działaniu w sieciach różnego typu. Jak wiadomo z poprzednich rozdziałów, urządzenia sieciowe bywają narażone na zaldócenia, które prowadzą do przekłamania danych lub ich utraty. W systemach, w których trasy przekazywania informacji są zmieniane dynamicznie, istnieje także ryzyko wybrania dla dwóch pakietów tras o różnych długościach, co może skutkować dostarczeniem ich w niewłaściwej kolejności.

*Z uwagi na to, że protokół IP został zaprojektowany do współpracy ze wszystkimi urządzeniami sieciowymi (również tymi, w których mogą wystąpić pewne problemy), datagramy IP mogą być tracone, duplikowane, opóźniane, dostarczane w niewłaściwej kolejności lub dostarczane z niepoprawnymi danymi.*

Na szczęście, w stosie protokołów TCP/IP zostały zdefiniowane dodatkowe mechanizmy, które zapewniają obsługę większości nietypowych zdarzeń. Ponadto niektóre aplikacje są specjalnie opracowywane w taki sposób, aby wykorzystywały usługi typu best-effort, a nie mechanizmy gwarantujące detekcję i korekcję błędów (informacje na ten temat zostały przedstawione w dalszej części książki).

## 22.11. Enkapsulacja IP

W jaki sposób datagram może być transmitowany w sieci (fizycznej), która nie obsługuje danego formatu pakietów? Wymaga to zastosowania techniki nazywanej **enkapsulacją**. Polega ona na umieszczeniu datagramu w polu danych ramki. Komponenty sprzętowe sieci traktują ramkę zawierającą datagram tak samo, jak każdą inną transmitowaną ramkę. Urządzenia sieciowe nie analizują ani nie zmieniają zawartości pola danych. Rozwiązanie to zostało zilustrowane na rysunku 22.4.



Rysunek 22.4. Enkapsulacja datagramu IP w ramce

Skąd odbiorca uzyskuje informacje o tym, że ramka zawiera datagram IP lub inny zbiór danych? Nadawca i odbiorca muszą stosować jednakowe oznaczanie rodzaju danych, które są zapisywane w polu *typu danych* ramki. Gdy oprogramowanie komputera źródłowego umieszcza datagram w ramce, jednocześnie zapisuje w polu typu danych specjalną wartość zarezerwowaną dla protokołu IP. Z kolei jednostka docelowa odczytuje wartość tego pola i na jej podstawie wie, że w polu danych jest zapisany datagram IP. Zgodnie ze standardem Ethernet pole typu w ramce przenoszącej datagram IP powinno mieć wartość **0x0800**.

Każda ramka (w tym ta przenosząca datagram IP) musi mieć zdefiniowany adres docelowy. Proces enkapsulacji nie ogranicza się więc jedynie do zapisania pakietu w polu danych ramki. Wymaga również określenia adresu MAC komputera, do którego ramka powinna zostać dostarczona. Wymaga to zdefiniowania w komputerze nadawczym powiązania między adresem IP jednostki zdalnej a odpowiadającym jej adresem MAC. Za definiowanie takich odwzorowań odpowiada protokół ARP, który został opisany w następnym rozdziale.

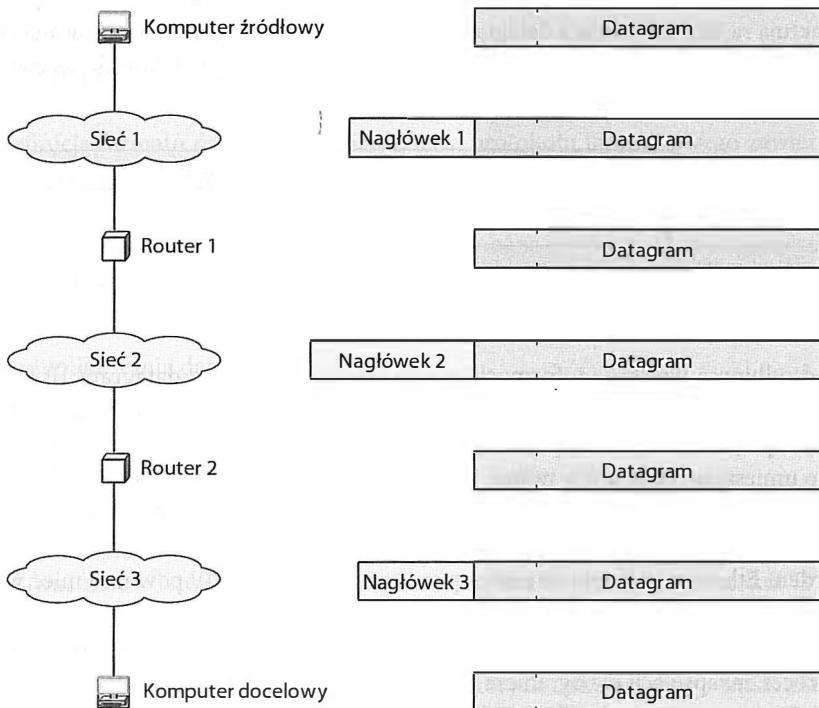
Podsumowując:

*Na czas transportu przez sieć fizyczną datagram jest zapisywany w polu danych ramki. Adres docelowy ramki odpowiada adresowi MAC najbliższej stacji, do której datagram jest przekazywany. Adres ten uzyskuje się przez odwzorowanie adresu IP następnego węzła na odpowiadający mu adres MAC.*

## 22.12. Transmisja datagramu w internecie

Procedura enkapsulacji odnosi się do jednorazowej emisji ramki. Po wybraniu najbliższego węzła nadawca zapisuje datagram w polu danych ramki i przesyła ją przez sieć fizyczną. W chwili dostarczenia danych do następnego węzła oprogramowanie odbiorcy wyod-

rębnia datagram IP i odrzuca pola ramki. Jeśli pakiet musi zostać przekazany przez kolejną sieć fizyczną, tworzona jest kolejna ramka. Proces ładowania pakietu w ramkę i wyodrębniania go z ramki w czasie transmisji na trasie między trzema sieciami (z dwoma routera mi) został przedstawiony na rysunku 22.5.



Rysunek 22.5. Datagram IP przekazywany przez internet

Jak wynika z rysunku, komputery i routery przechowują datagramy w pamięci bez dodatkowych nagłówków. Jednak w czasie przekazywania przez sieć fizyczną datagramy są zapisane w polu danych ramki (właściwie dla danej sieci). Rozmiar nagłówka ramki zależy od technologii danej sieci. Na przykład jeśli w sieci Sieć 1 obowiązuje standard Ethernet, wówczas nagłówek ramki 1 jest nagłówkiem ethernetowym. Analogicznie, jeśli Sieć 2 jest siecią Wi-Fi, nagłówek ramki 2 odpowiada nagłówkowi Wi-Fi.

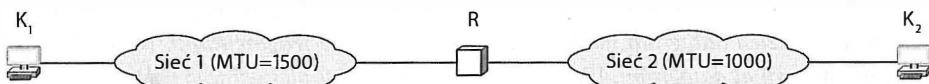
Ważne jest to, że nagłówki ramek nie są kumulowane podczas transmisji danych przez internet. Po odebraniu datagramu węzeł pośredniczący wyodrębnia pakiet z ramki, a następnie zapisuje go w polu danych ramki wychodzącej. Zatem w chwili dostarczenia datagramu do odbiorcy końcowego jedynym nagłówkiem ramki jest ten, który obowiązuje w sieci końcowej. Po jego usunięciu odbiorca dysponuje oryginalnym datagramem.

*Po odebraniu ramki z datagramem stacja docelowa wyodrębnia datagram z pola danych ramki i odrzuca nagłówek ramki.*

## 22.13. MTU i fragmentowanie datagramu

W każdej technologii sieciowej zdefiniowana jest pewna maksymalna ilość danych, którą można przenosić w ramce. Limit ten jest nazywany **maksymalną jednostką transmisyjną** (MTU — ang. *Maximum Transmission Unit*). Nie ma wyjątków w jego stosowaniu — urządzenia sieciowe są projektowane w taki sposób, żeby nie przetwarzali więcej danych, niż wynika to z parametru MTU. Datagram musi mieć więc rozmiar mniejszy od wartości MTU danej sieci lub równy temu parametrowi. W przeciwnym razie nie będzie podlegał enkapsulacji.

W internecie złożonym z heterogenicznych rozwiązań ograniczenia rozmiaru pola danych są istotnym problemem. Uwidacznia się to chociażby w sytuacji, w której router jest przyłączony do dwóch sieci o różnych wartościach MTU. Datagram odebrany z jednej sieci może mieć zbyt duży rozmiar, aby został przesłany przez drugą sieć. Przykład takiego zostało przedstawiony na rysunku 22.6 — router łączy dwie sieci o parametrach MTU wynoszących 1500 i 1000 bajtów.



Rysunek 22.6. Przyłączenie routera do dwóch sieci o różnych wartościach MTU

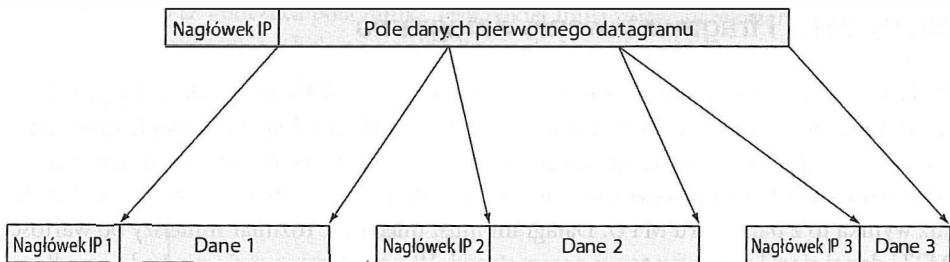
W przedstawionym przykładzie komputer  $K_1$  jest przyłączony do sieci o MTU 1500 i może przesyłać datagramy złożone z nie więcej niż 1500 oktetów. Komputer  $K_2$  pracuje w sieci, której parametr MTU wynosi 1000, co oznacza, że nie może wysyłać i odbierać datagramów przekraczających 1000 oktetów. Jeśli komputer  $K_1$  wyśle pakiet złożony z 1500 oktetów do komputera  $K_2$ , router nie będzie mógł zapisać datagramu w ramce emitowanej w sieci 2.

Aby rozwiązać problem różnych wartości MTU w poszczególnych sieciach, opracowano technikę **fragmentacji** pakietów. Jeśli datagram jest większy niż MTU sieci, w której powinien zostać przesłany, router dzieli go na mniejsze części nazywane **fragmentami**, a następnie emituje kolejne fragmenty.

Każdy fragment ma taki sam format jak inne datagramy. Jedynie odpowiedni bit w polu **znaczników** wyróżnia fragmenty spośród kompletnych datagramów<sup>64</sup>. Nagłówek zawiera również inne informacje potrzebne do **odtworzenia** pierwotnego datagramu w jednostce docelowej. Niezbędna jest wartość zapisana w polu **przesunięcie fragmentu**, która określa położenie danego fragmentu względem początku oryginalnego datagramu.

Aby podzielić datagram przed wyemitowaniem go w sieci, router wylicza maksymalny rozmiar pola danych na podstawie wartości MTU oraz rozmiaru nagłówka. Ustala również, ile fragmentów zostanie wygenerowanych i odpowiednio je numeruje. Następnie formuje poszczególne fragmenty, zapisując w ich nagłówkach oryginalne wartości pól **adres źródłowy** oraz **adres docelowy**. Kopiuje również część pola danych oryginalnego datagramu i wysyła powstałe pakiety. Proces podziału został przedstawiony graficznie na rysunku 22.7.

<sup>64</sup> Format nagłówka datagramu został przedstawiony na rysunku 22.2 na stronie 386.



Rysunek 22.7. Datagram IP podzielony na trzy fragmenty.

Fragment końcowy jest mniejszy niż pozostałe

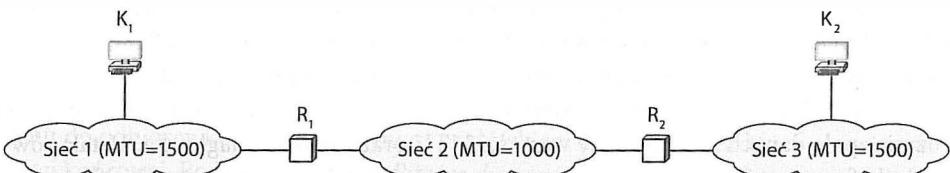
Podsumowując:

*W każdej sieci jest wyznaczona wartość MTU, która określa maksymalną ilość danych, jaka może zostać przesłana w ramce. Gdy router odbiera datagram o rozmiarze większym niż wartość MTU obowiązująca w sieci, w której dane powinny zostać przesłane, dzieli datagram na mniejsze części nazywane fragmentami. Każdy fragment ma format datagramu IP, ale przenosi jedynie część pierwotnego pola danych.*

## 22.14. Odtwarzanie datagramu z fragmentów

Proces odtwarzania pierwotnego datagramu na podstawie jego fragmentów jest niekiedy nazywany **reasemblacją**. Dzięki temu, że wszystkie fragmenty zawierają kopię oryginalnego nagłówka<sup>65</sup>, w każdym jest zapisany adres źródłowy oraz adres docelowy jednostek biorących udział w transmisji. Fragment odpowiadający ostatniej porcji danych ma w nagłówku ustawiony specjalny bit, na którego podstawie stacja końcowa może stwierdzić, czy wszystkie fragmenty zostały poprawnie odebrane.

Zgodnie ze specyfikacją IP odtwarzanie pierwotnego datagramu odbywa się tylko w jednostce końcowej. Przeanalizujmy działanie opisanego mechanizmu na przykładzie z rysunku 22.8.



Rysunek 22.8. Trzy sieci połączone przez dwa routery

Jeśli komputer  $K_1$  wyśle do komputera  $K_2$  datagram o rozmiarze 1500 oktetów, router  $R_1$  musi go podzielić na dwa fragmenty, które następnie dostarczy do routera  $R_2$ . Router  $R_2$

<sup>65</sup> Inne są jedynie pola opisujące sam podział na fragmenty.

nie odtwarza pierwotnych datagramów, lecz przekazuje wszystkie odebrane fragmenty na podany w nich adres docelowy. Oryginalne datagramy są odtwarzane dopiero w komputerze  $K_2$ , który jako odbiorca końcowy ma obowiązek zgromadzić poszczególne części pierwotnego zbioru danych i przywrócić ich oryginalną postać.

Przeniesienie obowiązku odtwarzania datagramów na jednostkę końcową ma dwie zalety:

- Zmniejsza ilość informacji na temat stanu połączenia, które trzeba by gromadzić w routerach — router nie musi w specjalny sposób traktować datagramów przenoszących fragmenty oryginalnego pola danych.
- Umożliwia dynamiczną zmianę tras. Gdyby odtwarzanie datagramów miało być realizowane w węzłach pośrednich, wszystkie pakiety z jednego połączenia miałyby być przekazywane przez te same routery.

Dzięki odroczeniu reasemblacji do chwili dostarczenia pakietu do odbiorcy końcowego, systemy IP mogą przekazywać poszczególne fragmenty datagramów różnymi trasami, niezależnie od tras wybranych podczas transmisji wcześniejszych pakietów. Trasy mogą się więc zmieniać w dowolnym czasie (na przykład w celu ominięcia uszkodzonego odcinka sieci).

## 22.15. Rejestrowanie fragmentów datagramu

Zgodnie z wcześniejszymi informacjami protokół IP nie gwarantuje poprawnego dostarczania danych. Poszczególne fragmenty (transmitowane są w taki sam sposób, jak inne datagramy) mogą więc być tracone lub dostarczane w niewłaściwej kolejności. Co ważniejsze, jeśli jedna stacja źródłowa emituje wiele datagramów przeznaczonych dla tego samego odbiorcy, fragmenty różnych datagramów mogą docierać do jednostki końcowej w przypadkowej kolejności.

Jak zatem oprogramowanie IP przetwarza fragmenty docierające w niewłaściwej kolejności? Nadawca zapisuje w każdym wysyłanym datagramie specjalną wartość identyfikacyjną (w polu *identyfikator*). Podział datagramu na mniejsze części zawsze wiąże się z zapisaniem identyfikatora oryginału w odpowiednim polu każdego fragmentu. Na podstawie tej wartości identyfikatora oraz źródłowego adresu IP odbiorca może pogrupować nadchodzące fragmenty jednego datagramu. Dodatkowe pole — *przesunięcie fragmentu* — stanowi dla jednostki odbiorczej informację o tym, w której części odtwarzanego datagramu należy zapisać dany fragment.

## 22.16. Konsekwencje utraty pakietu

Jak wiadomo, protokół IP nie gwarantuje poprawnego dostarczenia datagramów — jeśli z jakichkolwiek powodów ramki zostaną odrzucone, zawarte w nich datagramy lub fragmenty datagramów zostaną utracone. Jest również oczywiste, że pierwotnego datagramu

nie można odtworzyć, jeśli nie zostaną dostarczone wszystkie jego fragmenty. Może się zdarzyć tak, że część fragmentów datagramu zostanie dostarczona poprawnie, a pewne z nich zostaną opóźnione lub utracone.

Jednostka odbiorcza nie może przechowywać fragmentów przez nieskończenie długi czas, ponieważ każda porcja danych zajmuje pewien obszar pamięci. Aby uniknąć zajęcia całej dostępnej pamięci operacyjnej, wyznaczany jest pewien maksymalny czas odtwarzania pakietu. Wraz z pierwszym odebranym fragmentem odbiorca uruchamia **zegar odtwarzania** (ang. *reassemble timer*). Jeśli wszystkie fragmenty dotrą do danej stacji przed upływem czasu odtwarzania, oprogramowanie wyłącza zegar i odbudowuje datagram. Jeśli jednak upłynie dopuszczalny czas, wszystkie zgromadzone wcześniej fragmenty zostaną usunięte z pamięci.

Użycie zegara wprowadza zasadę „wszystko albo nic” — albo wszystkie fragmenty zostaną odebrane i pakiet zostanie odtworzony, albo oprogramowanie IP odrzuci dany datagram. Nie ma żadnego mechanizmu, który informowałby nadawcę o tym, które fragmenty oryginalnego pakietu zostały poprawnie dostarczone.

Taki sposób postępowania jest uzasadniony, ponieważ nadawca nie wie, w jaki sposób datagram został podzielony. Ponadto w przypadku ewentualnej retransmisji datagramu trasa mogłaby się zmienić, a to oznacza, że retransmitowane dane niekoniecznie musiałyby być przekazywane przez te same routery. Retransmitowany datagram nie musiałby więc zostać podzielony w taki sam sposób, jak pierwotny.

## 22.17. Fragmentowanie fragmentów

Po wykonaniu fragmentacji router przekazuje każdy fragment wzdłuż trasy prowadzącej do stacji docelowej. Co się stanie, jeśli na jednym z etapów transmisji fragment natrafi na sieć, która ma mniejszą wartość MTU? Mechanizm fragmentacji został opracowany w taki sposób, aby umożliwić również podział podzielonych wcześniej pakietów. Jeśli na przykład trasa między sieciami składa się z odcinków o coraz mniejszych wartościach MTU, każdy z routerów dzieli otrzymany fragment na jeszcze mniejsze fragmenty. Choć jest to możliwe, projektanci sieci starają się unikać takich sytuacji w internecie.

Niemniej żaden węzeł IP nie jest w stanie odróżnić oryginalnych fragmentów od części tych fragmentów. Nawet odbiorca nie dysponuje informacją o tym, czy nadchodzący fragment jest wynikiem pojedynczego podziału datagramu na części, czy taka operacja została wykonana wielokrotnie. Dzięki temu, że wszystkie fragmenty są przetwarzane w jednakowy sposób, odbiorca nie musi poprzedzać odtwarzania datagramu odtwarzaniem jego fragmentów. Rozwiążanie to powoduje znaczne oszczędności w gospodarowaniu zasobami procesora oraz eliminuje konieczność przekazywania dodatkowych informacji o nagłówku każdego fragmentu.

## 22.18. Podsumowanie

Specyfikacja protokołu internetowego zawiera definicje datagramu IP jako podstawowej jednostki transmisyjnej w sieci TCP/IP. Każdy datagram przypomina swoją budową ramkę warstwy sprzętowej. Zawiera nagłówek oraz pole danych. Podobnie jak w ramce sprzętowej, w nagłówku zapisane są informacje niezbędne do dostarczenia datagramu do odpowiedniej jednostki końcowej. Jednak w przeciwieństwie do ramki sprzętowej w nagłówku datagramu nie występują adresy MAC, ale adresy IP.

Działające w routerach oprogramowanie IP wykorzystuje tablice routingu do wyznaczania kolejnych węzłów na trasie datagramu. Każdy wpis w tablicy routingu zawiera informacje o sieci docelowej. Dlatego liczba wpisów jest proporcjonalna do liczby sieci w internecie. Wybierając trasę, router porównuje prefiks sieci zapisany w adresie docelowym z wartościami zarejestrowanymi w tablicy routingu. Aby uniknąć niejednoznaczności, w przypadku wielokrotnych dopasowań wybierane jest to o dłuższym prefiksie.

Choć oprogramowanie IP wyznacza następne węzły na trasie datagramu, adresy kolejnych routerów nigdy nie są odnotowywane w nagłówkach tych datagramów. Zapisany w nagłówku adres zawsze wskazuje ostatecznego odbiorcę danych.

Przygotowanie datagramu do transmisji polega na zapisaniu go w polu danych ramki. W każdej technologii sieciowej istnieje ograniczenie maksymalnego rozmiaru danych — wyznaczany jest parametr MTU (maksymalnej jednostki transmisyjnej). Jeśli rozmiar datagramu przekroczy wartość MTU, datagram podlega fragmentacji. Fragment może być w razie potrzeby podzielony na kolejne fragmenty. Odtwarzanie pierwotnego datagramu jest realizowane jedynie w stacji końcowej. W operacji tej wykorzystywany jest zegar odtwarzania, który może spowodować odrzucenie datagramu w przypadku utraty jego fragmentów składowych.

## ZADANIA

- 22.1. Wymień dwa podstawowe mechanizmy komunikacji, które trzeba uwzględnić podczas projektowania sieci internetowej.
- 22.2. W jaki sposób w projekcie internetu uwzględniane są sieci heterogeniczne o różnych formatach ramek?
- 22.3. Napisz program komputerowy, który wyodrębni z datagramu IP adres źródłowy i docelowy, a następnie wyświetli je na ekranie w formacie dziesiętnym.
- 22.4. Napisz program komputerowy, który wyodrębni wszystkie pola nagłówka IP. Wartości pól powinny zostać wyświetlone na ekranie w formacie szesnastkowym lub dziesiętnym, zależnie od rodzaju pola.
- 22.5. Jaka jest maksymalna długość datagramu IP?
- 22.6. Napisz program komputerowy, który pobierze tablicę routingu (w formacie przedstawionym na rysunku 22.3b) oraz listę adresów docelowych, a następnie przeszuka sekwencyjnie tablicę routingu i wyznaczy adres następnego węzła na trasie do każdej z wymienionych stacji docelowych.

- 22.7. Jakie wartości zostaną zapisane w polach *długość nagłówka* i *całkowita długość*, jeśli datagram przenosi jedną 8-bitową wartość i nie trzeba definiować żadnych opcji dodatkowych?
- 22.8. Który prefiks zostanie wykorzystany przez algorytm routingu, jeśli adres docelowy datagramu pasuje do dwóch prefiksów?
- 22.9. Czy w datagramie IP jest kiedykolwiek rejestrowany adres routera pośredniczącego w transmisji danych? Uzasadnij odpowiedź.
- 22.10. Założmy, że dwa routery zostały błędnie skonfigurowane i powstała pętla routingu dla pewnego adresu docelowego (*D*). Wyjaśnij, dlaczego datagram adresowany do stacji *D* nie będzie krążył w sieci nieskończonym dłużej.
- 22.11. Jakie problemy mogą wystąpić podczas przesyłania datagramu przez internet?
- 22.12. W którym miejscu ramki jest zapisywany datagram IP?
- 22.13. Jeśli w jednym ze śródkowych odcinków internetu zostanie przechwycony datagram IP, ile nagłówków ramek będzie poprzedzało ten datagram?
- 22.14. Do czego służy parametr MTU?
- 22.15. Na ile fragmentów zostanie podzielony datagram o rozmiarze 1480 bajtów, który musi zostać przesłany przez sieć o MTU wynoszącym 500 bajtów? Uzasadnij odpowiedź.
- 22.16. Która jednostka internetowa odpowiada za odtwarzanie datagramu na podstawie pakietów?
- 22.17. Skąd jednostka odtwarzająca datagram wie, że dana grupa fragmentów pochodzi z tego samego datagramu?
- 22.18. Czy w przypadku utraty jednego fragmentu odbiorca żąda dostarczania jego nowej kopii? Uzasadnij odpowiedź.
- 22.19. Zapoznaj się z dokumentami RFC 1149 oraz RFC 1217. Czy są to istotne standardy sieciowe (podpowiedź: sprawdź daty)?
- 22.20. Opracuj emulator bramy internetowej, który będzie losowo odrzucał, duplikował i opóźniał pakiety.



# Zawartość rozdziału

- 23.1. Wprowadzenie 401
- 23.2. Odwzorowanie adresów 401
- 23.3. Protokół odwzorowania adresu (ARP) 403
- 23.4. Format komunikatu ARP 403
- 23.5. Enkapsulacja ARP 405
- 23.6. Buforowanie ARP i przetwarzanie komunikatów 406
- 23.7. Teoretyczna granica stosowania adresów 408
- 23.8. Internetowy protokół komunikatów sterujących (ICMP) 408
- 23.9. Format komunikatu i enkapsulacja ICMP 410
- 23.10. Oprogramowanie, parametry i konfiguracja protokołu 411
- 23.11. Protokół dynamicznej konfiguracji stacji (DHCP) 411
- 23.12. Działanie protokołu DHCP i optymalizacja pracy 413
- 23.13. Format komunikatu DHCP 414
- 23.14. Pośrednictwo w dostępie do serwera DHCP 415
- 23.15. Translacja adresów sieciowych (NAT) 415
- 23.16. Działanie usługi NAT i adresy prywatne 416
- 23.17. Translacja NAT na poziomie warstwy transportowej (NAPT) 418
- 23.18. Operacja NAT a dostęp do serwerów 419
- 23.19. Oprogramowanie NAT i systemy przeznaczone do sieci domowych 420
- 23.20. Podsumowanie 420

# 23

## *Protokoły i technologie uzupełniające*

### **23.1. Wprowadzenie**

Rozdziały tej części książki odnoszą się do internetu i związanych z nim technologii. Po przedstawieniu idei wymiany danych między sieciami oraz architektury internetu zaprezentowano w niej system adresowania IP, mechanizm adresowania bezklasowego, format datagramu IP oraz podstawy routingu pakietów. Tematem poprzedniego rozdziału była enkapsulacja, fragmentacja i odtwarzanie datagramów.

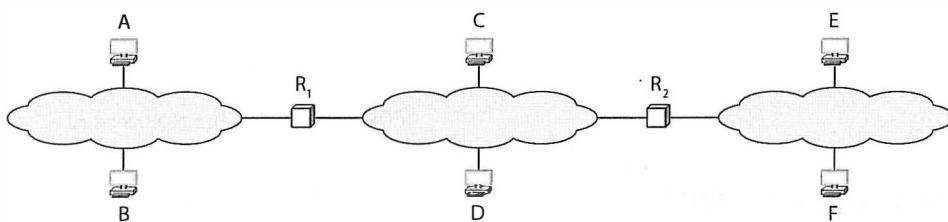
W tym rozdziale kontynuowane są rozważania na temat pracy internetowej, a dokładniej na temat odwzorowania adresów, zgłaszania błędów, uruchamiania stacji sieciowych oraz translacji adresów. Każda z prezentowanych technologii odpowiada za pewien aspekt komunikacji, a ich współdziałanie z innymi protokołami transmisyjnymi istotnie zwiększa funkcjonalność internetu. Kolejne rozdziały zawierają omówienie protokołu warstwy transportowej oraz protokołów routingu.

### **23.2. Odwzorowanie adresów**

Z informacji zamieszczonych w rozdziale 22. wiadomo, że jednostka źródłowa oraz każdy router sieciowy wykorzystuje docelowy adres IP do wyboru następnego węzła na trasie, po czym zapisuje datagram w polu danych ramki i wysyła tę ramkę do sieci. Jednym z najważniejszych etapów procesu routingu jest odwzorowanie adresów — mechanizmy routingu bazują na adresach IP, natomiast transmisja danych w obszarze jednej sieci wymaga określenia adresu MAC jednostki odbiorczej. Oprogramowanie IP musi więc zamienić adres IP kolejnego węzła na odpowiadający mu adres MAC. Zgodnie z ogólną zasadą:

*Adresy IP są wartościami abstrakcyjnymi narzuconymi przez oprogramowanie. Ponieważ urządzenia sieciowe nie dysponują mechanizmami umożliwiającymi zlokalizowanie komputera na podstawie jego adresu IP, przed przesaniem ramki do sieci adres następnego skoku musi zostać zastąpiony odpowiednim adresem MAC.*

Zamiana adresu IP komputera na odpowiadający mu adres sprzętowy jest nazywana **odwzorowaniem adresu**. W operacji tej adres IP jest **odwzorowywany** na odpowiedni adres MAC. Odwzorowanie adresu ma zasięg lokalny. Dany komputer może odwzorować adres innego komputera tylko wtedy, gdy obydwie jednostki pracują w tej samej sieci fizycznej (dany komputer nigdy nie może ustalić adresu sprzętowego jednostki działającej w sieci zdalnej). Odwzorowanie adresu jest zawsze ograniczone do jednej sieci. Jako przykład przeanalizujmy sieć o konfiguracji przedstawionej na rysunku 23.1.



Rysunek 23.1. Przykład internetu złożonego z trzech sieci i przyłączonych do nich komputerów

Jeśli router R<sub>1</sub> zamierza przesyłać datagram do routera R<sub>2</sub>, musi odwzorować adres IP routera R<sub>2</sub> na właściwy adres MAC. Analogiczna zależność zachodzi między komputerami A i B. Aby aplikacja uruchomiona w stacji A mogła przesyłać dane do stacji B, oprogramowanie stosu protokołów stacji A musi odwzorować adres IP jednostki B na jej adres MAC. Dopiero po wykonaniu tej operacji możliwe jest przesyłanie danych.

Jeśli jednak komputer A przesyła dane komputera F (przyłączonego do sieci zdalnej), oprogramowanie jednostki A nie próbuje odwzorowywać adresów stacji F. Ustala jedynie, że pakiet musi zostać dostarczony do routera R<sub>1</sub> i poprzestaje na odwzorowaniu adresu tegoż routera. Gdy router R<sub>1</sub> odbierze dane, ustali, że następny skok wiedzie do routera R<sub>2</sub>. Wykona więc operację odwzorowania adresu węzła R<sub>2</sub>. W analogiczny sposób router R<sub>2</sub> uzyska adres jednostki F.

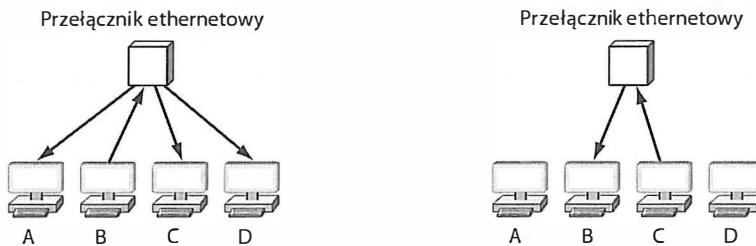
Podsumowując:

*Ustalenie powiązania między adresem protokołu a adresem sprzętowym jest nazywane **odwzorowaniem adresów**. Komputer lub router wykonują operację odwzorowania adresu, gdy muszą przesyłać pakiet do innej jednostki w ramach tej samej sieci fizycznej. Komputer nigdy nie próbuje odwzorowywać adresów jednostek spoza sieci własnej.*

### 23.3. Protokół odwzorowania adresu (ARP)

W jaki sposób oprogramowanie sieciowe odwzorowuje adres protokołu warstwy wyższej na adres sprzętowy? Odpowiedź zależy od protokołu oraz systemu adresowania w warstwie sprzętowej. W rozwiązaniach internetowych istotne jest jedynie odwzorowanie adresów IP. Większość urządzeń pracuje natomiast zgodnie ze specyfikacją Ethernet. Z tego powodu odwzorowywanie adresów jest zdominowane przez jedną technikę — wykorzystanie zaprojektowanego specjalnie na potrzeby Ethernetu **protokołu odwzorowania adresu** (ARP — ang. *Address Resolution Protocol*).

Zasada działania protokołu ARP nie jest szczególnie skomplikowana. Założymy, że komputer B musi zamienić adres IP komputera C na jego adres MAC. Wysyła w tym celu rozgłoszenie z informacją „poszukuję adresu MAC komputera posługującego się adresem IP o wartości C”. Rozgłoszenie jest rozsyłane tylko w obrębie danej sieci. Jeśli odbierze je komputer C, wygeneruje odpowiedź adresowaną bezpośrednio do jednostki B z informacją „jestem komputerem o adresie C, mój adres MAC to M”. Opisana wymiana komunikatów została również przedstawiona graficznie na rysunku 23.2.



Rysunek 23.2. Wymiana komunikatów ARP zainicjowana przez komputer B w celu odwzorowania adresu jednostki C

Z analizy rysunku wynika, że żądanie ARP dociera do wszystkich komputerów w sieci, natomiast odpowiedź jest przekazywana do wskazanej jednostki. Strona inicjująca uwzględnia w żądaniu informację, która jest weryfikowana przez wszystkie komputery w czasie przetwarzania żądania.

### 23.4. Format komunikatu ARP

Twórcy protokołu ARP nie ograniczyli jego zastosowań jedynie do odwzorowywania adresów IP w sieciach ethernetowych. Z tego względu komunikat nie ma wstępnie ustalonego formatu. Standard opisuje jedynie ogólną postać pakietu ARP i definiuje sposób dostosowywania formatu komunikatu do poszczególnych adresów oraz typów sieci. Uniezależnienie komunikatu ARP od warstwy sprzętowej wynika z tego, że autorzy rozwiązania uwzględnili możliwość zmiany technologii sieciowej, a tym samym rozmiaru adresu sprzętowego. Wprowadził więc pole o stałej długości w początkowej części pakietu, które zawiera informację o rozmiarze wykorzystywanego adresu sprzętowego. W przypadku protokołu ARP stosowanego w sieci Ethernet pole długości adresu ma wartość 6, ponieważ

adres ethernetowy składa się z 48 bitów, czyli 6 oktetów. Aby zagwarantować uniwersalność protokołu, jego twórcy wprowadzili również pole informujące o długości adresu protokołu warstwy wyższej (nie tylko adresu sprzętowego).

Warto więc zapamiętać, że mechanizm ARP nie służy jedynie do odwzorowywania adresów IP na określone adresy sprzętowe. Teoretycznie można go zastosować do zamiany dowolnego adresu warstwy wyższej na dowolny adres sprzętowy. W praktyce uniwersalność protokołu ARP jest rzadko wykorzystywana. Większość implementacji ma na celu powiązanie adresów IP z adresami ethernetowymi. Można więc stwierdzić, że:

*Mimo że format komunikatu ARP nadaje się do współdziałania z dowolnym protokołem warstwy wyższej i dowolnymi rodzajami adresów, mechanizm ARP jest niemal zawsze wykorzystywany do odwzorowywania adresów IP na 48-bitowe adresy ethernetowe.*

Na rysunku 23.3 przedstawiono format komunikatu ARP odpowiadający protokołowi IP w wersji 4 (którego adresy mają długość 4 oktetów) oraz ethernetowym adresom sprzętowym (o długości 6 oktetów). Każdy wiersz struktury reprezentuje 32 bity komunikatu ARP. Znaczenie poszczególnych pól zostało opisane w dalszej części punktu.

0	8	16	24	31			
Typ adresu sprzętowego		Typ adresu protokołu					
Długość adresu sprzętowego	Długość adresu protokołu	Operacja					
Adres sprzętowy nadawcy (4 pierwsze oktety)							
Adres sprzętowy nadawcy (2 ostatnie oktety)		Adres protokołu nadawcy (2 pierwsze oktety)					
Adres protokołu nadawcy (2 ostatnie oktety)		Adres sprzętowy celu (2 pierwsze oktety)					
Adres sprzętowy celu (4 ostatnie oktety)							
Adres protokołu celu (wszystkie 4 oktety)							

Rysunek 23.3. Format komunikatu ARP stosowany do odwzorowywania adresów IPv4 na adresy ethernetowe

**TYP ADRESU SPRZĘTOWEGO.** Jest to 16-bitowe pole, które informuje o rodzaju wykorzystywanego adresu sprzętowego. Ethernetowi odpowiada wartość 1.

**TYP ADRESU PROTOKOŁU.** Jest to 16-bitowe pole, które informuje o rodzaju adresu wykorzystywanego w protokole wyższej warstwy. Protokołowi IPv4 odpowiada wartość 0x0800.

**DŁUGOŚĆ ADRESU SPRZĘTOWEGO.** Jest to 8-bitowa liczba całkowita wyrażająca w bajtach rozmiar adresu sprzętowego.

**DŁUGOŚĆ ADRESU PROTOKOŁU.** Jest to 8-bitowa liczba całkowita wyrażająca w bajtach rozmiar adresu protokołu warstwy wyższej.

**OPERACJA.** To 16-bitowe pole służy do odróżniania żądań (wartość 1) od odpowiedzi (wartość 2).

**ADRES SPRZĘTOWY NADAWCY.** Jest to pole o długości określonej parametrem **DŁUGOŚĆ ADRESU SPRZĘTOWEGO**, które zawiera adres sprzętowy nadawcy komunikatu.

**ADRES PROTOKOŁU NADAWCY.** Jest to pole o długości określonej parametrem **DŁUGOŚĆ ADRESU PROTOKOŁU**, które zawiera adres protokołu wykorzystywany przez nadawcę.

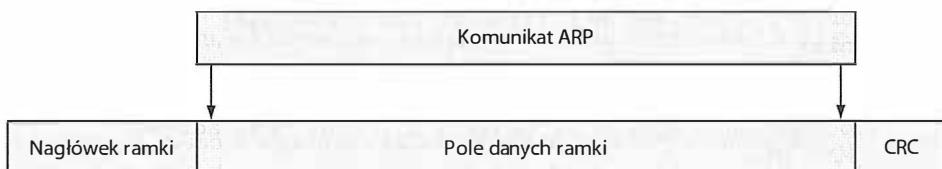
**ADRES SPRZĘTOWY CELU.** Jest to pole o długości określonej parametrem **DŁUGOŚĆ ADRESU SPRZĘTOWEGO**, które zawiera adres sprzętowy jednostki docelowej.

**ADRES PROTOKOŁU CELU.** Jest to pole o długości określonej parametrem **DŁUGOŚĆ ADRESU protokołu**, które zawiera adres protokołu wykorzystywany przez jednostkę docelową.

Jak wynika z analizy rysunku, komunikat ARP uwzględnia pola opisujące dwa powiązania. Pierwsze powiązanie dotyczy adresów nadawcy, a drugie odnosi się do adresów odbiorcy wywołań ARP, który w specyfikacji jest określony mianem **celu**. W czasie formowania żądania nadawca nie zna adresu sprzętowego celu (tę informację właśnie próbuje uzyskać). Zatem pole *adres sprzętowy celu* zostaje wypełnione zerami. W odpowiedzi powiązanie adresów celu odnosi się do komputera, który wysłał żądanie (zainicjował wymianę komunikatów). Para adresów celu dostarczana w odpowiedzi nie niesie więc żadnych informacji. Jest to pozostałość po wcześniejszej wersji tego protokołu.

## 23.5. Enkapsulacja ARP

Podczas przesyłania przez sieć komunikat ARP jest zapisywany w polu danych ramki sprzętowej. Podobnie jak w przypadku protokołu IP, informacja ARP jest traktowana jak każdy inny rodzaj transportowanych danych — urządzenia sieciowe nie analizują komunikatu ARP ani nie interpretują wartości jego pól. Mechanizm enkapsulacji wiadomości ARP w ramkach ethernetowych został pokazany na rysunku 23.4.



Rysunek 23.4. Enkapsulacja komunikatu ARP w ramce ethernetowej

Pole *typu danych* ramki odzwierciedla fakt transportowania komunikatu ARP. Nadawca musi zapisać w nim odpowiednią wartość przed wyemitowaniem ramki. Natomiast odbiorca jest zobowiązany do weryfikowania tych wartości w każdej nadchodzącej ramce. Identyfikator protokołu ARP w sieci Ethernet ma wartość 0x0806. Ten sam identyfikator

typu jest używany w żądaniach i odpowiedziach ARP. Nie można więc wykorzystać tej wartości do odróżnienia żądań od odpowiedzi. Odbiorca musi w tym względzie polegać na polu *operacja*.

## 23.6. Buforowanie ARP i przetwarzanie komunikatów

Mechanizm ARP musi być wykorzystywany do odwzorowywania adresów. Jednak generowanie żądania ARP przed każdym datagramem okazuje się bardzo nieefektywne — wysłanie jednego datagramu wymaga przesłania trzech ramek (żądania ARP, odpowiedzi ARP oraz samego datagramu). Poza tym większa część komunikacji między komputerami polega na transmisji sekwencji pakietów. W takich przypadkach nadawca musiałby ponawiać wymianę komunikatów ARP wielokrotnie.

Aby zmniejszyć natężenie ruchu, w programie obsługi protokołu ARP zaimplementowano mechanizm, który wyodrębnia i zapisuje informacje dostarczane w odpowiedzi w celu wykorzystania ich podczas wysyłania kolejnych pakietów. Dane te nie są przechowywane nieskończonie długo. Mechanizm ARP utrzymuje prostą tablicę powiązań, która pełni rolę **pamięci podręcznej** — składające się na nią wpisy są uaktualniane wraz z nadaniem odpowiedzi, a najstarszy jest usuwany w przypadku przepełnienia tablicy lub po upływie odpowiedniego czasu (na przykład dwóch minut). Gdy konieczne jest odwzorowanie adresu, moduł ARP najpierw przeszukuje pamięć podręczną. Jeśli odnajdzie w niej stosowny wpis, odczytuje adres MAC bez konieczności generowania żądania. Jeżeli jednak w buforze nie ma odpowiednich informacji, wysyła żądanie ARP, oczekuje na odpowiedź, uaktualnia pamięć podręczną i przekazuje pozyskany adres do dalszego wykorzystania.

Warto zauważyć, że w przeciwieństwie do większości mechanizmów buforowania pamięć ARP nie jest uaktualniana wraz z każdym odwołaniem do jej zawartości (tj. w przypadku odczytu). Aktualizacja wpisów następuje tylko w chwili odebrania komunikatu ARP z sieci (żądania lub odpowiedzi). Zarządzanie nadchodzącymi komunikatami ARP zostało opisane w algorytmie 23.1.

### Algorytm 23.1. Przetwarzanie nadchodzących informacji ARP

Dane:

Odebrany komunikat ARP (żądanie lub odpowiedź)

Zadanie:

Przetworzyć komunikat i zaktualizować pamięć podręczną ARP

Realizacja:

Wyodrębnienie adresu IP nadawcy (I) oraz jego adresu MAC (M).

if (adres I jest już zapisany w pamięci ARP) {

Zastąpienie adresu MAC w pamięci wartością M.

}

if (zażądzano odpowiedzi, a celem jest dany komputer) {

Dodanie nadawcy do pamięci ARP, jeśli odpowiadający mu wpis nie istniał wcześniej.

Przygotowanie i wysłanie odpowiedzi.

}

Jak wynika z algorytmu, mechanizm ARP wykonuje dwie operacje podczas przetwarzania nadchodzących komunikatów. Po pierwsze, odczytuje informacje na temat powiązania adresów nadawcy i uaktualnia nimi swoją pamięć podręczną (jeśli wpis na temat danego nadawcy został wcześniej zapisany w pamięci podręcznej). Uaktualnienie bufora obejmuje również przypadki, w których zmieniony został adres sprzętowy nadawcy. W drugim etapie analizowana jest wartość pola *operacja*, co pozwala na ustalenie, czy komunikat jest żądaniem, czy odpowiedzią na żądanie. Odebranie odpowiedzi oznacza, że wcześniej zostało wygenerowane żądanie i system oczekuje na informacje o powiązaniu adresów stacji docelowej. Jeśli komunikat jest żądaniem, odbiorca porównuje zawartość pola *adres protokołu celu* z lokalnym adresem protokołu warstwy wyższej. Jeśli wartości są jednakowe, dany komputer jest adresatem zapytania i musi odesłać odpowiedź ARP. Generowanie odpowiedzi sprowadza się do zamiany pól z informacjami o powiązaniu adresów nadawcy z polami adresów odbiorcy, wstawienia adresu sprzętowego w polu *adres sprzętowy nadawcy* oraz zmiany wartości pola *operacja* na 2 (oznaczającą odpowiedź).

Działanie mechanizmu ARP zostało dodatkowo zoptymalizowane — w przypadku odebrania żądania, na które musi zostać wygenerowana odpowiedź, komputer odczytuje dane na temat powiązania adresów nadawcy i zapisuje je w pamięci podręcznej z przeznaczeniem do późniejszego wykorzystania. Usprawnienie staje się oczywiste, gdy uwzględni się dwa poniższe fakty:

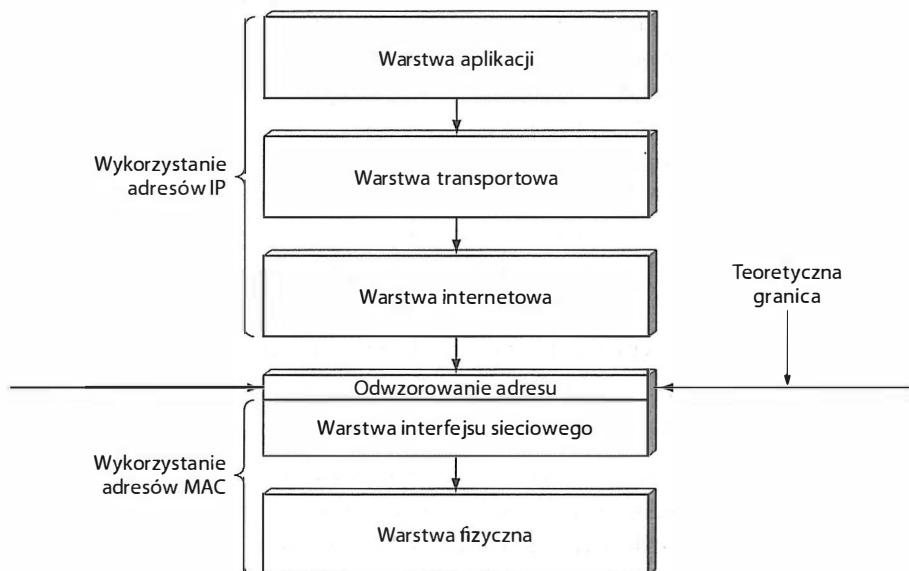
- Większość przypadków komunikacji między komputerami ma charakter dwustronny — jeśli wiadomość jest przekazywana z A do B, to istnieje duże prawdopodobieństwo, że będzie konieczne przesłanie odpowiedzi z B do A.
- Komputer nie może zapisywać dowolnie wielu informacji na temat powiązania adresów z uwagi na ograniczenie rozmiaru dostępnej pamięci.

Pierwsza właściwość stanowi odpowiedź na pytanie o to, dlaczego zapamiętywanie powiązania między adresami nadawcy zwiększa wydajność mechanizmu ARP. Komputer A wysyła żądanie ARP do jednostki B tylko wtedy, gdy ma również pakiet do dostarczenia. Jest więc bardzo prawdopodobne, że gdy komputer B odbierze pakiet ze stacji A, będzie musiał odesłać informacje zwrotne. Dzięki temu, że zapisze informacje na temat powiązania między adresami komputera A już w czasie przetwarzania żądania ARP, nie będzie musiał wykonywać tej operacji później, podczas przesyłania zasadniczego pakietu.

Drugi fakt wyjaśnia, dlaczego nowe wpisy są dodawane do pamięci podręcznej ARP tylko w przypadku, w którym dany komputer jest celem żądania ARP, a nie gdy żądania są dostarczane do innych komputerów. Gdyby wszystkie jednostki rejestrowały wszystkie informacje na temat powiązań między adresami, ich pamięci podręczne szybko uległyby wyczerpaniu, mimo że większość z nich nigdy nie rozpoczęłaby wymiany danych z innymi stacjami sieci. Mechanizm ARP rejestruje jedynie te adresy, które prawdopodobnie będą potrzebne w przyszłości.

### 23.7. Teoretyczna granica stosowania adresów

Z informacji zamieszczonych w rozdziale 1. wiadomo, że model TCP/IP składa się z pięciu warstw. Odwzorowanie adresów jest operacją powiązaną z warstwą interfejsu sieciowego (tj. z warstwą drugą). Mechanizm ARP wprowadza pewien teoretyczny podział między obszarami zastosowania adresów MAC i adresów IP. Ukrywa w pewien sposób szczegóły adresowania sprzętowego i umożliwia wyższym warstwom oprogramowania korzystanie z adresów IP. Wyznacza więc granicę między warstwą interfejsu sieciowego a wyższymi warstwami stosu protokołów — aplikacje oraz oprogramowanie wyższych warstw są przystosowane do posługiwania się adresami protokołów. Graficzna reprezentacja opisanego podziału została przedstawiona na rysunku 23.5.



Rysunek 23.5. Granica między obszarami zastosowania adresów IP a adresów MAC

Ważne, aby zapamiętać, że:

*Mechanizm ARP wyznacza teoretyczną granicę w ramach stosu protokołów — warstwy powyżej tej granicy posługują się adresami IP, natomiast warstwy niższe korzystają z adresów MAC.*

### 23.8. Internetowy protokół komunikatów sterujących (ICMP)

Wiadomo, że specyfikacja IP definiuje komunikację typu best-effort, czyli taką, w której datagramy mogą być tracone, duplikowane, opóźniane lub dostarczane w niewłaściwej kolejności. Mogłoby się więc wydawać, że w działaniu takiego mechanizmu nie jest

potrzebna żadna kontrola błędów. Trzeba jednak pamiętać, że usługi typu best-effort nie są rozwiązaniami zaprojektowanymi w sposób niedbały. Mechanizmy IP starają się unikać błędów i zgłaszać ewentualne problemy. Tak naprawdę, jeden przykład detekcji błędów w protokole IP został już opisany — jest nim sprawdzanie błędów transmisyjnych za pomocą sumy kontrolnej nagłówka IP. Gdy komputer tworzy datagram IP, uwzględnia w nim wartość kontrolną obejmującą wszystkie informacje zawarte w nagłówku. Wartość ta jest weryfikowana przez jednostkę odbierającą datagram w celu sprawdzenia, czy nie doszło do przekłamań w transmisji nagłówka.

Pakiet IP zawiera również pole *TTL* (czasu życia), które zapobiega krążeniu datagramu w sieci w przypadku wystąpienia pętli routingu.

Działania podejmowane po stwierdzeniu niewłaściwej wartości sumy kontrolnej nie są skomplikowane — datagram jest natychmiast odrzucany bez jakiegokolwiek dalszego przetwarzania. Odbiorca nie może polegać na żadnej wartości zapisanej w nagłówku, ponieważ nie wiadomo, które bity zostały zmienione. Nie może nawet odesłać komunikatu o błędzie, gdyż nie ma pewności, że adres nadawcy jest poprawny. Odbiorcy nie pozostaje nic innego, jak tylko odrzucić pakiet.

Do rozwiązywania nieco mniej poważnych problemów służy **internetowy protokół komunikatów sterujących** (ICMP — ang. *Internet Control Message Protocol*). Towarzyszy on protokołowi IP i ma na celu informowanie źródła danych (komputera, który wysłał datagram) o błędach w transmisji. Co ciekawe, protokoły IP i ICMP są wzajemnie zależne — działanie IP zależy od błędów zgłaszanych przez protokół ICMP, natomiast protokół ICMP wykorzystuje IP do przenoszenia komunikatów o błędach.

Mimo że specyfikacja protokołu ICMP obejmuje ponad dwadzieścia typów komunikatów, wykorzystywanych jest tylko kilka z nich. Lista najważniejszych wraz z krótkim opisem została przedstawiona w tabeli 23.1.

**Tabela 23.1.** Przykłady komunikatów ICMP wraz z ich numerami i opisem przeznaczenia

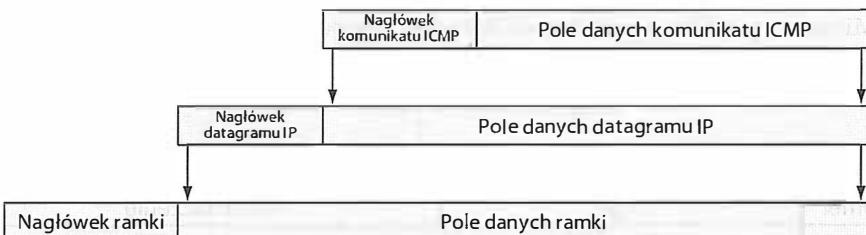
Numer	Typ	Przeznaczenie
0	Odpowiedź echa ( <i>echo reply</i> )	Wykorzystywany przez polecenie <code>ping</code>
3	Cel nieosiągalny ( <i>destination unreachable</i> )	Nie można dostarczyć datagramu
5	Przekierowanie ( <i>redirect</i> )	Wymuszenie na komputerze zmiany trasy
8	Żądanie echa ( <i>echo request</i> )	Wykorzystywany przez polecenie <code>ping</code>
11	Przekroczenie czasu ( <i>time exceeded</i> )	Przekroczona wartość TTL lub upłynął czas dostarczania fragmentów
12	Błąd parametrów ( <i>parameter problem</i> )	Nagłówek IP został niepoprawnie sformowany
30	Śledzenie trasy ( <i>traceroute</i> )	Wykorzystywany przez polecenie <code>traceroute</code>

Z analizy zestawienia wynika, że protokół ICMP dostarcza komunikaty dwojakiego rodzaju — powiadomienia o błędach oraz komunikaty informacyjne. Na przykład pakiety

typu *przekroczenie czasu* lub *cel nieosiągalny* są powiadomieniami o błędach, które występują wówczas, gdy datagram nie może zostać poprawnie dostarczony do odbiorcy. Jednostka docelowa jest nieosiągalna, jeśli żadna trasa nie prowadzi pod podany adres. Z kolei przekroczenie czasu występuje wtedy, gdy licznik TTL osiągnie wartość 0 lub kolejne fragmenty datagramu nie zostaną dostarczone w czasie krótszym niż przewidziany na odtworzenie pakietu. Z drugiej strony, takie komunikaty jak *żądanie echo* i *odpowiedź echo* nie są powiadomieniami o błędach. Są natomiast wykorzystywane przez polecenie ping do sprawdzania połączenia między stacjami. Urządzenie, które odbierze żądanie echo, odsyła odpowiedź zawierającą te same dane, które zostały przekazane w żądaniu. Polecenie ping wysyła więc żądanie do zdalnej jednostki, czeka na odpowiedź, a następnie wyświetla informację o tym, czy wskazana stacja jest dostępna w sieci, czy nie (o niedostępności decyduje brak odpowiedzi w ustalonym czasie).

### 23.9. Format komunikatu i enkapsulacja ICMP

Mechanizm ICMP wykorzystuje do przekazywania komunikatów protokół IP. Gdy router musi wysłać wiadomość ICMP, tworzy datagram IP, a następnie zapisuje w jego polu danych treść powiadomienia ICMP. Sam datagram jest przesyłany przez sieć w standardowy sposób, czyli podlega enkapsulacji w ramce transmisyjnej. Obydwa poziomy enkapsulacji przedstawiono na rysunku 23.6.



Rysunek 23.6. Dwa poziomy enkapsulacji podczas wysyłania komunikatu ICMP

Datagramy przenoszące powiadomienia ICMP nie mają szczególnych priorytetów. Są dostarczane na standardowych zasadach, ale z pewnym wyjątkiem. Jeśli podczas przesyłania komunikatu ICMP o błędzie również wystąpi błąd, jednostka nadawcza nie zostanie o tym fakcie poinformowana. Powód wydaje się oczywisty. Twórcy protokołu chcieli uniknąć przeciążenia sieci z powodu transmisji komunikatów o błędach.

*Protokół ICMP przenosi zarówno powiadomienia o błędach, jak i komunikaty informacyjne. Transport powiadomień wymaga użycia protokołu IP, który z kolei wykorzystuje mechanizm ICMP do raportowania błędów.*

## 23.10. Oprogramowanie, parametry i konfiguracja protokołu

Dotychczasowe rozważania na temat protokołów internetowych odnosiły się do sytuacji, w których komputer lub router były uruchomione, działał system operacyjny, a oprogramowanie protokołu zostało odpowiednio zainicjowane. Nasuwa się jednak pytanie, w jaki sposób oprogramowanie protokołu rozpoczyna swoje działanie w komputerze lub routerze. W przypadku routera odpowiedź jest prosta. Administrator sieci musi określić wartości początkowe takich parametrów jak adres IP (każdego połączenia sieciowego) i rodzaj protokołu, a także wstępne wartości poszczególnych wpisów tablicy routingu. Konfiguracja jest zapisywana na dysku, z którego jest odczytywana w czasie uruchamiania systemu.

Konfiguracja komputera jest znacznie bardziej skomplikowana i zazwyczaj wymaga wykonania dwóch zadań. Pierwsze z nich jest realizowane w czasie uruchamiania komputera. System operacyjny ustala pewne parametry konfiguracyjne, które umożliwiają oprogramowaniu protokołu komunikację w ramach sieci lokalnej. W drugim etapie oprogramowanie protokołu pozyskuje dodatkowe informacje na temat adresu IP komputera, maski podsieci oraz adresów lokalnego serwera DNS. Wspomniane oprogramowanie ma charakter **parametryzowanego** obrazu binarnego, którego uruchomienie wiąże się z wprowadzeniem pewnego zbioru ustawień. Ten sam obraz binarny można wykorzystywać w wielu komputerach. Nie trzeba go również zmieniać wraz ze zmianą połączenia sieciowego. Oprogramowanie protokołu można **konfigurować**, dostosowując je do określonej sytuacji.

*Oprogramowanie protokołu jest parametryzowane, aby umożliwić uruchamianie skompilowanej wersji obrazu na wielu komputerach działających w różnych środowiskach sieciowych. Po uruchomieniu kopii oprogramowania w danym komputerze trzeba je skonfigurować, wprowadzając informacje na temat komputera oraz sieci, do których jest przyłączony.*

## 23.11. Protokół dynamicznej konfiguracji stacji (DHCP)

Za pozyskiwanie przez komputer parametrów sieciowych odpowiada wiele różnych mechanizmów. Jako jedno z pierwszych rozwiązań tego typu został opracowany **protokół odwrotnego odzworowania adresów** (RARP — ang. *Reverse Address Resolution Protocol*). Umożliwia on uzyskanie adresu IP z serwera RARP. W opisany wcześniej mechanizmie ICMP uwzględniono komunikaty *żądania maski* oraz *wykrywania routera*, które umożliwiają pobranie informacji o zastosowanej masce podsieci oraz o adresie routera. Każde z tych rozwiązań było wykorzystywane niezależnie. Żądania były wysyłane rozgłoszeniowo, a konfiguracja komputerów przebiegała od najniższej warstwy do warstw wyższych.

Wraz z rozwojem protokołów internetowych wprowadzono pojedynczy mechanizm dostarczania wielu parametrów (wymagający wysłania tylko jednego żądania) — **protokół**

**uruchomieniowy** (BOOTP — ang. *Bootstrap Protocol*). Dostarczał on adres IP komputera, maskę podsieci oraz adres domyślnego routera. Dzięki niemu dana stacja sieciowa mogła w jednym kroku uzyskać wszystkie informacje, które są potrzebne do skonfigurowania stosu IP.

Podobnie jak inne protokoły konfiguracyjne BOOTP wymagał od komputera wysłania rozgłoszeniowego żądania. Jednak w przeciwieństwie do innych rozwiązań komunikacja z serwerem bazowała na protokole IP — żądanie było wysyłane z adresem docelowym złożonym z samych jedynek logicznych oraz adresem źródłowym odpowiadającym samym zerom logicznym. Serwer BOOTP wykorzystywał adres MAC odbieranej ramki do skierowania odpowiedzi do właściwego komputera. Dzięki temu komputer nieposiadający adresu IP mógł się komunikować z serwerem BOOTP.

W początkowej wersji mechanizmu BOOTP przydział adresów był stały. Serwer dysponował bazą danych adresów IP, które przydzielał określonym stacjom. Żądania generowane przez stacje sieciowe zawierały niepowtarzalny identyfikator (zazwyczaj adres MAC komputera), na którego podstawie serwer odszukiwał właściwy adres IP. Problemem było jednak to, że działanie systemu BOOTP wymagało ingerencji administratora — zanim komputer mógł skorzystać z mechanizmu BOOTP do pobrania adresu, administrator sieci musiał wpisać w serwerze adresy IP znanych jednostek.

Rozwiązanie sprawdzało się w działaniu, dopóki liczba komputerów pozostawała niezmienna. Ręczne wprowadzanie adresów nie pozwalało jednak na automatyczne dostosowywanie się systemu do nagłych zmian. Doskonałym przykładem ograniczeń opisywanego mechanizmu jest sieć w kawiarence, która umożliwia korzystanie z internetu przypadkowym użytkownikom. Aby wyeliminować tego typu problemy, organizacja IETF rozszerzyła zakres funkcjonalny protokołu BOOTP, zmieniając jednocześnie jego nazwę na **protokół dynamicznej konfiguracji stacji** (DHCP — ang. *Dynamic Host Configuration Protocol*).

Mechanizm DHCP zapewnia automatyczny przydział adresów IP dowolnym komputerom przyłączającym się do sieci. Jest w pewnym sensie adaptacją rozwiązań **plug-and-play** w środowisku sieciowym.

*Protokół DHCP umożliwia przyłączającemu się do sieci komputerowi pobranie parametrów konfiguracyjnych bez konieczności wprowadzania zmian w bazie danych przez administratora.*

Działanie protokołu DHCP jest bardzo zbliżone do pracy mechanizmu BOOTP. W czasie uruchamiania komputer wysyła w sposób rozgłoszeniowy **żądanie DHCP**, na które serwer reaguje, przesyłając **odpowiedź DHCP**<sup>66</sup>. Administrator ma możliwość skonfigurowania serwera DHCP w taki sposób, aby dostarczał on adresy na podstawie statycznego wpisu w bazie danych (tak jak w przypadku BOOTP) lub pobierane z puli adresów dynamicznych (przydzielanych na żądanie). Zazwyczaj stałe adresy są przypisywane serwerom,

<sup>66</sup> W specyfikacji DHCP informacja zwrotna z serwera jest nazywana **ofertą** — serwer **oferuje** klientowi określony adres IP.

natomiast jednostki klienckie korzystają z adresów dynamicznych. Adresy przydzielane na żądanie nie mają jednak nieograniczonego czasu obowiązywania. Mechanizm DHCP przydziela je na pewien okres, nazywany czasem **dzierżawy**<sup>67</sup>. Idea dzierżawy umożliwia serwerowi DHCP odzyskiwanie nieużywanych od pewnego czasu wartości adresów. Gdy upłynie czas dzierżawy, serwer przenosi adres do puli dostępnych, co pozwala na przydzielanie go kolejnemu komputerowi klienckiemu. Dzierżawy mają bardzo duże znaczenie w ciągłej pracy serwera, ponieważ pozwalają na kontrolowanie zasobów i odzyskiwanie adresów również w przypadkach, w których komputery posługujące się nimi zostaną niespodziewanie (lub w wyniku awarii) wyłączone.

Z chwilą wygaśnięcia dzierżawy komputer może zrzec się adresu lub negocjować z serwerm DHCP jej przedłużenie. Negocjacje są realizowane równolegle z innymi działaniami. Zazwyczaj serwery DHCP akceptują przedłużenie dzierżawy, dzięki czemu komputer może pracować dalej bez przerywania pracy aplikacji lub innych systemów komunikacji sieciowej. Czasami jednak serwer odmawia przedłużenia dzierżawy, co jest wynikiem działań administratora, który z powodów technicznych lub administracyjnych wyłącza funkcję przydziału adresów. Jako przykład można sobie wyobrazić sieć w auli uniwersyteckiej. Serwer uczelni może być tak skonfigurowany, aby dzierżawy adresów kończyły się wraz z końcem zajęć (dzięki temu na kolejnych zajęciach adresy mogą być przydzielone następnym komputerom). Nad działaniem mechanizmu DHCP całkowitą kontrolę sprawuje serwer — jeśli serwer zabroni przedłużenia dzierżawy, stacja kliencka musi przestać korzystać z adresu.

## 23.12. Działanie protokołu DHCP i optymalizacja pracy

Choć działanie protokołu jest niezbyt skomplikowane, uwzględnia kilka czynników, które pozwalają na optymalizację pracy sieci. Najważniejsze trzy to:

- obsługa utraty lub powielenia pakietu,
- buforowanie adresów serwerów,
- unikanie jednoczesnych zapytań.

Po pierwsze, protokół DHCP został zaprojektowany w taki sposób, że utrata lub powielenie pakietów nie skutkuje błędną konfiguracją stacji. Jeśli komputer nie odbierze odpowiedzi, retransmituje żądanie, a jeśli odbierze duplikat odpowiedzi, ignoriuje go. Po drugie, jednostki, które odszukały serwer za pomocą komunikatu wykrywania, rejestrują adres serwera w pamięci podręcznej. Dzięki temu odnawianie dzierżawy jest znacznie efektywniejsze.

Po trzecie, mechanizm DHCP uwzględnia zabezpieczenia przed synchronizacją żądań. Sytuacja synchronizacji żądań może wystąpić w przypadku jednoczesnego włączenia wszystkich komputerów w sieci po awarii zasilania. Aby uniknąć ryzyka zalania serwera żdaniami, wprowadzono obowiązek losowania po stronie klienckiej pewnej wartości czasu oczekiwania przed wysłaniem (lub ponowieniem) żądania.

---

<sup>67</sup> Administrator określa czas dzierżawy podczas definiowania puli adresowej.

### 23.13. Format komunikatu DHCP

Z uwagi na to, że protokół DHCP powstał na bazie protokołu BOOTP, format komunikatu DHCP nieznacznie różni się od komunikatu BOOTP. Struktura komunikatu DHCP została pokazana na rysunku 23.7.

0	8	16	24	31
Operacja	Typ sprzętu	Długość adresu	Węzły	
Identyfikator transakcji				
Liczba sekund				Znaczniki
Adres IP klienta				
Proponowany adres IP				
Adres IP serwera				
Adres IP routera				
Adres sprzętowy klienta (16 oktetów)				
:				
Nazwa serwera (64 oktetów)				
:				
Nazwa pliku rozruchowego (128 oktetów)				
:				
Opcje (zmienny rozmiar)				
:				

Rysunek 23.7. Format komunikatu DHCP

Poza polem *opcji* wszystkie wartości w komunikacie DHCP mają określony stały rozmiar. Siedem pierwszych pól zawiera informacje potrzebne do przetworzenia samego komunikatu. Parametr *operacja* wskazuje, czy komunikat jest **żądaniem**, czy **odpowiedzią**. Do ustalenia konkretnego typu komunikatu (określenia, czy komunikat opisuje poszukiwanie serwera, żądanie przydziału adresu, czy jest potwierdzeniem przydziału lub odrzucenia adresu) służy pole *typ wiadomości* zawarte w sekcji *opcji*. Wartość pola *operacja* informuje jedynie o tym, czy pakiet został przesłany z klienta do serwera, czy z serwera do klienta.

Parametry *typ sprzętu* oraz *długość adresu* odzwierciedlają rodzaj urządzenia sieciowego oraz długość adresu sprzętowego danej jednostki. Za pomocą pola *znaczniki* klient informuje serwer o tym, czy może odbierać rozgłoszenia, czy wymaga bezpośredniego dostarczenia odpowiedzi. Wartość *węzły* odpowiada liczbie serwerów, które przekierowały dane żądanie. Natomiast *identyfikator transakcji* jest polem, na którego podstawie klient może dopasować odpowiedź do wygenerowanego żądania. Wartość *liczba sekund* informuje o czasie, który upłynął od rozpoczęcia procedury uruchamiania komputera. Jeśli klient dysponuje adresem IP (adres ten został przydzielony przez inny system niż DHCP), wypełnia w żądaniu pole *adres IP klienta*.

Kolejne parametry są wykorzystywane w odpowiedzi i niosą informacje, na które oczekuje nadawca żądania. Jeśli klient nie dysponuje adresem IP, serwer wypełnia pole *propozycja adresu IP*. Dodatkowo definiuje wartości *adres IP serwera* oraz *nazwa serwera*, aby dostarczyć komputerowi informacje na temat własnej lokalizacji. Pole *adres IP routera* zawiera adres IP domyślnego routera.

Poza konfiguracją protokołu mechanizm DHCP umożliwia komputerom pozyskiwanie informacji o miejscu składowania obrazu startowego. Jeśli taka informacja jest potrzebna danej stacji sieciowej, wypełnia ona w żądaniu pole *nazwa pliku rozruchowego* (może w ten sposób na przykład wskazać system operacyjny LINUX). Serwer nie odsyła jednak pliku obrazu, ale ustala położenie tego pliku i wykorzystuje pole *nazwa pliku rozruchowego* do określenia nazwy obrazu. Samo pobranie pliku obrazu wymaga zastosowania przez jednostkę kliencką innego protokołu (na przykład TFTP).

## 23.14. Pośrednictwo w dostępie do serwera DHCP

Mimo że odwołania do serwera DHCP mają charakter rozgłoszeniowy, administrator nie musi instalować osobnego serwera w każdej sieci. Może natomiast skorzystać z usług **pośrednika DHCP** (ang. *DHCP relay agent*), który przekaże żądania klienckie i odpowiedzi serwera do odpowiednich jednostek. W każdej sieci musi pracować co najmniej jeden pośrednik i musi on dysponować adresem IP odpowiedniego serwera DHCP. Odpowiedź serwera jest przekazywana przez pośrednika do stacji klienckiej.

Mogłoby się wydawać, że konieczność stosowania wielu pośredników DHCP nie jest pod żadnym względem korzystniejsza od uruchamiania wielu serwerów DHCP. Administratorzy wolą jednak korzystać z pośredników. Są ku temu dwa powody. Po pierwsze, w sieci o wielu pośrednikach i jednym serwerze DHCP zarządzanie adresami jest skupione w jednym urządzeniu. Administrator nie musi więc korzystać z kilku systemów, aby wdrożyć nową politykę dzierżawy lub sprawdzić bieżący stan sieci. Po drugie, wiele komercyjnych routerów zawiera mechanizmy, które umożliwiają przekazywanie żądań ze wszystkich przyłączonych do routera sieci, a ich konfiguracja nie nastręcza większych trudności (sprowadza się do włączenia odpowiedniej funkcji i podania adresu IP serwera) i nie zmienia się zbyt często.

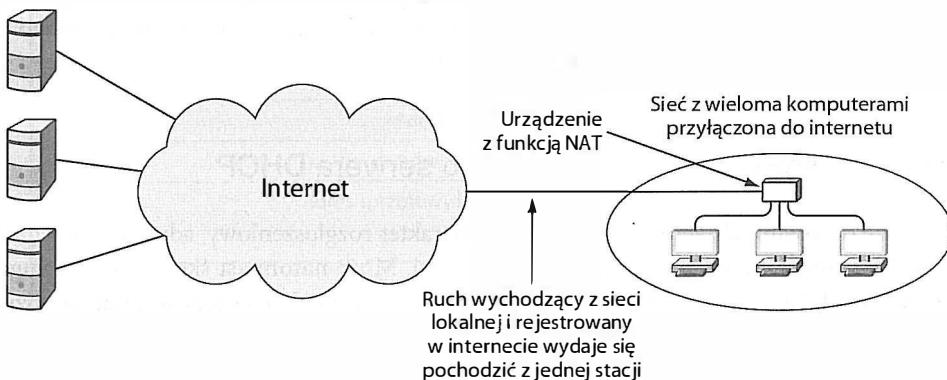
## 23.15. Translacja adresów sieciowych (NAT)

Wraz z rozwojem internetu adresy stają się coraz bardziej deficytowym towarem. Jednym ze sposobów ich zaoszczędzenia jest wydzielanie podsieci oraz adresowanie bezklaśowe (CIDR)<sup>68</sup>. Opracowano również trzeci sposób, który pozwala wielu komputerom na korzystanie z jednego globalnego adresu IP. Jest nim technika **translacji adresów sieciowych** (NAT — ang. *Network Address Translation*). Zapewnia przejrzystą komunikację, czyli taką, w której każdy komputer w sieci lokalnej korzysta bez przeszkód z połączenia

<sup>68</sup> Więcej informacji na temat podsieci i adresowania bezklasowego znajduje się w rozdziale 21.

internetowego, a ruch sieciowy rejestrowany w jednostkach internetowych wydaje się zawsze pochodzić z jednego komputera, a nie z wielu jednostek. Stacja pracująca w sieci lokalnej posługuje się standardowym oprogramowaniem TCP/IP i wymienia w standar-dowy sposób informacje z komputerami funkcjonującymi w internecie.

Usługa NAT jest usługą wtrąconą (ang. *in-line*). Oznacza to, że jest realizowana przez urządzenie pośredniczące w transmisji danych między siecią lokalną a internetem. Choć teoretycznie operacje NAT nie zależą w żaden sposób od innych funkcji i usług, większość rozwiązań tego typu implementuje się w punktach dostępowych Wi-Fi lub w routerach wewnętrznych. Na rysunku 23.8 przedstawiono typową konfigurację sieci z funkcją NAT.



Rysunek 23.8. Teoretyczna architektura sieci z funkcją NAT

### 23.16. Działanie usługi NAT i adresy prywatne

Celem usługi NAT jest przesłonięcie sieci wewnętrznej w taki sposób, aby z perspektywy jednostki internetowej wydawała się pojedynczym komputerem o poprawnym adresie IP — wszystkie wysypane z sieci datagramy powinny sprawiać wrażenie wyemitowanych przez pojedynczy komputer, a wszystkie dostarczane do sieci datagramy powinny być przesy-łane tak, jakby były transmitowane do pojedynczej stacji. Z kolei użytkownikowi stacji wewnętrznej powinno się wydawać, że w internecie akceptowane są adresy prywatne, a zawierające je pakiety podlegają klasycznym mechanizmom routingu.

Oczywiście, nie można przypisać pojedynczego adresu wielu komputerom. Jeśli dwa komputery lub większa ich liczba otrzymają ten sam adres, wystąpi konflikt wynikający z tego, że na pojedyncze zapytanie ARP generowane będą wiele odpowiedzi. Z tego względu każdy komputer pracujący w sieci musi mieć niepowtarzalny adres IP. W przy-padku usługi NAT rozwiązaniem problemu jest użycie dwóch rodzajów adresów. Urzą-dzenie z funkcją NAT dysponuje pojedynczym globalnie akceptowalnym adresem IP, tak jak każda inna stacja internetowa. Komputery w sieci lokalnej posługują się natomiast **prywatnymi adresami** (nazywanymi również **adresami nieroutowanymi**). W tabeli 23.2 przedstawiono zakresy adresów IP zarezerwowane przez organizację IETF do użytku prywatnego.

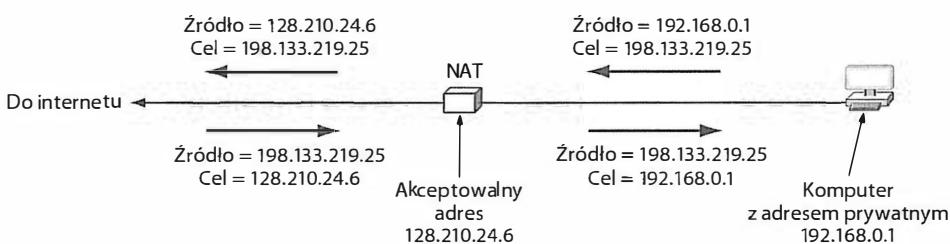
**Tabela 23.2.** Zakresy prywatnych (nieroutowalnych) adresów IP, wykorzystywanych w mechanizmie NAT

Zakres	Opis
10.0.0.0/8	Zakres prywatnych adresów IP w klasie A
169.254.0.0/16	Zakres prywatnych adresów IP w klasie B
172.16.0.0/12	Ciągły przedział 16 zakresów adresów IP z klasy B
192.168.0.0/16	Ciągły przedział 256 zakresów adresów IP z klasy C

Jako przykład przeanalizujmy sieć, w której urządzenie NAT współdziała z komputerami o adresach z przedziału zarezerwowanego do użytku prywatnego o wartości 192.168.0.0. W celu zapewnienia niepowtarzalności adresów (i uniknięcia konfliktów) stacjom przydzielono adresy 192.168.0.1, 192.168.0.2 itd.

Niestety, prywatnych adresów nie można używać do przesyłania danych w internecie — routery internetowe są skonfigurowane w taki sposób, aby odrzucały datagramy zawierające adresy nieroutowalne. Adresacja prywatna obowiązuje więc jedynie w sieci wewnętrznej. Zanim datagram opuści sieć lokalną, urządzenie z funkcją NAT musi zamienić prywatny adres IP na adres IP obowiązujący w internecie. Analogiczna operacja musi zostać wykonana podczas przekazywania datagramu do jednostki w sieci lokalnej. Wówczas jednak translacja polega na zamianie globalnego adresu IP na adres prywatny.

Działanie usługi NAT w podstawowej formie sprowadza się do tłumaczenia źródłowych adresów w chwili przekazywania datagramów z sieci lokalnej do internetu oraz docelowych adresów podczas przenoszenia datagramów z internetu do sieci lokalnej. Założymy, że urządzenie z funkcją NAT posługuje się adresem globalnym o wartości 128.210.24.6. Przeanalizujmy proces przesyłania datagramu z komputera o adresie 192.168.0.1 do jednostki o adresie 198.133.219.25. Operacja tłumaczenia adresów realizowana podczas transmisji w obydwu kierunkach została przedstawiona graficznie na rysunku 23.9.



**Rysunek 23.9.** Podstawowa forma translacji NAT, która zapewnia tłumaczenie źródłowego adresu datagramów wychodzących oraz zamianę docelowego adresu datagramów przychodzących

Podsumowując:

*Podstawowe działanie usługi NAT polega na modyfikowaniu źródłowych adresów IP w datagramach przesyłanych z sieci wewnętrznej do internetu oraz zamianie docelowych adresów IP w datagramach przekazywanych z internetu do sieci wewnętrznej.*

Większość implementacji usługi NAT wykorzystuje **tablicę translacji** do przechowywania informacji niezbędnych do przepisania adresu. Przykładowa tablica translacji adresów, odpowiadająca przykładowi pokazanemu na rysunku 23.9, została przedstawiona w tabeli 23.3.

**Tabela 23.3.** Tablica translacji adresów  
zawierająca odwzorowania właściwe dla przykładu z rysunku 23.9

Kierunek	Pole	Poprzednia wartość	Nowa wartość
Wychodzący	Źródłowy adres IP	192.168.0.1	128.210.24.6
	Docelowy adres IP	198.133.219.25	bez zmian
Przychodzący	Źródłowy adres IP	198.133.219.25	bez zmian
	Docelowy adres IP	128.210.24.6	192.168.0.1

W jaki sposób poszczególne wartości są zapisywane w tablicy translacji? Choć może je tam wpisać administrator systemu, zazwyczaj są one automatycznie rejestrowane przez mechanizm NAT — oprogramowanie NAT dodaje do tablicy translacji odpowiedni wpis za każdym razem, gdy komputer z sieci wewnętrznej przesyła pakiet do internetu. Na przykład gdy urządzenie przekazuje pierwszy pakiet przesyłany z komputera o adresie 192.168.0.1 do jednostki o adresie 198.133.219.25, dodaje odpowiedni wpis do tablicy translacji. Dzięki temu gdy odbierze odpowiedź z jednostki 198.133.219.25, może odszukać wcześniejszy wpis i na jego podstawie przetłumaczyć adres docelowy na 192.168.0.1.

### 23.17. Translacja NAT na poziomie warstwy transportowej (NAPT)

Przedstawiona w poprzednim punkcie operacja NAT działa poprawnie jedynie w przypadkach, w których każdy komputer w sieci lokalnej komunikuje się z innym serwerem internetowym. Jeśli dwie jednostki lokalne spróbują odwołać się do jednego serwera zdalnego ( $X$ ), w tablicy translacji adresów zostanie zarejestrowana większa liczba wartości  $X$  i usługa NAT nie będzie mogła wyznaczyć odpowiedniej stacji docelowej dla datagramów wchodzących do sieci. Rozwiążanie to nie sprawdza się również w sytuacjach, w których więcej aplikacji działających w jednym komputerze próbuje się jednocześnie komunikować z różnymi stacjami internetowymi.

Najczęściej wykorzystywana wersja usługi NAT rozwiązuje obydwa problemy — umożliwia dowolnej liczbie aplikacji uruchomionych na dowolnej liczbie komputerów lokalnych komunikowanie się z dowolną liczbą komputerów działających w internecie. Choć z technicznego punktu widzenia mechanizm ten nazywa się **translację adresów sieciowych i portów** (NAPT — ang. *Network Address and Port Translation*), jest tak często stosowany, że nawet doświadczeni inżynierowie utożsamiają określenie NAT z funkcją NAPT.

Kluczem do zrozumienia zasady działania funkcji NAPT jest przypomnienie sobie, że w celu rozróżnienia usług sieciowych aplikacje posługują się **numerami portów**. War-

tości te są charakterystyczne dla protokołów UDP i TCP, które zostały opisane w rozdziałach 25. i 26. Poza przechowywaniem źródłowych i docelowych adresów IP usługa NAPT rejestruje również numery portów i kojarzy datagramy ze strumieniami TCP bądź UDP. Jej działanie nie ogranicza się więc do warstwy IP, ale obejmuje również przetwarzanie nagłówków warstwy transportowej. W rezultacie tablica translacji adresów mechanizmu NAPT składa się z czteroelementowych wierszy, uwzględniających źródłowe i docelowe adresy IP oraz źródłowe i docelowe numery portów.

Przeanalizujmy zawartość tablicy translacji na przykładzie, w którym przeglądarka komputera o adresie 192.168.0.1 oraz przeglądarka uruchomiona w systemie o adresie 192.168.0.2 wykorzystują port 30000. Założymy, że obydwie ustanawiają połączenie TCP z serwerem WWW pracującym na porcie 80. W przekazywaniu pakietów pośredniczy urządzenie z funkcją NAPT, które korzysta z adresu 128.10.24.6. Aby uniknąć konfliktu portów, jednostka NAPT musi wybrać inny źródłowy TCP realizowanych połączeń. Jeden ze sposobów rozwiązania problemu został zademonstrowany w tabeli 23.4.

Tabela 23.4. Przykład translacji NAPT w przypadku dwóch połączeń z jednym serwerem

Kierunek	Pole	Poprzednia wartość	Nowa wartość
Wychodzący	Źródłowy adres IP : źródłowy port TCP	192.168.0.1:30000	128.10.24.6:40001
Wychodzący	Źródłowy adres IP : źródłowy port TCP	192.168.0.2:30000	128.10.24.6:40002
Przychodzący	Docelowy adres IP: docelowy port TCP	128.10.24.6:40001	192.168.0.1:30000
Przychodzący	Docelowy adres IP: docelowy port TCP	128.10.24.6:40002	192.168.0.2:30000

Obydwa komputery lokalne wymienione w tabeli korzystają z portu 30000. Ponieważ system operacyjny wyznacza kolejne numery portów, wystąpienie dwóch takich samych wartości portów jest prawdopodobne. Dzięki mechanizmowi NAPT eliminowana jest niejednoznaczność odwzorowań. W przedstawionym przykładzie jedno z połączeń jest realizowane za wykorzystaniem portu 40001, natomiast drugiemu został przydzielony port 40002.

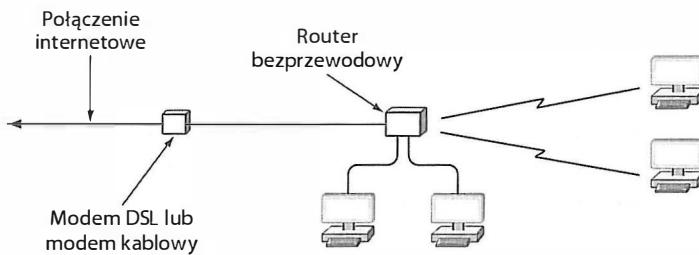
## 23.18. Operacja NAT a dostęp do serwerów

Z zamieszczonych wcześniej informacji wiadomo, że system NAT buduje tablicę translacji w sposób automatyczny, monitorując ruch wychodzący i dodając nowe wpisy za każdym razem, gdy aplikacja wewnętrzna ustanawia nowe połączenie z jednostką zewnętrzną. Niestety, automatyczne rozbudowywanie tablicy translacji nie sprawdza się w przypadku połączeń inicjowanych w internecie. Jeśli w sieci wewnętrznej będą działały dwa komputery z usługami WWW, system NAT nie będzie mógł ustalić, do którego serwera należy przekazać żądania przychodzące z sieci zdalnych. W dostępie do serwerów sieci wewnętrznej pomocna okazuje się funkcja **dwustronnej translacji NAT** (ang. *Twice NAT*). Działanie

mechanizmu wymaga współpracy z serwerem nazw domenowych (DNS). Gdy aplikacja internetowa odwzorowuje adres domenowy komputera lokalnego, serwer DNS zwraca publiczny adres IP przypisany urządzeniu z funkcją NAT i jednocześnie dodaje nowy wpis do tablicy translacji adresów. Nowy wiersz pojawia się w tablicy przed odebraniem pierwszego pakietu z jednostki zewnętrznej. Choć rozwiążanie to nie jest szczególnie eleganckie, sprawdza się w większości zastosowań. Nie można go jedynie wykorzystywać w sytuacjach, w których aplikacja kliencka posługuje się od razu adresem IP, bez wcześniejszego odwołania do serwera nazw domenowych, lub gdy odwzorowanie nazw jest zapewniane przez serwer buforujący.

### 23.19. Oprogramowanie NAT i systemy przeznaczone do sieci domowych

Mechanizm NAT jest szczególnie użyteczny w sieciach domowych i niewielkich sieciach firmowych, które są przyłączone do internetu za pomocą łączyszerokopasmowych. Umożliwia bowiem współdzielenie połączenia przez większą liczbę komputerów bez konieczności kupowania od dostawcy usług internetowych dodatkowych adresów IP. Wdrożenie rozwiązania polega na zastosowaniu oprogramowania, które przekształca jeden komputer w urządzenie NAT świadczące usługę innym komputerom, lub na użyciu niedrogiego urządzenia realizującego funkcję NAT. Moduły tego typu są często nazywane **routerami bezprzewodowymi**<sup>69</sup>. Na przykład firma Linksys sprzedaje urządzenia, które wykonują operacje NAT na pakietach dostarczanych za pomocą czterech przewodowych portów ethernetowych lub radiowego połączenia Wi-Fi. Zasada działania routera bezprzewodowego została przedstawiona na rysunku 23.10.



Rysunek 23.10. Połączenia routera bezprzewodowego

### 23.20. Podsumowanie

Mechanizmy IP wykorzystują protokół odwzorowania adresów (ARP) do powiązania adresu IP najbliższego urządzenia na trasie pakietu z właściwym adresem MAC. Standard ARP definiuje format komunikatu, który jest wymieniany przez komputery w celu odwzo-

<sup>69</sup> Nazwa urządzenia wynika z tego, że zapewnia ono również połączenia bezprzewodowe komputerom lokalnym.

rowania adresu, a także zasady enkapsulacji danych oraz reguły przetwarzania wiadomości ARP. Ponieważ w różnych sieciach obowiązują różne systemy adresowania, specyfikacja ARP uwzględnia jedynie ogólny wzorzec komunikatu i pozwala na dopasowanie go do konkretnego schematu adresowania. Zgodnie ze specyfikacją ARP żądania powinny być wysyłane rozgłoszeniowo, ale odpowiedzi muszą być kierowane do właściwych stacji. Aby uniknąć wysyłania żądań przed każdym pakietem danych, wykorzystuje się pamięć podręczną ARP.

Działaniu protokołu IP towarzyszy praca mechanizmu powiadamiania o błędach. Funkcję tę realizuje internetowy protokół komunikatów sterujących (ICMP). Mechanizm ICMP jest wykorzystywany przez routery w przypadku odebrania datagramu o niepoprawnej zawartości nagłówka lub gdy dostarczenie datagramu do jednostki docelowej okazuje się niemożliwe. Powiadomienia ICMP są zawsze odsyłane do jednostki, która wysłała datagram (nigdy do routerów pośrednich). Poza komunikatami o błędach protokół ICMP pozwala na przesyłanie pewnych komunikatów diagnostycznych, takich jak *żądanie echo* i *odpowiedź echo* używane w poleceniu ping. Każde powiadomienie ICMP ma specjalny format, a właściwy podział wiadomości na pola składowe jest możliwy dzięki identyfikatorowi typu komunikatu. Sam komunikat ICMP jest przesyłany w polu danych datagramu IP.

W początkowej fazie rozwoju sieci IP do pozyskiwania poszczególnych parametrów konfiguracyjnych wykorzystywane były niezależne protokoły. Obecnie stosowany jest protokół dynamicznej konfiguracji stacji (DHCP), który zastąpił protokół BOOTP. Umożliwia on pobranie wszystkich potrzebnych parametrów pracy sieciowej za pomocą jednego żądania. Odpowiedź serwera DHCP zawiera informacje na temat adresu IP, adresu domyślnego routera oraz adresu serwera nazw. W rozwiązaniach bazujących na mechanizmie dynamicznego przydziału adresów serwer DHCP wyznacza czas dzierżawy, w którym jednostka kliencka może korzystać z przyznanego adresu. Po upływie tego czasu musi przedłużyć dzierżawę lub zaprzestać używania adresu.

Mechanizm NAT pozwala większej liczbie komputerów pracujących w sieci lokalnej na komunikowanie się z jednostkami internetowymi za wykorzystaniem pojedynczego adresu IP. Usługa NAT odpowiada za przepisywanie pól w nagłówku IP w czasie przekazywania datagramu między siecią wewnętrzną a internetem. W przypadku aplikacji klienckich translacja NAT może być definiowana automatycznie wraz z pierwszym zarejestrowanym pakietem wychodzącym. Rozwiążanie to jest stosowane w kilku wariantach. Najczęściej wykorzystuje się odmianę NAPT, która uwzględnia w działaniu nagłówki warstwy transportowej i odwzorowuje nie tylko adresy IP, ale również numery portów. Mechanizm NAPT umożliwia dowolnej liczbie aplikacji uruchomionych w dowolnej liczbie komputerów wewnętrznych komunikowanie się z dowolną liczbą jednostek stacji pracujących w internecie.

## Do dalszego studiowania

Więcej informacji na temat mechanizmu NAPT znajduje się w dokumentach RFC 2663 i RFC 2766.

## ZADANIA

- 23.1. Wynikiem wyszukiwania następnego węzła w tablicy routingu jest adres IP kolejnego routera. Jaka operacja musi zostać dodatkowo wykonana, zanim będzie można przesyłać datagram?
- 23.2. Jakiego określenia używa się od opisania powiązania między adresem protokołu a adresem sprzętowym?
- 23.3. Czy protokół ARP może być stosowany w sieciach, które nie obsługują transmisji rozgłoszeniowej? Uzasadnij odpowiedź.
- 23.4. Ilu odpowiedzi spodziewa się komputer wysyłający rozgłoszeniowo żądanie ARP? Wyjaśnij zagadnienie.
- 23.5. Ile oktetów zajmuje komunikat ARP w przypadku wykorzystywania adresów IP i adresów ethernetowych?
- 23.6. W jaki sposób komputer ustala, czy dostarczona ramka jest datagramem IP, czy komunikatem ARP?
- 23.7. Założmy, że komputer otrzymuje dwie odpowiedzi na jedno żądanie ARP. Pierwsza odpowiedź zawiera informację o tym, że adresem MAC jednostki zdalnej jest  $M_1$ . W drugiej odpowiedzi adres ten ma wartość  $M_2$ . W jaki sposób odpowiedzi te zostaną zinterpretowane przez mechanizm ARP?
- 23.8. Protokół ARP zapewnia odwzorowanie adresów tylko w jednej sieci. Czy jest uzasadnione przesyłanie żądania ARP do zdalnego serwera w polu danych datagramu IP? Uzasadnij odpowiedź.
- 23.9. W jakich przypadkach algorytm 23.1 dodaje nowe wpisy do pamięci podręcznej ARP?
- 23.10. Ile rodzajów adresów jest wykorzystywanych w warstwach poniżej ARP?
- 23.11. Który komunikat ICMP zostanie wysłany w przypadku stwierdzenia niepoprawnej wartości jednego z pól nagłówka datagramu?
- 23.12. Który komunikat ICMP zostanie wysłany w przypadku wystąpienia pętli routingu? Opisz przebieg procesu.
- 23.13. Założmy, że użytkownik wprowadził adres rozgłoszenia kierowanego jako parametr poleceńa ping. Jakie są możliwe rezultaty takiego działania? Wyjaśnij zagadnienie.
- 23.14. W niektórych wersjach polecenia traceroute wysyłane są komunikaty ICMP. Działanie innych polega na generowaniu komunikatów UDP. Sprawdź, który wariant został wykorzystany w komputerze, z którego korzystasz.
- 23.15. Które pola ramki ethernetowej komputer musi przeanalizować, aby ustalić, czy ramka zawiera powiadomienie ICMP?
- 23.16. Sporządź listę najważniejszych parametrów sieciowych, które trzeba dostarczyć komputerowi w czasie jego uruchamiania.
- 23.17. Jaka jest główna różnica między protokołami BOOTP i DHCP?
- 23.18. Niektóre aplikacje sieciowe odraczają procedurę konfiguracji do czasu skorzystania z usługi. Na przykład system operacyjny może opóźnić operację wyszukiwania drukarek sieciowych aż do czasu, gdy użytkownik zleci wydruk dokumentu. Jaka jest najważniejsza zaleta takiego sposobu postępowania? A jaka jest jego wada?
- 23.19. Protokół DHCP umożliwia odszukanie serwera w sieci zdalnej. W jaki sposób komputer dostarcza komunikaty DHCP do serwera pracującego w innej sieci?

- 23.20. Opracuj algorytm alternatywny w odniesieniu do DHCP, który będzie zawierał mechanizm kojarzenia adresów. Załóż, że kopia tego algorytmu zostanie uruchomiona w każdym komputerze. W wyniku jego działania każdy komputer powinien otrzymać niepowtarzalny adres IP.
- 23.21. Jakie jest główne zastosowanie mechanizmu NAT?
- 23.22. Wiele urządzeń NAT wykorzystuje przedział adresowy 10.0.0.0/8, ponieważ jest on najbardziej ogólny. Wyjaśnij dlaczego.
- 23.23. Zgodnie z informacjami zawartymi w tabeli 23.3 dostawca usług internetowych przydzielił urządzeniu sieci lokalnej jeden adres IP. Jaki to adres?
- 23.24. Rozbuduj zestawienie z tabeli 23.3 o odwzorowania, które zostaną zdefiniowane, gdy trzeci komputer spróbuje skomunikować się z tym samym serwerem internetowym.
- 23.25. Opracuj tablicę NAPT dla przypadku, w którym trzy komputery sieci lokalnej utrzymują połączenia z trzema niezależnymi internetowymi serwerami WWW.
- 23.26. Jaka kluczowa informacja wykorzystywana przez mechanizm NAPT nie występuje w większości fragmentów IP?
- 23.27. W celu zoptymalizowania procesu odtwarzania datagramów niektóre wersje systemu Linux wysyłają ostatnie fragmenty jako pierwsze, a pozostałe w poprawnej kolejności. Wyjaśnij dlaczego.
- 23.28. Jakie adresy IP mogą zostać przypisane komputerom w sieci lokalnej, w której funkcjonuje router bezprzewodowy?

## *Zawartość rozdziału*

- 24.1. Wprowadzenie 425
- 24.2. Sukces protokołu IP 425
- 24.3. Potrzeba zmian 426
- 24.4. Model klepsydry i trudności we wprowadzaniu zmian 427
- 24.5. Nazwa i numer wersji 428
- 24.6. Funkcje IPv6 428
- 24.7. Format datagramu IPv6 429
- 24.8. Format podstawowego nagłówka protokołu IPv6 429
- 24.9. Jawny i niejawny rozmiar nagłówka 431
- 24.10. Fragmentacja, odtwarzanie datagramów i MTU trasy 431
- 24.11. Przeznaczenie wielokrotnych nagłówków 433
- 24.12. Adresacja IPv6 434
- 24.13. Zapis adresów IPv6 w formacie szesnastkowym z dwukropkami 435
- 24.14. Podsumowanie 436

# 24

## Przyszłość protokołu IP (IPv6)

### 24.1. Wprowadzenie

W poprzednich rozdziałach została opisana bieżąca wersja protokołu internetowego IPv4. Przedstawiono w nim datagram IP składający się z nagłówka i pola danych. Jak wiadomo, w nagłówku znajdują się informacje potrzebne do prawidłowego dostarczenia pakietu do odbiorcy, w tym docelowy adres IP. Każde pole nagłówka ma stały rozmiar, co zwiększa efektywność przetwarzania datagramów. Rozdział 22. zawierał omówienie sposobu enkapsulowania datagramu IP w ramce na czas przesyłania go przez sieć fizyczną.

Tematem tego rozdziału jest przyszłość protokołu internetowego. Omówienie rozpoczyna się od analizy słabych i mocnych stron bieżącej wersji mechanizmu IP, po czym następuje prezentacja nowej wersji protokołu IP opracowanej przez organizację IETF. Obejmuje ona nowe funkcje protokołu oraz sposoby eliminowania ograniczeń bieżącego rozwiązania.

### 24.2. Sukces protokołu IP

Bieżąca wersja protokołu IP cieszy się ogromną popularnością. Umożliwiła zbudowanie internetu z heterogenicznych sieci, przetrwała dramatyczne zmiany w technologii sprzętowej oraz istotne zwiększenie skali zastosowań. Protokoły internetowe uwzględniają wiele abstrakcyjnych mechanizmów, które pozwalają aplikacjom na komunikowanie się bez znajomości architektury internetu lub nawet urządzeń transmisyjnych. Współdziałanie z różnymi urządzeniami jest możliwe dzięki uniezależnieniu protokołu od takich elementów jak schemat adresowania, format datagramów, mechanizm enkapsulacji oraz proces fragmentacji.

Dowodem na wszechstronność i skalowalność rozwiązań IP jest różnorodność aplikacji korzystających z protokołu IP oraz rozmiar globalnej sieci internetowej. Nie mniej

istotna jest niezależność mechanizmu od zmian sprzętowych. Mimo że został on zaprojektowany przed upowszechnieniem się sieci lokalnych, doskonale sprawdza się we współpracy z technologiami późniejszymi o kilka generacji. Funkcjonuje poprawnie mimo zwiększenia szybkości transmisji i rozmiarów ramek.

Podsumowując:

*Sukces bieżącej wersji protokołu IP jest bezdiskusyjny. Rozwiążanie to doskonale sprawdza się w działaniu mimo zmian technologii sprzętowych, budowania sieci heterogenicznych oraz zwiększenia skali zastosowań.*

### 24.3. Potrzeba zmian

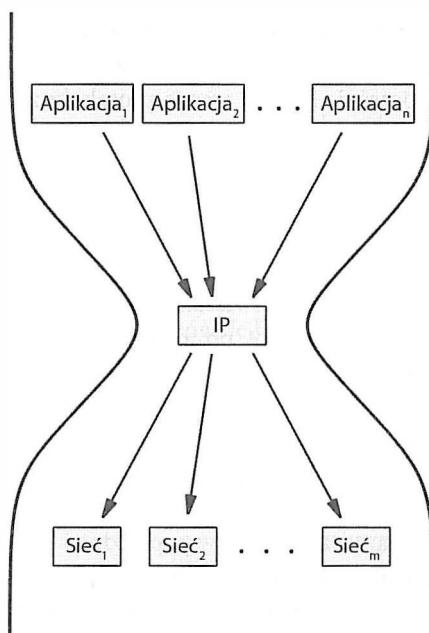
Skoro protokół IP działa tak dobrze, to po co go zmieniać? W czasie, w którym powstawał, istniało niewiele sieci komputerowych. Projektanci mechanizmu zdecydowali o przeznaczeniu 32 bitów na adres stacji, co pozwala na utworzenie ponad miliona sieci. Globalny internet rozrasta się jednak w tempie wykładniczym, a jego rozmiar podwaja się w okresie krótszym niż rok. Wkrótce wyczerpią się zasoby prefiksów sieciowych i dalszy rozwój sieci będzie niemożliwy. Dlatego jedną z podstawowych przesłanek ku temu, by rozpocząć prace nad nową wersją protokołu IP, było zwiększenie przestrzeni adresowej — dłuższe ciągi adresów są niezbędne do zagwarantowania ciągłego rozwoju internetu.

Drugi powód zmian to konieczność wprowadzenia pewnych dodatkowych funkcji, które są niezbędne w działaniu niektórych aplikacji. Jako przykład warto tutaj rozważyć aplikację przesyłającą dźwięk i obraz w czasie rzeczywistym. Jej działanie polega na dostarczaniu danych w regularnych odstępach czasu i jest zależne od niskiej wartości fluktuacji opóźnienia. Niestety, zmiany tras powodują zazwyczaj zmiany opóźnienia w dostarczaniu danych między punktami końcowymi, a to z kolei wpływa na fluktuację opóźnień. Mimo że nagłówki datagramów bieżącej wersji protokołu IP zawierają pola rodzaju usługi, w specyfikacji protokołu nie uwzględniono usług czasu rzeczywistego. Dlatego wiele osób postuluje wprowadzenie w nowej wersji standardu IP mechanizmów, które zapewnią zachowanie stałych tras podczas przekazywania pakietów aplikacji czasu rzeczywistego.

Inna grupa użytkowników protokołu przekonuje, że nowa wersja rozwiązań IP powinna uwzględniać znacznie bardziej złożony system adresowania i bardziej wyrafinowane techniki routingu. Proponuje się wprowadzenie zmian w adresacji i routingu, które pozwolą na powielanie usług. Na przykład firma Google utrzymuje liczne centra danych na całym świecie. Byłyby więc bardzo ważne, aby po wpisaniu w przeglądarce adresu [google.com](http://google.com) pakietы zostały skierowane do najbliższego ośrodka firmy Google. Wiele aplikacji zachęca użytkowników do **pracy grupowej**. Takie współdziałanie może być efektywne tylko, jeśli rozwiązania internetowe umożliwiają tworzenie grup i modyfikowanie ich składu oraz zapewnianie dostarczanie każdego pakietu do wszystkich członków grupy.

## 24.4. Model klepsydry i trudności we wprowadzaniu zmian

Choć w czasie rozpoczętia prac nad nową wersją protokołu IP (w 1993 roku) nikt nie miał wątpliwości, że zasoby adresowe się wyczerpują, nie było powodu do radykalnych działań i protokół IP nie został zmieniony. Przyczynami były, oczywiście, upowszechnienie się bieżącej wersji mechanizmu oraz wysoki koszt wprowadzenia zmian. Protokół IP jest najważniejszym elementem komunikacji internetowej. Wszystkie aplikacje korzystają z mechanizmów IP. Sam protokół może natomiast współdziałać ze wszystkimi bieżącymi technologiami transmisyjnymi. Eksperci zajmujący się problemami internetu twierdzą, że komunikacja sieciowa przypomina **model klepsydry**, a protokół IP znajduje się w najważszej punkcie tej klepsydry. Graficzna reprezentacja tej idei została przedstawiona na rysunku 24.1.



Rysunek 24.1. Model klepsydry opisujący komunikację internetową z wykorzystaniem protokołu IP

Zależność innych rozwiązań od protokołu IP oraz konsekwencje zmiany mechanizmu IP prowadzą do następującego wniosku:

*Z uwagi na kluczowe znaczenie protokołu IP we wszelkich formach komunikacji internetowej, wprowadzenie jakichkolwiek zmian w protokole IP oznacza zmianę całego internetu.*

## 24.5. Nazwa i numer wersji

Wraz z rozpoczęciem prac nad nową wersją protokołu IP uczestnicy projektu zaczęli się zastanawiać nad nazwą rozwiązań. Pod wpływem popularnego programu telewizyjnego postanowili nadać specyfikacji nazwę *IP – The Next Generation*, a w pierwszych wzmiankach o nowym protokole używano określenia **IPng**. Niestety, nazwa IPng była wykorzystywana również w opisie wielu konkurencyjnych rozwiązań, przez co stała się niejednoznaczna.

Po zakończeniu prac projektanci musieli w jakiś sposób wyróżnić nowy protokół spośród innych dostępnych specyfikacji. Postanowiono użyć oficjalnego numeru wersji, który jest zapisywany w nagłówku ostatecznej, zatwierzonej wersji protokołu. Jednak wybrany numer okazał się pewnym zaskoczeniem. Ponieważ dotychczasowy protokół IP ma wersję 4, spodziewano się, że wybrana zostanie kolejna liczba, czyli 5. Wersja 5 była już jednak zarezerwowana dla eksperymentalnej odmiany protokołu, określonej też skrótem ST. W rezultacie nowa wersja protokołu IP została oznaczona numerem 6, a sam protokół stał się znany jako **IPv6**. Jednocześnie bieżącą wersję protokołu zaczęto nazywać **IPv4**, aby odróżnić ją od rozwiązania IPv6.

## 24.6. Funkcje IPv6

W specyfikacji IPv6 zachowano wiele rozwiązań projektowych wprowadzonych z powodzeniem w wersji IPv4. Podobnie jak IPv4, protokół IPv6 jest protokołem bezpołączeniowym — każdy datagram zawiera adres docelowy i jest przesyłany niezależnie od pozostałych. Nagłówek datagramu uwzględnia również pole maksymalnej liczby węzłów, przez które datagram może zostać przekazany przed usunięciem. Co ważniejsze, w standardzie IPv6 zachowano większość ogólnych funkcji definiowanych w opcjach IPv4.

Mimo zgodności na poziomie idei funkcjonowania, szczegółowy sposób działania IPv6 całkowicie odbiega od rozwiązań zawartych w poprzedniej wersji. W mechanizmie IPv6 zastosowano znacznie dłuższe adresy oraz zupełnie nowy format nagłówka. Ponadto sam nagłówek jest dzielony na szereg nagłówków składowych o ustalonym rozmiarze. Zatem w przeciwieństwie do protokołu IPv4, w którym najważniejsze informacje zajmują wstępnie ustalone pola, a mniej istotne parametry są zapisywane w polu opcji (o zmiennej długości), nagłówek IPv6 z założenia ma zmienną długość.

Nowe cechy protokołu IPv6 można podzielić na pięć ogólnych kategorii:

- **Rozmiar adresu.** Każdy adres IPv6, zamiast na 32 bitach, jest zapisywany na 128 bitach. Dzięki temu wyznacza przestrzeń adresową, która nie ograniczy rozwoju internetu przez wiele kolejnych dekad.
- **Format nagłówka.** Nagłówek IPv6 znacznie różni się od nagłówka protokołu IPv4. Zmiany odnoszą się do niemal każdego pola. Niektóre z pól zostały zastąpione innymi.
- **Nagłówki rozszerzające.** W przeciwieństwie do protokołu IPv4, bazującego na wstępnie określonym formacie nagłówka, oprogramowanie protokołu IPv6 zapro-

suje informacje dodatkowe w oddzielnich nagłówkach. Datagram składa się więc z podstawowego nagłówka IPv6, po którym następują nagłówki rozszerzające (o ile zostały zdefiniowane) oraz dane.

- **Obsługa ruchu z aplikacji czasu rzeczywistego.** W specyfikacji IPv6 uwzględniono mechanizmy, które umożliwiają nadawcy i odbiorcy wyznaczenie trasy o wysokiej jakości transmisji danych w ramach istniejącej sieci i skojarzenie z tą trasą określonego rodzaju pakietów. Choć rozwiązanie to zostało opracowane z myślą o przesyłaniu danych z aplikacji audiowizualnych (wymagających dużej przepustowości), można je zastosować również do powiązania pewnych datagramów z trasami o niższym koszcie przesyłu danych.
- **Rozszerzalność protokołu.** W przeciwieństwie do standardu IPv4, specyfikacja IPv6 nie definiuje wszystkich dozwolonych funkcji protokołu. Autorzy rozwiązania opracowali schemat, zgodnie z którym nadawca ma możliwość uzupełniania datagramu o własne informacje. Protokół IPv6 jest więc znacznie bardziej elastyczny niż IPv4, ponieważ umożliwia dodawanie nowych funkcji w razie potrzeby.

Sposób implementacji nowych rozwiązań został omówiony w kolejnym punkcie, w którym przedstawiono strukturę datagramu IPv6 oraz schemat adresowania.

## 24.7. Format datagramu IPv6

Datagram IPv6 składa się z wielu nagłówków. Jednak zgodnie z rysunkiem 24.2 na początku każdego datagramu znajduje się **nagłówek podstawowy** (ang. *base header*), po którym następują **nagłówki rozszerzające** (jeśli zostały zdefiniowane) oraz pole danych.



Rysunek 24.2. Ogólny format datagramu IPv6

Na rysunku przedstawiono ogólną strukturę datagramu. Poszczególne pola nie zostały narysowane we właściwej skali. Dotyczy to przede wszystkim nagłówków rozszerzających, które mogą być większe lub mniejsze niż nagłówek podstawowy. Ponadto pole danych większości datagramów jest znacznie większe od rozmiaru nagłówków.

## 24.8. Format podstawowego nagłówka protokołu IPv6

Mimo że nagłówek IPv6 jest dwa razy większy od nagłówka IPv4, zawiera mniej informacji. Jego format jest widoczny na rysunku 24.3.

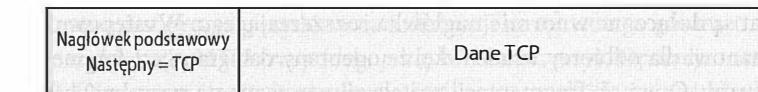
Wersja	Klasa ruchu	Etykieta strumienia
Długość pola danych	Następny nagłówek	Limit przeskóków
—	Adres źródłowy	—
—	—	—
—	—	—
—	Adres docelowy	—
—	—	—

Rysunek 24.3. Format podstawowego nagłówka datagramu IPv6

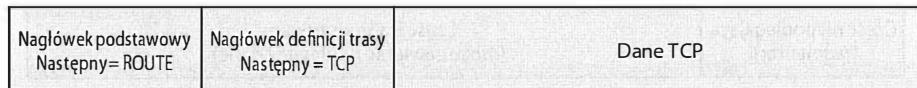
Jak nietrudno zauważać na rysunku, większa część nagłówka jest zajęta przez pola *adres źródłowy* i *adres docelowy*. Każde z nich zajmuje szesnaście oktetów, czyli cztery razy więcej bitów niż adres IPv4. Podobnie jak w standardzie IPv4 adres źródłowy identyfikuje nadawcę datagramu, a adres docelowy ostatecznego odbiorcę datagramu.

Poza adresami w nagłówku podstawowym znajduje się sześć pól. Wartość *wersja* informuje o tym, że jest to wersja 6 protokołu. Pole *klasa ruchu* definiuje rodzaj strumienia danych zgodnie z klasyfikacją **różnicowanych usług** (ang. *differentiated services*). Określa jednocześnie ogólne parametry transmisji, które muszą być zagwarantowane danemu pakietowi. Na przykład aby przesyłać informacje o interakcjach użytkownika (o naciśkanych klawiszach i przesuwaniu myszki), należałoby użyć klasy, która gwarantuje najmniejsze opóźnienie w dostarczaniu datagramów. Z kolei przesyłając dane audiowizualne, nadawca powinien wybrać klasę o najmniejszej fluktuacji opóźnienia. Pole *długość pola danych* odpowiada analogicznej wartości z nagłówkiem IPv4. Jednak w przeciwieństwie do rozwiązania stosowanego w protokole IPv4, wartość *długości pola danych* wyznacza rozmiar samego bloku danych (bez uwzględniania rozmiaru nagłówka). Wartość *limit przeskóków* jest odpowiednikiem wartości TTL ze standardu IPv4. Wyzerowanie licznika przeskóków powoduje usunięcie pakietu z sieci. Pole *etykieta strumienia* było początkowo przeznaczone do kojarzenia datagramu z określona trasą w sieci. Jednak etykietowanie ruchu przenoszonego między dwoma jednostkami końcowymi nie zyskało akceptacji, przez co przeznaczenie tego pola stało się mniej istotne.

Pole *następny nagłówek* służy do określenia rodzaju informacji zawartych w obszarze, który następuje po bieżącym nagłówku. Na przykład jeśli datagram zawiera nagłówki rozszerzające, wartość *następny nagłówek* opisuje rodzaj kolejnego nagłówka. Jeśli jednak w pakiecie nie uwzględniono nagłówków rozszerzających, pole to przechowuje informację o rodzaju danych zapisanych w polu danych. Przykład zastosowania pola *następny nagłówek* został przedstawiony na rysunku 24.4.



(a)



(b)

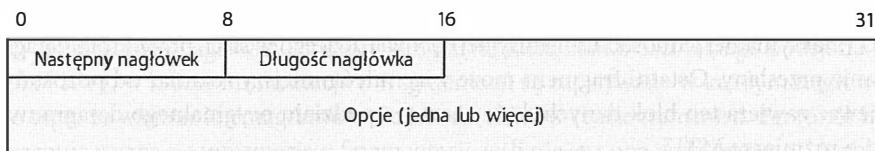
**Rysunek 24.4.** Pole typu kolejnego nagłówka w datagramie IPv6

składającym się z nagłówka podstawowego i bloku danych TCP (a) oraz w datagramie obejmującym nagłówek podstawowy, nagłówek definicji trasy oraz blok danych TCP (b)

## 24.9. Jawny i niejawny rozmiar nagłówka

W standardzie zdefiniowano identyfikatory każdego obsługiwanej rodzaju nagłówka. Nie ma więc niejednoznaczności w interpretacji wartości *następny nagłówek*. Odbiorca przetwarza nagłówki w odpowiedniej kolejności, a wartość wspomnianego pola zawsze określa rodzaj informacji, które następują po bieżącym bloku.

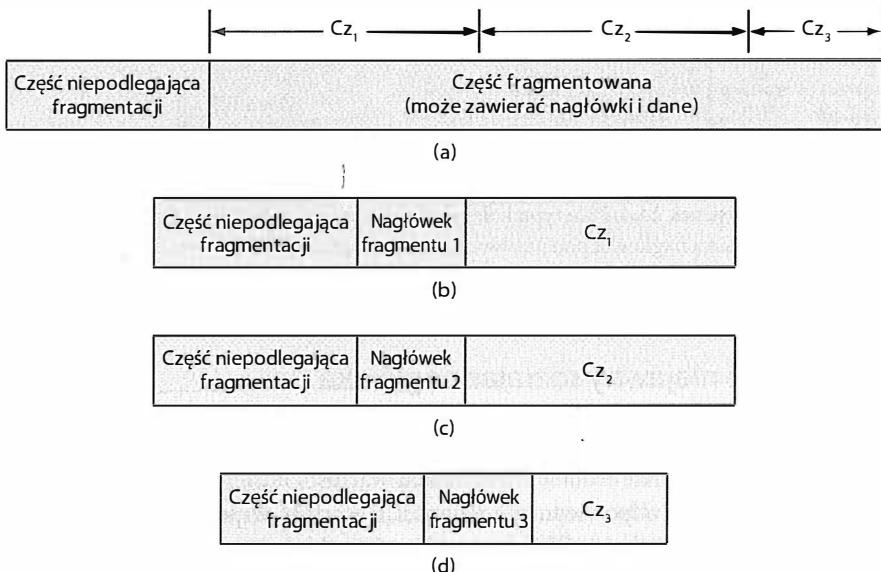
Niektóre nagłówki mają stały rozmiar. Na przykład nagłówek podstawowy zawsze składa się z 40 oktetów. Aby odczytać kolejny element datagramu, oprogramowanie IPv6 po prostu dodaje wartość 40 do adresu nagłówka podstawowego. Niektóre rozszerzenia nie mają jednak wstępnie zdefiniowanego rozmiaru. W takich przypadkach informacja o granicy nagłówka jest zapisywana w nim samym. Na rysunku 24.5 został pokazany **nagłówek opcji IPv6**, który pełni tę samą funkcję, co pole opcji w protokole IPv4.

**Rysunek 24.5.** Nagłówek opcji stanowiący rozszerzenie nagłówka IPv6 z jawnie określonym rozmiarem

## 24.10. Fragmentacja, odtwarzanie datagramów i MTU trasy

Choć mechanizm fragmentacji w protokole IPv6 działa podobnie do rozwiązania znanego ze standardu IPv4, między obydwooma algorytmami występują pewne różnice. Podobnie jak w systemie IPv4 identyfikator pierwotnego datagramu jest kopowany do każdego fragmentu. Zmieniany jest również parametr długości pola danych, tak aby odpowiadał rozmiarowi fragmentu. Jednak w przeciwieństwie do protokołu IPv4 mechanizm IPv6 nie uwzględnia w podstawowym nagłówku pól sterujących procesem fragmentacji. Informacje

na ten temat są dołączane w formie nagłówka rozszerzającego. Występowanie takiego nagłówka stanowi dla odbiorcy wskazówkę, że odebrany datagram jest fragmentem większego datagramu. Operacja fragmentacji została zilustrowana na rysunku 24.6.



Rysunek 24.6. Datagram IPv6 (a) podzielony na fragmenty (b) – (d)

Obszary oznaczone jako *części niepodlegające fragmentacji* odpowiadają nagłówkowi bazowemu, który określa trasę dostarczania pakietu. Aby zagwarantować poprawne dostarczenie wszystkich fragmentów datagramu, obszary te muszą zostać skopiowane bez jakichkolwiek zmian.

Podobnie jak w przypadku protokołu IPv4, rozmiar fragmentu zależy od parametru MTU (maksymalnej jednostki transmisyjnej) obowiązującego w sieci, przez którą datagram zostanie przesłany. Ostatni fragment może więc mieć mniejszy rozmiar od pozostałych, ponieważ zawiera ten blok danych, który został z podziału oryginalnego datagramu na bloki o rozmiarze MTU.

Sama operacja fragmentacji różni się istotnie od mechanizmu stosowanego w protokole IPv4. W rozwiązaniach IPv4 za podział datagramu odpowiada router, który odbierze pakiet o zbyt dużym rozmiarze, aby można go było przesyłać w kolejnej sieci. W systemie IPv6 fragmentacja należy do zadań jednostki nadawczej. Oznacza to, że komputer nadawcy musi dobrąć rozmiar datagramu tak, aby nie wymagał on dalszego podziału. Jeśli którykolwiek z routerów pośredniczących w transporcie pakietu stwierdzi, że rozmiar datagramu przekracza wartość parametru MTU w danej sieci, usuwa pakiet z sieci i wysyła do stacji nadawczej komunikat o błędzie.

W jaki sposób stacja końcowa może wyznaczyć rozmiar datagramu tak, aby nie wymagał on podziału? Musi pozyskać informacje o wartości MTU każdego odcinka trasy do jednostki docelowej, a następnie wybrać z nich wartość najmniejszą. Najmniejsza wartość MTU na trasie między stacją źródłową a docelową jest nazywana parametrem **MTU trasy** (ang.

*path MTU*), natomiast proces pozyskiwania informacji na temat tego parametru jest określany jako **poszukiwanie MTU trasy** (ang. *path MTU discovery*). Operacja ustalania wartości MTU jest procesem iteracyjnym. Stacja końcowa wysyła do jednostki zdalnej serię datagramów o różnych rozmiarach i oczekuje na ewentualny komunikat o błędzie<sup>70</sup>. Ten, który zostanie przekazany bez potrzeby fragmentacji, wyznacza MTU trasy.

Podsumowując:

*W protokole IPv6 fragmentacja jest wykonywana po stronie nadawczej, a nie przez routery. Jeśli jej wykonanie jest konieczne, jednostka źródłowa otrzymuje komunikat ICMP z informacją o błędzie. Zmniejsza wówczas rozmiar fragmentu aż do wartości, która gwarantuje poprawne dostarczenie danych.*

## 24.11. Przeznaczenie wielokrotnych nagłówków

Dlaczego w protokole IPv6 stosowane są oddzielne nagłówki rozszerzające? Są ku temu dwa powody:

- ekonomia,
- rozszerzalność.

Powód ekonomiczny jest najłatwiejszy do wytłumaczenia. Zamknięcie informacji o cechach datagramu w niezależnych nagłówkach pozwala na zaoszczędzenie pasma transmisyjnego. Choć w standardzie IPv6 zdefiniowano wiele funkcji, autorzy rozwiązania zakładają, że tylko niewielki ich zbiór będzie uwzględniany w poszczególnych transmitowanych pakietach. Wydzielenie osobnych nagłówków pozwala na opracowanie dużego zbioru funkcji, ale bez konieczności włączania choćby pojedynczych pól danego rozwiązania do zasadniczego nagłówka. Doskonałym przykładem zasadności takiego podejścia jest protokół IPv4, w którym informacje o fragmentacji są zapisywane w każdym nagłówku, mimo tego, że większość datagramów nie jest dzielona w czasie transmisji. Protokół IPv6 nie marnuje pasma na informacje o fragmentacji, jeśli nie jest ona wykonywana. Uwzględniając fakt, że większość datagramów wymaga zdefiniowania niewielkiej liczby parametrów, pomijanie zbędnych pól znacznie zmniejsza zajętość łączy. Poza tym mniejsze datagramy są szybciej przesyłane.

Aby zrozumieć znaczenie rozszerzalności protokołu, wystarczy przeanalizować proces dodawania do niego nowej funkcji. W przypadku takiego protokołu jak IPv4 (o stałym rozmiarze nagłówka) oznacza on całkowitą zmianę formatu nagłówka — struktura nagłówka musi zostać zmieniona w taki sposób, aby uwzględniała informację o nowym elemencie. W rozwiązaniu IPv6 wszystkie wcześniejsze nagłówki pozostają niezmienione. Operacja sprowadza się natomiast do zdefiniowania nowej wartości identyfikatora *następny nagłówek* oraz ustalenia formatu nowego nagłówka.

<sup>70</sup> Protokołowi IPv6 towarzyszy nowa wersja protokołu ICMP.

Inną zaletą definiowania nowej funkcji w osobnym nagłówku jest to, że można ją testować przed wymuszeniem zmian we wszystkich komputerach w internecie. Założymy na przykład, że dwóch użytkowników komputerów chce przetestować nową technikę szyfrowania datagramów. Muszą, oczywiście, ustalić między sobą, jakiego będzie format eksperymentalnego nagłówka. Nadawca może wówczas dodawać nowy nagłówek do datagramu, a odbiorca będzie poprawnie interpretował nadchodzące dane. Jeśli testowany nagłówek będzie zapisywany za nagłówkiem odpowiedzialnym za routing, routery internetowe będą mogły bez przeszkoły przesyłać pakiety, mimo że nie znają formatu dodatkowego nagłówka<sup>71</sup>. Jeżeli testy potwierdzą przydatność nowego rozwiązania, wystarczy uwzględnić je w standardzie.

## 24.12. Adresacja IPv6

Podobnie jak w protokole IPv4, w specyfikacji IPv6 wyróżniono niepowtarzalny identyfikator każdego połączenia między komputerem a siecią fizyczną. Zatem jeśli komputer (lub router) jest przyłączony do trzech sieci, musi mieć przypisane trzy adresy IPv6. W standardzie IPv6 jest również zdefiniowany podział na prefiks opisujący sieć oraz sufiks wskazujący komputer w danej sieci.

Mimo zastosowania podobnego rozwiązania w wyznaczaniu adresów jednostek adresowej IPv6 znacznie różni się od technik doboru adresów właściwych mechanizmowi IPv4. Różnica ujawnia się przede wszystkim w przeznaczeniu poszczególnych składowych adresu. W nowym systemie również istnieje dowolność w ustalaniu granicy podziału na prefiks i sufiks (tak jak w adresowaniu CIDR). Jednak twórcy standardu wprowadzili wielopoziomową hierarchię. Choć podział adresu nie jest stały, zazwyczaj przyjmuje się, że najwyższy poziom w hierarchii należy do dostawcy usług internetowych, kolejny odpowiada organizacji (na przykład przedsiębiorstwu), następny ośrodkowi itd. W specyfikacji IPv6 został zdefiniowany także zbiór adresów specjalnych, których przeznaczenie jest nieco inne niż w protokole IPv4. W standardzie IPv6 nie ma na przykład specjalnego adresu rozgłoszenia kierowanego. Każdy z adresów należy natomiast do jednej z trzech grup wymienionych w tabeli 24.1.

Jak wynika z zestawienia, pozostały adresy emisji pojedynczej oraz multiemisji. Rozgłoszenia kierowane zostały natomiast usunięte, ponieważ stanowiły zagrożenie dla systemu bezpieczeństwa sieci. Aby umożliwić lokalne rozgłaszczenie danych, uwzględniono specjalną grupę multiemisji, która obejmuje wszystkie komputery i routery w sieci lokalnej.

Adresowanie z kategorii emisji dowolnej było pierwotnie nazywane adresowaniem **klastrowym**. Jego celem jest umożliwienie zwielokrotniania usług. Adresy z tej kategorii można na przykład przypisać grupie komputerów świadczących jedną usługę. Wysłanie żądania na adres emisji dowolnej spowoduje dostarczenie go do jednego z komputerów grupy (tj. jednego węzła w klastrze). Jeśli analogiczne żądanie zostanie wygenerowane w sieci zdalnej, datagram może zostać dostarczony do innego członka grupy. Rozwiązanie to pozwala więc na jednoczesne przetwarzanie wielu żądań.

---

<sup>71</sup> Jeśli eksperymentalny nagłówek zostanie zapisany przed nagłówkiem routingu, router odrzuci datagram.

Tabela 24.1. Trzy rodzaje adresów IPv6

Rodzaj	Przeznaczenie
Emisja pojedyncza (ang. <i>unicast</i> )	Adres ten odpowiada pojedynczemu komputerowi. Datagram wysłany na ten adres jest dostarczany najkrótszą trasą do wskazanej jednostki.
Multiemisja (ang. <i>multicast</i> )	Adres ten odpowiada grupie komputerów, a skład grupy może się zmienić w dowolnym czasie. Protokół IPv6 gwarantuje dostarczenie kopii każdego datagramu do wszystkich członków grupy.
Emisja dowolna (ang. <i>anycast</i> )	Adres ten odpowiada grupie komputerów o wspólnym prefiksie. Datagram wysłany na ten adres jest dostarczany tylko do jednego komputera z grupy (zlokalizowanego najbliżej nadawcy).

## 24.13. Zapis adresów IPv6 w formacie szesnastkowym z dwukropkami

Zapisywanie 128-bitowych wartości adresu IPv6 zgodnie z wcześniejszym standardem jest dość trudne. Oto przykład adresu zapisanego w notacji dziesiętnej z kropkami (charakterystycznej dla protokołu IPv4):

105.220.136.100.255.255.255.0.0.18.128.140.10.255.255

Aby skrócić zapis, twórcy protokołu IPv6 postanowili zastosować bardziej zwartą notację, nazywaną **notacją szesnastkową z dwukropkami** (ang. *colon hexadecimal notation*). Zakłada ona grupowanie wartości 16-bitowych i rozdzielanie poszczególnych grup znakami dwukropka. Ten sam adres zapisany w formacie szesnastkowym z kropkami ma wartość:

69DC:8864:FFFF:0:1280:8C0A:FFFF

Jak nietrudno zauważyć, notacja szesnastkowa wymaga użycia mniejszej liczby znaków. Efekt skrócenia uzyskuje się dodatkowo przez zastosowanie **kompresji zer**, czyli zastępowanie sekwencji zer dwoma znakami dwukropka. Na przykład adres:

FF0C:0:0:0:0:0:0:B1

można zapisać jako:

FF0C::B1

Kompresja zer jest dość istotnym usprawnieniem, biorąc po uwagę dużą przestrzeń adresową oraz zaproponowany schemat alokacji adresów. Autorzy specyfikacji spodziewają się bowiem, że większość adresów IPv6 będzie zawierała długie ciągi zer. W standardzie IPv6 uwzględniono również mechanizm odwzorowania dotychczasowych adresów IPv4 na adresy z przestrzeni IPv6. Każdy adres rozpoczynający się od 96 bitów zerowych zawiera na pozostałych 32 bitach adres IPv4.

## 24.14. Podsumowanie

Choć bieżąca wersja protokołu IP doskonale sprawdzała się przez wiele lat, wykładowczy rozwój internetu sprawił, że dotychczasowa 32-bitowa przestrzeń adresowa została wyczerpana. Organizacja IETF opracowała więc nową wersję protokołu IP, która do reprezentacji adresów wykorzystuje 128 bitów. Aby odróżnić nową wersję protokołu od wcześniejszej, w nazwie standardu uwzględnia się także numer tej wersji. Zatem bieżący wariant protokołu to IPv4, a przyszły to IPv6.

Mimo zmian implementacyjnych, w standardzie IPv6 zachowano wiele rozwiązań z IPv4. Podobnie jak w IPv4 mechanizm IPv6 jest systemem bezpołączeniowym, w którym dwa komputery wymieniają krótkie komunikaty nazywane datagramami. Jednak w przeciwieństwie do protokołu IPv4 (w którym nagłówek datagramu zawiera pola odzwierciedlające wszystkie funkcje mechanizmu) specyfikacja IPv6 definiuje oddzielne nagłówki odpowiadające poszczególnym zastosowaniom protokołu. Datagram IPv6 składa się z nagłówka podstawowego, po którym następują: zero lub więcej nagłówków rozszerzających i dane.

Podobnie jak w standardzie IPv4 specyfikacja IPv6 wyniesza przypisywanie adresów do każdego połączenia sieciowego. Zatem jednostka przyłączona do większej liczby sieci (na przykład router) musi dysponować wieloma adresami. W porównaniu z rozwiązaniami IPv4 zmienione zostało znaczenie adresów specjalnych. Zamiast znanego z protokołu IPv4 adresu rozgłoszeniowego wyróżniono adresy multiemisji i emisji dowolnej, które odpowiadają grupie jednostek. Adres multiemisji odnosi się do komputerów pracujących w różnych sieciach, które są traktowane jak pojedynczy odbiorca (każdy komputer otrzymuje kopię datagramu wysłanego na adres grupy). Natomiast adres emisji dowolnej usprawnia replikację usług (datagram wysłany na adres misji dowolnej jest dostarczany do jednego komputera z grupy; zazwyczaj najbliższego względem nadawcy).

Aby uczynić adresy IPv6 łatwiejszymi do zapamiętania, twórcy protokołu opracowali notację szesnastkową z dwukropkami. Zgodnie z jej wytycznymi bity adresu są grupowane po 16 i zapisywane w formacie szesnastkowym. Do rozdziału poszczególnych grup służy znak dwukropka. Dzięki zasadzie kompresji zer można z adresu usunąć długie ciągi o zerowej wartości. Wynikowy adres ma format bardziej zwarty niż adres protokołu IPv4.

## ZADANIA

- 24.1. Jaki jest podstawowy powód zmiany protokołu IPv4 na IPv6?
- 24.2. Co w komunikacji internetowej opisuje model klepsydry?
- 24.3. Wymień najważniejsze cechy protokołu IPv6 i krótko je scharakteryzuj.
- 24.4. Jaki rozmiar ma najmniejszy nagłówek datagramu IPv6?
- 24.5. Jakie jest znaczenie pola *następny nagłówek* w nagłówku datagramu IPv6?
- 24.6. Która część datagramu IPv6 podlega fragmentacji?
- 24.7. Dlaczego w protokole IPv6 stosuje się oddzielne nagłówki rozszerzające, a nie pojedyncze nagłówki o stałym rozmiarze?

- 24.8.** Wymień trzy typy adresów IPv6 i opisz ich przeznaczenie.
- 24.9.** Napisz program komputerowy, który pobierze 128-bitową liczbę i wyświetli ją w notacji szesnastkowej z dwukropkami.
- 24.10.** Dodaj do programu z poprzedniego zadania funkcję kompresji zer.

**24.14. Podsumowanie** Wprowadzenie do tego i dwóch wcześniejszych rozdziałów w tym drzewie i podsumowaniu zostało stwierdzone, że jednostki końcowe nie mają możliwości tworzenia protokołu UDP, aby skutecznie komunikować się z jednostkami średnimi. Organizacja IEEE stworzyła więc nowy protokół UDP, który ma możliwość wykorzystania 128 bitów aby oznaczyć morelowe numerowanie jednostek końcowych IPx4, a mniej więcej IPx6.

Mimo że sam implementacyjny, w standardzie IEEE nie było wiele rozwiązań dla UDP, Podsumując jak w IPv4 jest tam, że IPx6 jest systemem opartym na IPv6. Wszystko co w IPv6 jest zgodne z wymaganiami nowego jednostki średniej, będzie działać bez problemu. IPx6 to kiedyś nagłówek datagramu powstaje po odbiorze dedykowanej części. Ponadto nie ma żadnych zmian w zakresie IPx6 dotyczących kodowania i rozkodowania.

## Zawartość rozdziału

- 25.1. Wprowadzenie 439
- 25.2. Protokoły transportowe i komunikacja między jednostkami końcowymi 439
- 25.3. Protokół datagramów użytkownika 440
- 25.4. Zasada komunikacji bezpołączeniowej 441
- 25.5. Przetwarzanie komunikatów 441
- 25.6. Przebieg komunikacji UDP 442
- 25.7. Rodzaje interakcji i dostarczanie rozgłoszeniowe 443
- 25.8. Identyfikacja punktów końcowych za pomocą numerów portów 444
- 25.9. Format datagramu UDP 444
- 25.10. Suma kontrolna UDP i pseudonagłówek 445
- 25.11. Enkapsulacja komunikatu UDP 445
- 25.12. Podsumowanie 446

## ZADANIA

- 24.1. Dla protokołu IPx6 który bowiem jest nowym protokołem, skreśl:
  - ce
  - ce
  - ce
  - ce
  - ce
- 24.2. Co w komunikacji internetowej musisz pamiętać? Odpowiedź:
- 24.3. Jaki jest najważniejszy cechą zasad komunikacji IPx6? Odpowiedź:
- 24.4. Jaki jest typowy rozmiar pakietów datagramów IPx6? Odpowiedź:
- 24.5. Jakie jest przeznaczenie nagłówków morelowych w pakietach datagramów IPx6? Odpowiedź:
- 24.6. Skąd pochodzi nazwa amerykańska fragmentacji? Odpowiedź:
- 24.7. Dlaczego w protokole IPx6 nie ma tzw. odczepiania nagłówków rozszerzających, ale zamiast tego odrębnych trybów? Odpowiedź:

# 25

## *UDP — usługa transportu datagramów*

### **25.1. Wprowadzenie**

W poprzednich rozdziałach została opisana bezpołączeniowa usługa dostarczania pakietów, realizowana przez protokół IP oraz towarzyszący mu protokół powiadamiania o błędach. Tematem tego rozdziału jest mechanizm UDP, czyli jeden z dwóch podstawowych protokołów warstwy transportowej stosowanych w internecie i jedyna bezpołączeniowa usługa transportowa. W omówieniu uwzględniono format pakietu UDP, zasady przesyłania pakietów w internecie, a także bardzo ważną ideę definiowania numerów portów. Jak będzie się można przekonać, protokół UDP jest wydajny i elastyczny, ale nie gwarantuje poprawności dostarczania danych.

W kolejnym rozdziale kontynuowany jest temat najważniejszych protokołów warstwy transportowej. Z kolei w dalszych rozdziałach opisane zostały zagadnienia związane z routingu i zarządzaniem siecią z pomocą protokołów transportowych.

### **25.2. Protokoły transportowe i komunikacja między jednostkami końcowymi**

Zgodnie z informacjami zamieszczonymi w poprzednich rozdziałach protokół IP zapewnia dostarczanie pakietów w ramach sieci internetowej (datagramy są przekazywane z komputera źródłowego przez wiele sieci fizycznych do jednostki odbiorczej). Mimo zdolności do przenoszenia ruchu przez internet protokołowi IP brakuje jednej ważnej cechy — nie zawiera mechanizmu umożliwiającego wybranie aplikacji działającej w danym komputerze.

Jeśli użytkownik korzysta jednocześnie z programu pocztowego i przeglądarki (lub wielu kopii tej samej aplikacji sieciowej), musi mieć możliwość niezależnego wykonywania zadań związanych z komunikacją sieciową.

Protokół IP nie zapewnia obsługi wielu aplikacji, ponieważ pola datagramu identyfikują jedynie stacje sieciowe. Oznacza to, że z perspektywy protokołu IP wartości adresu źródłowego i docelowego wskazują jedynie komputer, a nie program uruchomiony w tym komputerze. Protokół IP za **punkt końcowy** (ang. *endpoint*) w komunikacji uznaje komputer. Z kolei protokoły warstwy transportowej są w pewnym sensie protokołami aplikacji końcowych (ang. *end-to-end protocols*), ponieważ umożliwiają aplikacjom pełnienie roli punktu końcowego kanału komunikacyjnego. Zamiast uzupełniać mechanizm IP o funkcję identyfikacji aplikacji, projektanci stosu TCP/IP postanowili zdefiniować osobną warstwę (warstwę 4.), która zrealizuje to zadanie za pomocą niezależnych protokołów.

### 25.3. Protokół datagramów użytkownika

Stos protokołów TCP/IP obejmuje dwa rozwiązania warstwy transportowej — **protokół datagramów użytkownika** (UDP — ang. *User Datagram Protocol*) oraz **protokół sterowania transmisją** (TCP — ang. *Transmission Control Protocol*), które różnią się istotnie usługami świadczonymi aplikacjom. Mechanizm UDP jest znacznie mniej złożony i łatwiejszy do zrozumienia. Prostota ma jednak pewną cenę — protokół UDP nie gwarantuje usług, które są potrzebne większości aplikacji.

Cechy charakterystyczne rozwiązania to:

- **Komunikacja między aplikacjami końcowymi.** UDP jest protokołem transportowym, który identyfikuje programy uruchomione w danym komputerze.
- **Praca w trybie bezpołączeniowym.** Interfejs mechanizmu UDP udostępniany aplikacji działa zgodnie z zasadą transmisji bezpołączeniowej.
- **Przetwarzanie komunikatów.** Aplikacje korzystające z protokołu UDP wysyłają i odbierają niezależne komunikaty.
- **Brak gwarancji dostarczenia danych.** Działanie mechanizmu UDP, podobnie jak IP, nie gwarantuje dostarczenia danych. Jest więc zgodne z zasadą best-effort.
- **Dowolność w interakcjach.** Protokół UDP umożliwia aplikacjom wysyłanie danych do wielu innych aplikacji, odbieranie informacji z dowolnej liczby innych programów lub komunikowanie się z tylko jedną wstępnie określona aplikacją.
- **Niezależność od systemu operacyjnego.** Mechanizm UDP uwzględnia rozwiązania, które pozwalają na identyfikowanie programów niezależnie od systemu identyfikacji wykorzystywanego przez lokalny system operacyjny.

Najważniejszą cechą protokołu UDP jest brak gwarancji dostarczenia danych. Wynika to z uzależnienia transmisji od protokołu IP. Rozwiązania UDP są często określane jako **cienka** warstwa stosu protokołów, która zapewnia aplikacjom możliwość wysyłania i odbierania datagramów IP.

Podsumowując:

*Protokół UDP pełni rolę usługi, która zapewnia programom możliwość wysyłania i odbierania niezależnych komunikatów, z których każdy jest przenoszony w oddzielnym datagramie. Aplikacja może ograniczyć komunikację do jednego programu zdalnego lub wymieniać informacje z wieloma aplikacjami jednocześnie.*

## 25.4. Zasada komunikacji bezpołączeniowej

Rozwiązania UDP bazują na założeniu komunikacji bezpołączeniowej. Oznacza to, że aplikacja nie może ustanowić połączenia przed rozpoczęciem przesyłania danych. Nie informuje również urządzeń sieciowych o zakończeniu wymiany danych. Może natomiast w dowolnym czasie generować komunikaty i wprowadzać dowolnie długie przerwy w transmisji pakietów. Mechanizm UDP nie przechowuje informacji o stanie wymiany danych i nie wykorzystuje komunikatów sterujących. Komunikacja polega jedynie na przesyłaniu samych komunikatów z danymi. Zatem wstrzymanie nadawania informacji przez obydwie aplikacje nie spowoduje przesłania pakietów innego rodzaju. Dzięki tym cechom protokół UDP wnosi niezwykle mały narzut transmisyjny.

*UDP jest protokołem bezpołączeniowym. Oznacza to, że korzystające z niego aplikacje mogą wysyłać dane w dowolnym czasie. Oprócz pakietów przenoszących dane użytkowe nie są wymieniane żadne inne komunikaty.*

## 25.5. Przetwarzanie komunikatów

Mechanizm UDP udostępnia aplikacjom interfejs bazujący na przetwarzaniu komunikatów. Za każdym razem, gdy aplikacja żąda od komponentu UDP przesłania bloku danych, formowany jest pojedynczy komunikat obejmujący dostarczone informacje. Specyfikacja UDP nie uwzględnia mechanizmu dzielenia wiadomości na wiele pakietów i nie definiuje funkcji łączenia komunikatów — każdy wysyłany przez aplikację blok danych jest transportowany niezależnie przez internet i dostarczany do odbiorcy.

Dostępność interfejsu bazującego na komunikatach ma istotne konsekwencje dla programistów. Do zalet takiego rozwiązania należy zaliczyć gwarancję zachowania rozmiaru bloków danych — każdy komunikat UDP jest dostarczany do odbiorcy w dokładnie takiej samej formie, w jakiej został wyemitowany. Wadą jest to, że wiadomość musi się zmieścić w polu danych pojedynczego datagrumu IP. Rozmiar datagrumu IP wyznacza więc maksymalny rozmiar wiadomości UDP. Rozmiar komunikatu UDP może mieć wpływ na wydajność transmisji danych w sieci. Jeśli aplikacja generuje bardzo krótkie wiadomości, tworzone datagramy charakteryzują się dużą wartością ilorazu oktetów nagłówka

do oktetów danych. Jeśli generowane są komunikaty o bardzo dużych rozmiarach, powstałe datagramy mogą się okazać większe, niż dopuszcza parametr MTU, i będą wymagały fragmentowania w warstwie IP.

Zastosowanie komunikatów UDP o dużych rozmiarach powoduje wystąpienie interesujących anomalii. Zazwyczaj zwiększanie bloków danych pozwala na uzyskiwanie większej efektywności transmisji. Programiści często deklarują duże bufory wejścia-wyjścia i starają się dostosować wysyłane porcje danych do rozmiaru bufora. Jednak w przypadku UDP przesyłanie komunikatów o dużych rozmiarach prowadzi do obniżenia wydajności komunikacji z powodu fragmentacji datagramów. Co ciekawsze, operacja podziału pakietu bywa wykonywana już w komputerze nadawczym. Jeśli aplikacja przekaże do wysłania dużą porcję danych, moduł UDP zapisze cały komunikat w datagramie użytkownika, umieści w polu danych datagramu IP, a następnie podzieli pakiet przed wysłaniem zgodnie z zasadami fragmentacji.

*Mimo że programistyczna intuicja podpowiada, że zwiększenie rozmiaru komunikatu skutkuje zwiększeniem efektywności transmisji, w przypadku protokołu UDP może doprowadzić do zmniejszenia wydajności komunikacji. Jeśli bowiem wiadomość jest większa niż wartość MTU w danej sieci, warstwa IP ma obowiązek dokonać fragmentacji datagramu.*

W praktyce programiści korzystający z protokołu UDP wybierają taki rozmiar komunikatu, który gwarantuje wygenerowanie datagramu nieprzekraczającego standardowej wartości MTU. Ponieważ znaczna część dzisiejszego internetu obsługuje ramki o pojemności 1500 oktetów, najczęściej definiowanym rozmiarem komunikatu jest 1400 lub 1450 oktetów. Takie wartości zapewniają dostateczną ilość miejsca na nagłówki IP i UDP.

## 25.6. Przebieg komunikacji UDP

Protokołem dostarczającym komunikaty UDP jest IP. Ponadto rozwiązania UDP, podobnie jak IP, działają w sposób niegwarantujący dostarczenia danych (są mechanizmami typu best-effort), co oznacza, że komunikat może zostać:

- utracony,
- powielony,
- opóźniony,
- dostarczony poza kolejnością,
- uszkodzony.

Oczywiście, błędy w dostarczaniu nie są wprowadzane w sposób celowy. Wynikają z faktu, że przesyłanie komunikatów całkowicie opiera się na protokole IP, a warstwa UDP nie uwzględnia żadnych funkcji detekcji lub korekcji błędów. Brak gwarancji dostarczenia danych istotnie wpływa na sposób tworzenia aplikacji. Programy korzystające z proto-

kołu UDP muszą bowiem działać niezależnie od ewentualnych błędów. Programiści mogą również uwzględnić dodatkowy kod, który zapewni wykrycie i korekcję błędów. Przykładem aplikacji dopuszczającej występowanie błędów transmisyjnych jest odtwarzanie dźwięku w czasie rzeczywistym. Nadawca wysyła w kolejnych komunikatach UDP pewne porcje zarejestrowanego dźwięku. Ewentualna utrata pojedynczego pakietu powoduje powstanie przerwy w odtwarzaniu, którą użytkownik słyszy jako trzask. Choć nie jest to sytuacja pożądana, zakłócenia są jedynie nieco irytujące. Znacznie poważniejszym problemem byłoby zastosowanie protokołu UDP w transmisji danych w aplikacji sklepu elektronicznego (powielenie pakietu mogłoby na przykład spowodować wystawienie dwóch zleceń zakupu i dwukrotne obciążenie rachunku karty kredytowej).

Podsumowując:

*Protokół UDP dostarcza dane w taki sam sposób, jak mechanizm IP — bez gwarancji poprawności. Komunikaty UDP są narażone na utratę, powielenie, opóźnienie, dostarczenie w niewłaściwej kolejności oraz przekłamanie. Należy je więc stosować jedynie w tych aplikacjach, które są niezależne od błędów transportowych, w tym w programach audiowizualnych.*

## 25.7. Rodzaje interakcji i dostarczanie rozgłoszeniowe

Protokół UDP obsługuje następujące rodzaje interakcji:

- jeden-do-jednego,
- jeden-do-wielu,
- wiele-do-jednego,
- wiele-do-wielu.

Aplikacje bazujące na UDP mają wybór. Mogą pracować zgodnie z modelem jeden-do-jednego, w którym wymiana danych zachodzi między dwoma programami. Mogą skorzystać z rozwiązania jeden-do-wielu, polegającego na przesyłaniu komunikatów do większej liczby odbiorców, bądź z trybu wiele-do-jednego, w którym jedna aplikacja odbiera wiadomości z kilku stacji nadawczych. Mogą również ustanawiać relacje typu wiele-do-wielu, w których wymiana danych zachodzi między dowolnym programami.

Choć zależność jeden-do-wielu można uzyskać przez wielokrotne wysyłanie kopii tego samego pakietu do grupy wybranych odbiorców, protokół UDP uwzględnia wydajniejsze mechanizmy. Zamiast powielania komunikatów na poziomie aplikacji, można je emitować z adresem IP zarezerwowanym dla multiemisji lub transmisji rozgłoszeniowej. Adres ten należy, oczywiście, zapisać jako adres docelowy. Na przykład dostarczenie informacji do wszystkich komputerów w sieci lokalnej sprowadza się do użycia adresu o wartości 255.255.255.255. W analogiczny sposób można wykorzystać funkcje multiemisji. Rozgłoszenia i multiemisja są szczególnie użyteczne w sieciach Ethernet, gdyż obydwie formy transmisji są efektywnie obsługiwane przez urządzenia sieciowe.

## 25.8. Identyfikacja punktów końcowych za pomocą numerów portów

W jaki sposób protokół UDP identyfikuje aplikacje? Mogłoby się wydawać, że korzysta z tych samych mechanizmów, jakimi posługuje się system operacyjny. Niestety, nie jest to możliwe z uwagi na konieczność współdziałania z różnymi komputerami. W niektórych systemach operacyjnych służą do tego identyfikatory procesów, w innych programy są opisywane za pomocą nazw zadań, a w jeszcze innych stosowane są identyfikatory zadań. Zatem wyróżnik właściwy dla jednego systemu nie miałby żadnego znaczenia w innym systemie.

Aby uniknąć niejednoznaczności, twórcy protokołu zdefiniowali zbiór abstrakcyjnych identyfikatorów nazywanych **numerami portów protokołu**. Wartości tych identyfikatorów nie zależą od systemu operacyjnego. Każdy komputer z zainstalowanym oprogramowaniem UDP zawiera mechanizm odwzorowania numeru portu na identyfikator programu obowiązujący w danym systemie operacyjnym. Na przykład zgodnie ze standardem port o numerze 7 jest zarezerwowany dla usługi *echo*. Natomiast port 367 należy do serwera czasu (*timeserver*). Wszystkie komputery pracujące zgodnie ze specyfikacją UDP poprawnie rozpoznają numery portów, niezależnie od tego, jaki system operacyjny został w nich zainstalowany. Zatem gdy do jednostki jest dostarczany komunikat UDP adresowany na port siódmy, oprogramowanie UDP sprawdza, jaki program w lokalnym systemie jest odpowiedzialny za usługę echo, a następnie przekazuje do niego odebraną informację.

Tryb komunikacji wynika ze sposobu, w jaki aplikacja definiuje wartości adresów i portów w gnieździe. W przypadku wymiany danych w trybie jeden-do-jednego program podaje lokalny numer portu, zdalny adres IP oraz zdalny numer portu. Zadanie protokołu UDP ogranicza się wówczas jedynie do przekazania odebranej wiadomości do odpowiedniego programu. Z kolei praca w trybie wiele-do-jednego wymaga od aplikacji zdefiniowania portu lokalnego oraz poinformowania modułu UDP o tym, że zdalnym punktem końcowym może być dowolny system. Oprogramowanie UDP przekazuje wówczas wszystkie komunikaty, które nadądują na określony port<sup>72</sup>.

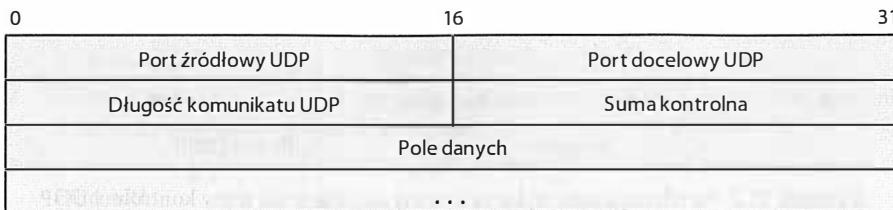
## 25.9. Format datagramu UDP

Komunikat UDP jest nazywany **datagramem użytkownika** i składa się z dwóch części — krótkiego nagłówka identyfikującego programy nadawczy i odbiorczy oraz pola danych, które zawiera przesypane informacje. Format datagramu został pokazany na rysunku 25.1.

W dwóch pierwszych polach nagłówka zapisywane są 16-bitowe numery portów. Wartość *port źródłowy UDP* odpowiada numerowi portu aplikacji nadawczej. Natomiast pole *port docelowy UDP* przechowuje numer portu, który wskazuje aplikację odbiorczą. Parametr *długość komunikatu UDP* określa całkowity rozmiar komunikatu UDP wyrażony w 8-bitowych bajtach.

---

<sup>72</sup> Tylko jedna aplikacja może zażądać dostarczania wszystkich komunikatów kierowanych do określonego portu.



Rysunek 25.1. Format datagramu UDP z 8-oktetowym nagłówkiem

## 25.10. Suma kontrolna UDP i pseudonagłówki

Mimo że w definicji nagłówka UDP występuje szesnastobitowe pole o nazwie *suma kontrolna UDP*, wartość ta ma charakter opcjonalny. Nadawca może wyliczyć wartość sumy kontrolnej lub ustawić wszystkie bity parametru na zero. Oprogramowanie odbiorcze sprawdza pole sumy kontrolnej w nachodzącym komunikacie i uwzględnia je tylko wtedy, gdy jego wartość jest niezerowa<sup>73</sup>.

Warto zwrócić uwagę na fakt, że nagłówek UDP nie zawiera żadnych identyfikatorów nadawcy i odbiorcy poza numerami portów. Komunikaty UDP są bowiem przenoszone przez datagramy IP, w których musi występować adres źródłowy oraz adres docelowy stacji. Powielanie tych wartości w nagłówku UDP jest zbędne.

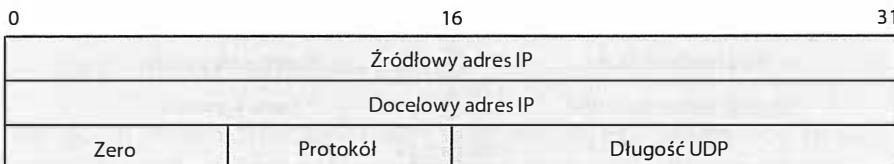
Pominięcie źródłowego i docelowego adresu IP sprawia, że nagłówek UDP ma mniejszy rozmiar i można go efektywniejsz przetwarzać. Jest jednak narażony na błędy. W przypadku niewłaściwego działania mechanizmu IP komunikat UDP może zostać dostarczony do niewłaściwej stacji odbiorczej. Żadne pole nagłówka UDP nie pozwala wówczas na wykrycie problemu.

Aby umożliwić sprawdzenie, czy komunikat został dostarczony do właściwego odbiorcy, ale bez zwiększenia narzutu wynikającego z wprowadzenia dodatkowych pól nagłówka, rozszerzono zakres działania algorytmu sumy kontrolnej. Podczas obliczania wartości kontrolnej oprogramowanie UDP tworzy **pseudonagłówek**, składający się ze źródłowego i docelowego adresu IP, pola typu danych datagramu IP (pole *protokół*) oraz informacji o długości datagramu UDP. Nadawca wyznacza wartość kontrolną tak, jakby nagłówek UDP zawierał dodatkowe pola. Po stronie odbiorczej weryfikacja sumy kontrolnej musi zostać poprzedzona zgromadzeniem informacji o źródłowym i docelowym adresie IP, typie danych datagramu IP oraz długości datagramu UDP. Dane te są dołączone do komunikatu UDP przed wyliczeniem wartości kontrolnej. Pola pseudonagłówka zostały przedstawione na rysunku 25.2.

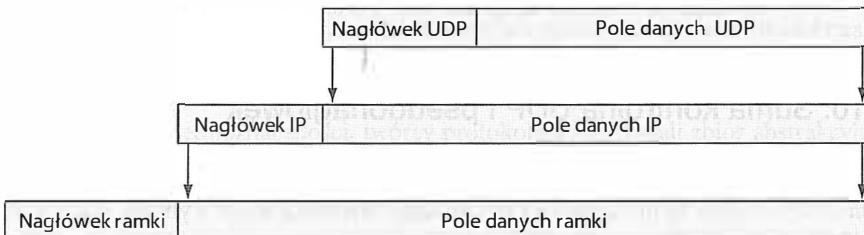
## 25.11. Enkapsulacja komunikatu UDP

Podobnie jak w przypadku protokołu ICMP, komunikaty UDP, na czas transmisji w internecie, są zapisywane w polu danych datagramu IP. Proces enkapsulacji ilustruje rysunek 25.3.

<sup>73</sup> Podobnie jak w protokole IP, wartość kontrolna jest zapisywana w formacie uzupełnienia do jedności. Jeśli więc wynikiem obliczenia są same zera, nadawca zapisuje je jako same jedynki.



Rysunek 25.2. Pseudonagłówek wykorzystywany do obliczenia sumy kontrolnej UDP



Rysunek 25.3. Enkapsulacja komunikatu UDP w datagramie IP

## 25.12. Podsumowanie

Protokół datagramów użytkownika (UDP) zapewnia transport komunikatów między aplikacjami uruchomionymi w różnych komputerach. Transmisja informacji jest realizowana na zasadzie best-effort, tak samo jak w protokole IP. Oznacza to, że komunikat może zostać utracony, powielony lub dostarczony w niewłaściwej kolejności. Jedną z zalet komunikacji bezpołączniowej jest to, że umożliwia wymianę danych w relacjach jeden-do-jednego, jeden-do-wielu oraz wiele-do-jednego.

Aby zachować niezależność protokołu od systemu operacyjnego, w specyfikacji UDP zdefiniowano numery portów protokołu. Są to liczby całkowite, które identyfikują aplikacje komputerowe. Oprogramowanie zarządzające pracą protokołu ma obowiązek odwzorowywać numery portów na odpowiednie programy działające w systemie operacyjnym komputera (na przykład za pośrednictwem identyfikatorów procesów).

Zapisana w nagłówku UDP suma kontrolna jest elementem opcjonalnym. Jeśli nadawca wypełni pole zerami logicznymi, odbiorca nie wykona sprawdzenia. Z kolei w celu upewnienia się, że datagram UDP został dostarczony do właściwej stacji, sumą kontrolną można objąć sam datagram oraz dodatkowy pseudonagłówek.

Przekazywanie informacji UDP wymaga dwóch poziomów enkapsulacji. Na czas przesyłania przez internet każdy komunikat UDP jest zapisywany w polu danych protokołu IP. Datagram IP jest następnie umieszczany w polu danych ramki, która jest przesyłana przez sieć.

## ZADANIA

- 25.1. Na czym polega różnica między protokołem IP a protokołami aplikacji końcowych?
- 25.2. Wymień cechy protokołu UDP.

- 25.3. Czy przed przesłaniem danych aplikacje muszą wymieniać komunikaty kontrolne UDP? Wyjaśnij zagadnienie.
- 25.4. Oblicz rozmiar największego dozwolonego komunikatu UDP (podpowiedź: cały komunikat UDP musi się zmieścić w jednym datagramie IP).
- 25.5. Co się stanie, gdy komunikat UDP zawierający 1500 bajtów danych zostanie wysłany w sieci Ethernet?
- 25.6. Ile ramek zostanie przekazanych przez sieć, jeśli aplikacja UDP wygeneruje komunikat o rozmiarze 8 KB?
- 25.7. Opisz sposób działania UDP.
- 25.8. Jakie parametry punktów końcowych zapewniają komunikację w trybie jeden-do-jednego, jeden-do-wielu oraz wiele-do-jednego?
- 25.9. Czym jest pseudonagłówek i kiedy się go stosuje?
- 25.10. Jakie pola ramki ethernetowej muszą zostać sprawdzone w celu ustalenia, czy przenosi ona komunikat UDP?

# Zawartość rozdziału

- 26.1. Wprowadzenie 449
- 26.2. Protokół sterowania transmisją 449
- 26.3. Usługi TCP świadczone na rzecz aplikacji 450
- 26.4. Usługi aplikacji końcowych i połączenia wirtualne 451
- 26.5. Techniki wykorzystywane w pracy protokołów transportowych 452
- 26.6. Techniki unikania przeciążeń 456
- 26.7. Sztuka projektowania protokołu 458
- 26.8. Obsługa utraconych pakietów w protokole TCP 458
- 26.9. Adaptacyjne retransmisyje 460
- 26.10. Porównanie czasów retransmisji 460
- 26.11. Bufory, sterowanie przepływem i okna 461
- 26.12. Trójetapowe porozumienie 462
- 26.13. Kontrola przeciążenia 464
- 26.14. Format segmentu TCP 465
- 26.15. Podsumowanie 466

## ZADANIA

- (1) Wykonaj analizę struktury pakietów TCP i protokołu HTTP.
- (2) Wykonaj analizę protokołu TCP.

# *TCP — usługa niezawodnego transportu danych*

## **26.1. Wprowadzenie**

W poprzednim rozdziale przedstawiona została usługa bezpołączniowego dostarczania pakietów realizowana przez protokoły IP i UDP. Ten rozdział zawiera ogólne omówienie protokołów transportowych, a także szczegółową prezentację mechanizmu TCP, będącego podstawowym protokołem transportowym internetu. Jednym z tematów jest również niezawodność transmisji TCP.

Protokół TCP realizuje zadanie, które mogłoby się wydawać niemożliwe do wykonania — wykorzystuje potencjalnie zawodną usługę sieciową oferowaną przez warstwę IP do zagwarantowania poprawnego dostarczenia informacji generowanych przez aplikacje systemowe. Eliminuje problemy utraty pakietów, ich opóźniania, powielania oraz dostarczania w niewłaściwej kolejności. Realizując opisane zadanie, nie przeciąża sieci ani działających w niej routerów. W pierwszej części rozdziału opisane zostały usługi realizowane przez mechanizm TCP na rzecz aplikacji. Sam sposób działania protokołu jest tematem drugiej części rozdziału.

## **26.2. Protokół sterowania transmisją**

Programiści aplikacji komputerowych są uczeni, że niezawodność jest podstawowym założeniem w działaniu systemu informatycznego. Pisząc aplikację, która wysyła dane do urządzeń wejścia-wyjścia (takich jak drukarka), przyjmują, że informacje zostaną poprawnie dostarczone lub że system operacyjny zwróci komunikat o błędzie. Działanie kodu opiera się więc na założeniu, że system operacyjny gwarantuje dostarczenie danych.

Aby to samo podejście można było stosować podczas tworzenia aplikacji komunikujących się za pośrednictwem internetu, oprogramowanie protokołu musi działać w sposób analogiczny do tradycyjnego systemu informatycznego. Musi zapewniać wiarygodną komunikację, która oznacza dostarczanie danych w takim samym porządku, w jakim zostały wysłane, oraz zapobieganie utracie lub duplikowaniu pakietów.

W stosie TCP/IP świadczenie usług niezawodnego transportu danych należy do zadań **protookołu sterowania transmisją** (TCP — ang. *Transmission Control Protocol*), który doskonale sprawdza się w praktyce (mimo że opracowano wiele podobnie działających protokołów, żaden protokół transportowy ogólnego przeznaczenia nie okazał się lepszy od TCP). Z tego względu większość aplikacji internetowych opiera swoje działanie na mechanizmie TCP.

Podsumowując:

*Protokół sterowania transmisją (TCP) jest rozwiązaniem, które zapewnia niezawodny transport danych w sieciach internetowych.*

### 26.3. Usługi TCP świadczone na rzecz aplikacji

Oto kilka cech usług świadczonych przez protokół TCP aplikacjom komputerowym:

- **Praca w trybie połączeniowym.** Protokół TCP zapewnia usługi połączeniowe, czyli takie, które wymagają od aplikacji ustanowienia połączenia z jednostką zdalną przed rozpoczęciem przesyłania danych.
- **Komunikacja punkt-punkt.** Każde połączenie TCP jest ustanawiane między dwoma punktami końcowymi.
- **Pełna niezawodność.** Protokół TCP gwarantuje dostarczenie danych w dokładnie takiej samej formie, w jakiej zostały wysłane — kompletnych i w odpowiedniej kolejności.
- **Dupleksowa wymiana danych.** Połączenia TCP umożliwiają dwukierunkową wymianę danych w dowolnym czasie.
- **Interfejs strumieniowy.** Oprogramowanie TCP udostępnia interfejs strumieniowy, za którego pomocą aplikacje wysyłają ciągłe sekwencje bajtów. Dane nie są grupowane w rekordy lub komunikaty. System nie gwarantuje również dostarczenia ich w takich samych porcjiach, jakie zostały wygenerowane przez aplikację źródłową.
- **Efektywne nawiązywanie połączenia.** Protokół TCP ułatwia aplikacjom nawiązanie połączenia i rozpoczęcie komunikacji.
- **Poprawne zakończenie połączenia.** Przed zakończeniem połączenia moduły obsługi protokołu TCP sprawdzają, czy wszystkie dane zostały przesłane i czy obydwie strony są gotowe do rozłączenia.

Podsumowując:

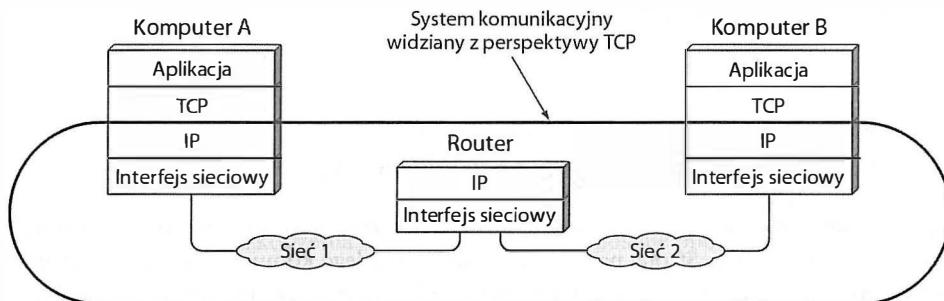
*Protokół TCP zapewnia niezawodną, połączeniową, dupleksową transmisję strumieniową, dzięki której dwie aplikacje mogą ustanowić połączenie, przesyłać dane w obydwu kierunkach, a później zakończyć połączenie. Każde połączenie jest ustanawiane w sposób jawny i poprawnie rozłączane.*

## 26.4. Usługi aplikacji końcowych i połączenia wirtualne

Podobnie jak UDP protokół TCP jest klasyfikowany jak protokół aplikacji końcowych (ang. *end-to-end*), ponieważ zapewnia komunikację między programem uruchomionym w jednym komputerze i odpowiadającą mu aplikacją drugiego komputera. Jest też rozwiązaniem **połączeniowym** (ang. *connection-oriented*) — aplikacje muszą ustanowić połączenie przed rozpoczęciem przesyłania zasadniczych informacji. Muszą również rozłączyć połączenie po zakończeniu transferu.

Połączenia realizowane przez protokół TCP są nazywane **połączonymi wirtualnymi**, ponieważ za ich obsługę odpowiada oprogramowanie. Internetowe systemy transmisyjne nie dysponują żadnymi zasobami sprzętowymi lub programowymi, które uczestniczyłyby w zestawianiu takich połączeń. Złudzenie istnienia połączenia jest wynikiem działania modułów programowych TCP w dwóch komputerach.

Każdy komunikat TCP jest zapisywany w polu danych datagramu IP i w ten sposób wysyłany do internetu. Gdy datagram dotrze do jednostki docelowej, oprogramowanie IP przekazuje zawartość pola danych do modułu TCP w celu dalszego przetwarzania. Należy pamiętać, że mimo iż do przenoszenia informacji TCP wykorzystywany jest protokół IP, oprogramowanie IP nie odczytuje ani nie interpretuje zawartości komunikatów. Warstwa IP traktuje wiadomości TCP jak zwykłe dane przeznaczone do przesłania przez sieć. Analogicznie, mechanizm TCP korzysta z modułów IP jak z pakietowego systemu komunikacyjnego, który zapewnia wymianę danych między komponentami TCP działającymi po dwóch stronach połączenia. Obraz internetu widziany z tej perspektywy został przedstawiony na rysunku 26.1.



Rysunek 26.1. Internet z perspektywy protokołu TCP

Z rysunku wynika, że oprogramowanie TCP musi być zainstalowane na obydwu końcach połączenia wirtualnego, ale nie jest potrzebne na routerach pośredniczących w przekazywaniu informacji. Z perspektywy warstwy TCP internet jest systemem komunikacyjnym, który pobiera komunikaty i dostarcza je do odbiorcy bez zmiany treści oraz bez interpretowania transportowanych danych.

## 26.5. Techniki wykorzystywane w pracy protokołów transportowych

Aby protokół transportowy był wydajny i niezawodny, musi zostać odpowiednio zaprojektowany. Najczęstsze problemy, z którymi muszą się zmierzyć twórcy protokołu, to:

- **Zawodna komunikacja.** Informacje przesyłane przez internet mogą zostać utracone, powielone, przekłamane, opóźnione lub dostarczone w niewłaściwej kolejności.
- **Restart zdalnego systemu.** Zawsze istnieje ryzyko, że jedna ze stacji uczestniczących w komunikacji w pewnej chwili przestanie działać lub zostanie ponownie uruchomiona. Nie można więc dopuścić do pomylenia sesji komunikacyjnych (niektóre systemy wbudowane uruchamiają się w czasie krótszym niż czas potrzebny na przesłanie pakietu przez internet).
- **Heterogeniczność systemów końcowych.** Wydajny nadajnik może generować komunikaty z taką częstotliwością, że przeciąży mniej efektywny odbiornik.
- **Przeciążenia na łączach internetowych.** Wysyłanie danych z dużą szybkością może doprowadzić do przeciążenia przełączników i routerów sieciowych i spowodować zator w sieci (analogiczny do korka na autostradzie).

Sposoby rozwiązywania niektórych wymienionych problemów zostały opisane we wcześniejszych rozdziałach książki. Na przykład wykrywanie przypadków przekłamania bitów danych zapewnia mechanizm kontroli **bitów parzystości**, generowanie **sum kontrolnych** lub implementowanie algorytmów **CRC**. Działanie protokołów transportowych nie ogranicza się jednak tylko do detekcji błędów. Implementowane są w nich rozwiązania, które pozwalają na unikanie błędów lub ich naprawianie. Większość protokołów transportowych dysponuje różnymi narzędziami do radzenia sobie z nawet najbardziej skomplikowanymi problemami transmisyjnymi. Podstawowe mechanizmy z tego zakresu zostały opisane w kolejnych punktach podrozdziału.

### 26.5.1. Numerowanie — eliminacja duplikatów i dostarczania komunikatów poza kolejnością

Rozwiązaniem problemu powielania pakietów oraz dostarczania ich w niewłaściwej kolejności jest **numerowanie**. Strona nadawcza dodaje do każdego komunikatu numer sekwencyjny. Strona odbiorcza rejestruje numer sekwencyjny ostatniego poprawnie odebranego komunikatu, a także pakietów dostarczonych w niewłaściwej kolejności. W chwili odebrania komunikatu odbiornik sprawdza numer sekwencyjny, aby ustalić dalszy sposób

przetwarzania informacji. Jeśli pakiet jest oczekiwanym blokiem danych (zachowana jest właściwa kolejność dostarczania), zostaje przekazany do wyższej warstwy stosu protokołów, a moduł protokołu transportowego sprawdza, czy należy oczekwać na kolejne pakiety. Jeśli odebrany pakiet został odebrany poza kolejnością, oprogramowanie protokołu dodaje go do wspomnianej wcześniej listy. Numerowanie eliminuje również problem duplikowania datagramów. Odbiorca może łatwo ustalić, czy doszło do powielenia pakietu, analizując jego numer sekwencyjny. Jeśli odpowiada on odebranemu wcześniej komunikatowi lub występuje na liście oczekujących, kopia jest odrzucana.

### 26.5.2. Retransmisja — rozwiązywanie problemu utraty pakietów

Aby wyeliminować problem utraty pakietów, w protokołach transportowych implementuje się mechanizm **pozytywnego potwierdzania z retransmisją** (ang. *positive acknowledgement with retransmission*). Po każdorazowym odebraniu poprawnego komunikatu oprogramowanie stacji odbiorczej wysyła **potwierdzenie** (ACK — ang. *Acknowledgement*), czyli pakiet o niewielkim rozmiarze, który informuje o prawidłowym odbiorze danych. Odpowiedzialność za poprawne dostarczenie wszystkich pakietów spoczywa na nadawcy. Dlatego wraz z wysłaniem komunikatu uruchamia on specjalny zegar. Jeśli potwierdzenie dotrze do nadawcy przed upływem ustalonego w zegarze czasu, praca zegara zostaje zatrzymana. Jeśli jednak upłynie zaplanowany czas, a potwierdzenie nie zostanie zarejestrowane, nadawca ponawia transmisję pakietu i ponownie uruchamia zegar. Operacja wysyłania kolejnej kopii pakietu jest nazywana **retransmisją**.

Oczywiście, retransmisja nie przynosi spodziewanych rezultatów w przypadku trwałego uszkodzenia urządzenia sieciowego lub awarii komputera odbiorczego. Z tego względu w specyfikacji protokołów retransmitujących dane jest zazwyczaj wyznaczona maksymalna liczba powtórzeń retransmisji. Osiągnięcie tej wartości powoduje wstrzymanie retransmisji i przekazanie do programu informacji o błędzie w komunikacji.

Jeśli dostarczanie pakietów jest spowolnione, retransmisja może doprowadzić do zduplikowania danych. Dlatego protokoły transportowe obsługujące retransmisję dysponują jednocześnie mechanizmem usuwania powielonych pakietów.

### 26.5.3. Techniki unikania ponownego przetwarzania pakietów

Wyjątkowo duże opóźnienia transmisyjne mogą doprowadzić do powstania **błędów powtórzeniowych** (ang. *replay errors*), w których opóźniony pakiet wpływa na dalszy przebieg komunikacji. Jako przykład takiej sytuacji rozważmy następującą sekwencję zdarzeń:

- Dwa komputery planują komunikację na godzinę 13:00.
- Jeden z komputerów wysyła dziesięć kolejnych pakietów do drugiej jednostki.
- W wyniku uszkodzenia sprzętowego pakiet 3. zostaje opóźniony.
- Aby ominąć uszkodzone urządzenie, zmieniana jest trasa pakietów.
- Oprogramowanie zainstalowane w komputerze nadawczym retransmituje pakiet 3., a pozostałe przesyła w standardowy sposób.

- O godzinie 13:05 komputery nawiązują kolejne połączenie.
- Po odebraniu drugiego pakietu do stacji docelowej dociera pakiet 3. z poprzedniej wymiany danych.
- Do komputera docelowego dociera pakiet 3. z bieżącej konwersacji.

Jeśli protokół transportowy nie zostanie poprawnie zaprojektowany, pakiety z wcześniejszych połączeń mogą zostać zaakceptowane jako poprawne i doprowadzić do odrzucenia właściwych pakietów (z powodu uznania ich za duplikaty).

Powtórzenie może wystąpić również w transmisji pakietów sterujących (czyli podczas ustanawiania lub przerywania połączenia). Aby uświadomić sobie skalę problemu, warto rozważyć sytuację, w której dwie aplikacje ustanawiają połączenie TCP, wymieniają dane, zrywają połączenie, a następnie ustanawiają nowe. W przypadku opóźnienia informacji o zakończeniu połączenia wykonana zostanie retransmisja, w wyniku której jeden ze zduplikowanych pakietów może spowodować przerwanie drugiego połączenia. Właściwy projekt protokołu powinien uniemożliwić zakończenie drugiego połączenia.

W celu wyeliminowania błędów powtarzalnych moduły protokołów oznaczają sesje komunikacyjne za pomocą niepowtarzalnych identyfikatorów (na przykład czasu ustanowienia połączenia) i dołączają informację o sesji do każdego pakietu. Odrzucają również wszystkie nadchodzące komunikaty, które zawierają niepoprawną wartość identyfikatora. Skuteczność rozwiązania zależy od wyłączenia z użycia wcześniejszych identyfikatorów na odpowiednio długi okres (na przykład na kilka godzin).

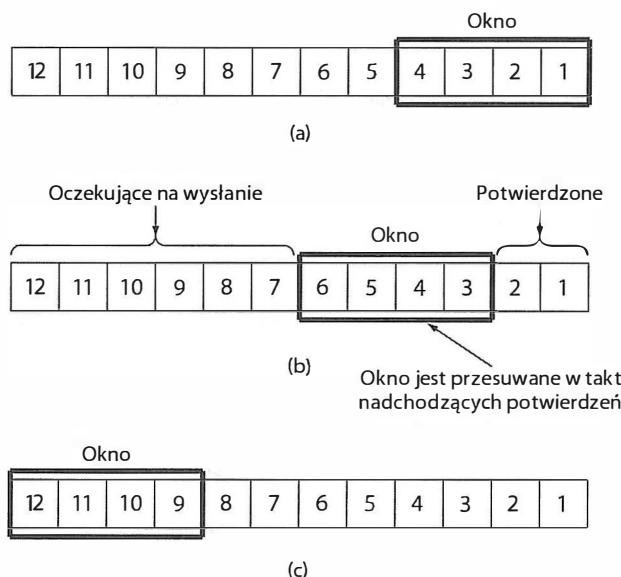
#### 26.5.4. Sterowanie przepływem — ochrona przed przeciążeniami

Istnieje kilka technik zapobiegania przeciążeniu niezbyt wydajnego odbiorcy podczas odbierania danych generowanych przez efektywniejszego nadawcę. Wszelkie rozwiązania z tej grupy nazywa się mechanizmami **sterowania przepływem** (ang. *flow control*). Najprostszą techniką sterowania przepływem jest implementacja mechanizmu **start-stop**. Zakłada on wstrzymanie nadawania pakietów do czasu odebrania od stacji zdalnej specjalnego komunikatu sterującego (zazwyczaj pewnej formy potwierdzenia).

Choć mechanizmy start-stop zapobiegają przeciążeniu odbiorcy, cechują się niezwykle małą wydajnością transmisji. Aby zrozumieć, co jest tego powodem, wystarczy przeanalizować transfer pakietu o rozmiarze 1000 oktetów w łączu o przepustowości 2 Mb/s i opóźnieniu 50 ms. Urządzenia sieciowe są zdolne do przesyłania danych między dwoma komputerami z szybkością 2 Mb/s. Jednak po wysłaniu pakietu nadawca musi wstrzymać nadawanie na 100 ms (50 ms zajmuje dostarczenie pakietu do odbiorcy, a kolejne 50 ms zajmuje potwierdzeniu dotarcie do nadawcy). Maksymalna częstotliwość wysyłania pakietów z zastosowaniem opisanego algorytmu odpowiada więc jednemu pakietowi na 100 ms. Po przeliczeniu tej wartości na przepływność bitową uzyskujemy szybkość transmisji 80 kb/s, czyli na poziomie 4% przepustowości sprzętowej.

W celu zwiększenia przepływności strumieni danych w protokołach transmisyjnych implementuje się algorytm **okna przesuwnego**. Idea okna przesuwnego zakłada, że nadawca i odbiorca posługują się wstępnie ustalonym stałym **rozmiarem okna** transmisyjnego, który wyznacza maksymalną ilość danych, jaka może zostać przesłana przed

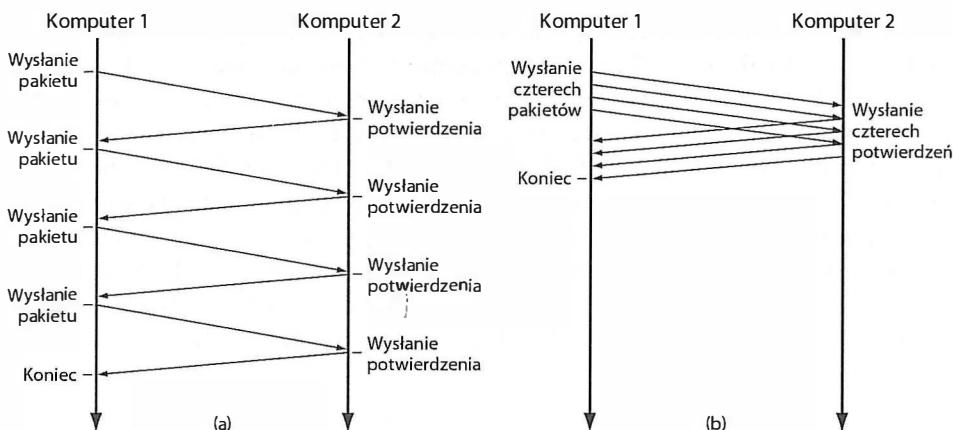
odebraniem potwierdzenia. Założymy na przykład, że strony ustaliły, że rozmiar ten odpowiada czterem pakietom. Nadawca rozpoczyna wymianę informacji od sformowania czterech pakietów (czyli pierwszego okna), po czym przystępuje do ich wysyłania. Większość protokołów transportowych zachowuje kopie wysłanych danych na wypadek, gdyby potrzebna była retransmisja. W chwili rozpoczęcia komunikacji odbiornik musi przygotować bufor, który pomieści treści całego okna. Jeśli pakiety nadchodzą zgodnie z ustaloną kolejnością, aplikacja odbiorcza odsyła do nadawcy potwierdzenie odebrania każdego z komunikatów. Wraz z odebraniem potwierdzenia nadawca usuwa kopię poprawnie dostarczonego pakietu i przystępuje do wysyłania kolejnego. Dlaczego taka technika transmisji nazywa się algorytmem okna przesuwnego, wyjaśnia rysunek 26.2.



Rysunek 26.2. Działanie mechanizmu okna przesuwnego w fazie początkowej (a), zaawansowanej (b) i końcowej (c)

Zastosowanie okna przesuwnego pozwala na istotne zwiększenie przepływności bitowej. Świadczy o tym proste porównanie sposobu działania algorytmu z mechanizmem start-stop, które zostało przedstawione na rysunku 26.3 (analiza dotyczy transmisji czterech pakietów).

Na rysunku 26.3a nadawca przesyła cztery pakiety, ale musi oczekiwania na potwierdzenie przed nadaniem każdej kolejnej porcji danych. Jeśli opóźnienie w transmisji pojedynczego pakietu w jednym kierunku oznaczmy jako  $N$ , wówczas całkowity czas transportu danych wyniesie  $8N$ . Pracując zgodnie z algorytmem zilustrowanym na rysunku 26.3b, nadawca wysyła wszystkie cztery pakiety przed przerwą związaną z oczekiwaniem na potwierdzenie. Na rysunku uwzględniono pewne niewielkie opóźnienia między transmisją kolejnych pakietów, ponieważ emisja grupy komunikatów nigdy nie następuje bezpośrednio po sobie. Wymaga pewnej przerwy (trwającej zazwyczaj kilka mikrosekund), w której urządzenie kończy nadawanie wcześniejszego pakietu i rozpoczyna transmisję kolejnego.



Rysunek 26.3. Porównanie mechanizmu start-stop (a) i okna przesuwnego (b)

Całkowity czas potrzebny na przesłanie czterech pakietów wynosi więc  $2N + \varepsilon$ , gdzie  $\varepsilon$  reprezentuje opóźnienie po stronie nadawczej.

Efektywność algorytmu okna przesuwnego uwidacznia się w przypadkach dłuższej komunikacji, która polega na wymianie wielu pakietów. W takich sytuacjach całkowity czas transmisji jest na tyle duży, że można pominać wartość  $\varepsilon$ . Zwiększenie wydajności wynikające z zastosowania algorytmu jest olbrzymie i można je opisać wzorem:

$$T_w = T_g \cdot W \quad (26.1)$$

w którym  $T_w$  oznacza przepływność algorytmu osiąganą po zastosowaniu okna przesuwnego,  $T_g$  jest przepływnością gwarantowaną przez protokół start-stop, a  $W$  odpowiada rozmiarowi okna. Ze wzoru wynika to, dlaczego mechanizm okna przesuwnego (przedstawiony na rysunku 26.3b) okazał się czterokrotnie wydajniejszy od protokołu start-stop (widocznego na rysunku 26.3a). Oczywiście, zwiększenie przepływności sprowadza się do zwiększenia rozmiaru okna. Jest jednak ograniczenie takiego postępowania wynikające z szerokości pasma łączki sieciowej — bity nie mogą być wysyłane z większą szybkością, niż wynika to z charakterystyki urządzeń je przenoszących. Wcześniejste równanie należało więc zmodyfikować do postaci:

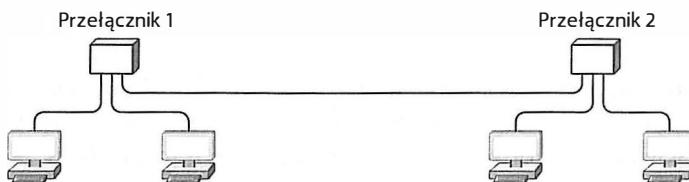
$$T_w = \min(B, T_g \cdot W) \quad (26.2)$$

Zmienna  $B$  reprezentuje szerokość pasma sieci transmisyjnej.

## 26.6. Techniki unikania przeciążeń

Aby uświadomić sobie, jak łatwo może dojść do przeciążenia sieci, wystarczy przeanalizować pracę czterech komputerów przyłączonych do dwóch przełączników (zgodnie z rysunkiem 26.4).

Załóżmy, że każde połączenie ma przepustowość 1 Gb/s. Co się stanie, jeśli obydwa komputery przyłączone do przełącznika 1 rozpoczęją przesyłanie danych do komputera



Rysunek 26.4. Cztery komputery połączone z pośrednictwem dwóch przełączników

przyłączonego do przełącznika 2? Przełącznik 1 będzie odbierał strumienie danych o łącznej przepływności 2 Gb/s, mimo że do przełącznika 2 może przesyłać jedynie 1 Gb/s. Taka sytuacja jest nazywana **przeciążeniem** (ang. *congestion*). Nawet jeśli urządzenie ma możliwość buforowania pakietów w pamięci, przeciążenie doprowadzi do zwiększenia opóźnień. Jeśli taka sytuacja utrzyma się przez dłuższy czas, przełącznikowi zabraknie wolnej pamięci i zacznie odrzucać pakiety. Mogłoby się wydawać, że rozwiązaniem problemu będzie retransmisja, jednak powoduje ona wprowadzenie do sieci dodatkowych pakietów. Podtrzymanie takiego stanu prowadzi do powstania **zatoru**. W przypadku transmisji internetowej przeciążenia zazwyczaj występują w routerach. Aby zapobiec ich występowaniu, protokoły transportowe na bieżąco monitorują stan sieci i reagują natychmiast po zarejestrowaniu pierwszych ich symptomów. Stosuje się dwa rozwiązania:

- Konfigurowanie systemów pośrednich (tj. routerów) w taki sposób, aby informowały nadawców o wystąpieniu przeciążenia.
- Wykorzystanie informacji o wzroście opóźnienia lub zwiększonej liczbie traconych pakietów do oszacowania skali przeciążenia.

Pierwsze rozwiązanie polega na implementowaniu w oprogramowaniu routerów mechanizmu wysyłającego specjalne komunikaty do stacji źródłowej po zarejestrowaniu przeciążenia lub ustawiającego specjalny bit w nagłówku każdego pakietu opóźnionego z powodu przeciążenia. W drugim przypadku komputer otrzymujący pakiet odsyła do nadawcy potwierdzenie odebrania danych wraz z informacją o zarejestrowanym przeciążeniu<sup>74</sup>.

Rozwiązanie polegające na szacowaniu skali przeciążenia na podstawie opóźnień i liczby utraconych pakietów wydaje się najodpowiedniejszym sposobem postępowania w przypadku internetu, ponieważ:

*Większość nowoczesnych sieci pracuje bezbłędnie. Ewentualne opóźnienia i utrata pakietów wynikają z przeciążeń, a nie z awarii sprzętowych.*

Właściwą reakcją na przeciążenie jest ograniczenie szybkości wysyłania pakietów. W protokołach wykorzystujących okno przesuwne efekt ten uzyskuje się przez zmniejszenie rozmiaru okna.

<sup>74</sup> Między wystąpieniem przeciążenia a dostarczeniem wiadomości do pierwotnego nadawcy może upływać sporo czasu.

## 26.7. Sztuka projektowania protokołu

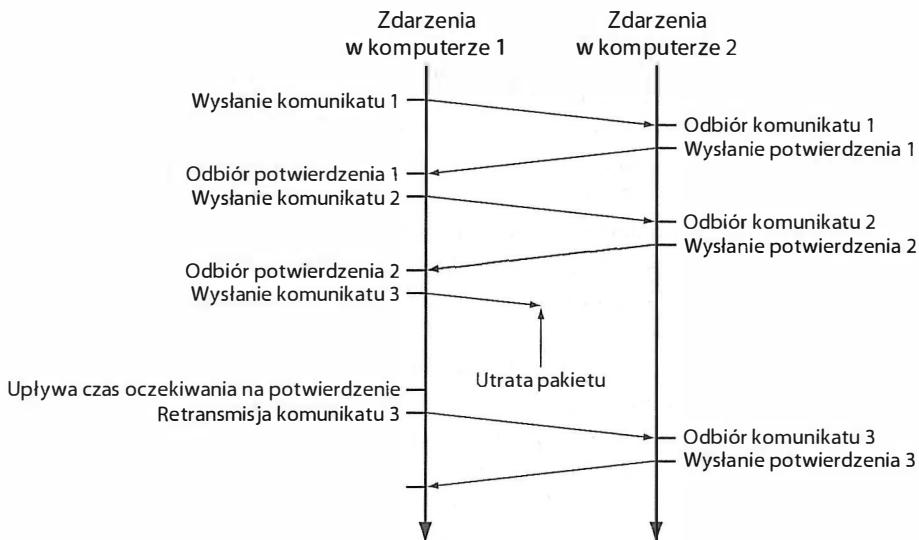
Choć zasady rozwiązywania większości problemów są doskonale znane, zaprojektowanie protokołu okazuje się niemałym problemem. Po pierwsze, efektywność komunikacji zależy od właściwego doboru wielu bardzo szczegółowych parametrów — niewielkie błędy projektowe mogą skutkować niewłaściwym działaniem mechanizmu, generowaniem niepotrzebnych pakietów lub wprowadzaniem zbędnych opóźnień. Na przykład w rozwiązaniach wykorzystujących technikę numerowania komunikatów każdy pakiet musi zawierać właściwy numer sekwencyjny. Pole numeru musi być więc dostatecznie duże, aby pozwalało na zapis wartości bez konieczności zbyt częstego przewijania licznika. Jednocześnie jednak musi mieć niezbyt duży rozmiar, aby nie marnować pasma. Po drugie, protokół może działać w nieoczekiwany sposób. Jako przykład rozważmy współdziałanie mechanizmów sterowania przepływem i unikania przeciążeń. Algorytm przesuwnego okna dąży do zajęcia jak największego pasma w celu zwiększenia przepływności danych. Mechanizm unikania przeciążeń wykonuje operacje o dokładnie odwrotnych skutkach — zmniejsza liczbę pakietów wprowadzanych do sieci w celu zabezpieczenia jej przed zatorem. Zachowanie równowagi między obydwooma rozwiązaniami bywa bardzo trudne. Agresywne sterowanie przepływem skutkuje przeciążeniem sieci, natomiast asekuracyjne wykorzystywanie mechanizmu kontroli przeciążeń prowadzi do nadmiernego obniżenia przepływności strumienia danych. Rozwiązania polegające na przełączaniu trybów pracy z agresywnego na zachowawczy wprowadza pewne oscylacje — stopień wykorzystania pasma powoli narasta do wystąpienia przeciążenia sieci, po czym zaczyna maleć aż do ustabilizowania sieci i ponownie zaczyna narastać.

Innym niebanalnym wyzwaniem jest restartowanie komputerów. Rozważmy przypadek, w którym dwie aplikacje ustanawiają między sobą połączenie, rozpoczynają przesyłanie danych, a następnie komputer odbierający informacje zostaje uruchomiony ponownie. Choć oprogramowanie protokołu w stacji odbiorczej nie prowadzi wymiany danych, moduł protokołu w komputerze nadawczym uznaje połączenie za poprawne. W przypadku błędnego zaprojektowania protokołu może wystąpić sytuacja, w której jednostka odbiorcza utworzy jednak połączenie i zacznie rejestrować dane od bieżącego etapu transmisji.

## 26.8. Obsługa utraconych pakietów w protokole TCP

Która z wymienionych wcześniej technik zapewnia niezawodną transmisję danych w protokole TCP? Odpowiedź jest złożona, ponieważ protokół TCP obejmuje wiele rozwiązań połączonych w nowatorski sposób. Nietrudno się domyślić, że obsługa utraconych pakietów polega na retransmisji. Dwukierunkowa wymiana danych pozwala obydwu stronom na uczestniczenie w tym procesie. Za każdym razem, gdy jeden z komputerów odbierze porcję danych, wysyła do drugiej jednostki **potwierdzenie**. Jednocześnie każdy nadawca, wysyłając pakiet, uruchamia zegar retransmisji, który powoduje ponowne przesłanie danych po upływie określonego czasu. Podstawowy mechanizm retransmisji w protokole TCP działa więc w sposób pokazany na rysunku 26.5.

Model retransmisji stosowany w protokole TCP jest podstawą sukcesu tego protokołu. Umożliwia bowiem wymianę danych niezależnie od przebiegu trasy w internecie.



Rysunek 26.5. Retransmisja po utracie pakietu

Z równym powodzeniem działa aplikacja przesyłająca pakiety przez łącze satelitarne do odbiorcy pracującego w innym kraju, jak i program dostarczający dane do komputera przyłączonego do sieci lokalnej w pokoju obok. Mechanizm TCP jest gotowy do retransmisji dowolnego utraconego komunikatu w dowolnej sieci. Nasuwa się jednak pytanie, jak długo komputer powinien oczekwać na dostarczenie potwierdzenia przed rozpoczęciem retransmisji. W sieci lokalnej potwierdzenia są dostarczane w czasie kilku milisekund. W komunikacji satelitarnej wiąże się to z opóźnieniem liczymy w setkach milisekund. Z jednej strony, oczekiwanie na potwierdzenie przez zbyt długi czas jest przyczyną nieaktywności sieci i obniżenia przepływności bitowej danych. Należyłoby się więc spodziewać, że w sieci lokalnej retransmisja nie będzie poprzedzona długą przerwą. Z drugiej strony, szybkie ponawianie transmisji nie sprawdza się w połączeniach satelitarnych, ponieważ zbędny ruch zwiększa wykorzystanie pasma i obniża przepływność użytecznych strumieni danych.

Projektanci protokołu TCP stanęli przed jeszcze trudniejszym wyzwaniem niż odróżnianie sieci lokalnych od zdalnych. Emitowanie zbitek datagramów prowadzi niekiedy do powstawania zatoru, a to z kolei powoduje gwałtowną zmianę opóźnień w przesyłaniu pakietów wzdułok określonej trasy. W praktyce całkowity czas wymagany do przesłania komunikatu i odebrania odpowiedzi wzrasta i maleje o rząd wielkości w czasie kilku milisekund.

*Opóźnienie w dostarczaniu danych do stacji docelowej i odbieraniu potwierdzeń zwrotnych zależy od bieżącego natężenia ruchu internetowego oraz odległości między stacjami. Jak wiadomo, protokół TCP umożliwia programom jednoczesną wymianę danych z wieloma komputerami zdalnymi. Dlatego biorąc pod uwagę wpływ natężenia ruchu na opóźnienia transmisyjne, musi monitorować wiele metryk opóźnienia, które ulegają gwałtownym zmianom.*

## 26.9. Adaptacyjne retransmisyjne

Przed opracowaniem mechanizmu TCP protokoły transportowe wykorzystywały ustalone wartości opóźnienia retransmisji. Twórcy protokołów lub administratorzy sieci dobierali odpowiednią wartość w zależności od spodziewanych parametrów sieci. Inżynierowie pracujący nad specyfikacją TXP uświadomili sobie, że wstępnie zdefiniowana wartość opóźnienia nie sprawdzi się w internecie. Podjęli więc decyzję o wdrożeniu **adaptacyjnego** algorytmu retransmisji. Dzięki temu moduł TCP monitoruje bieżące opóźnienie w ramach każdego połączenia i dostosowuje ustalenia zegara retransmisji do zmieniającej się charakterystyki sieci.

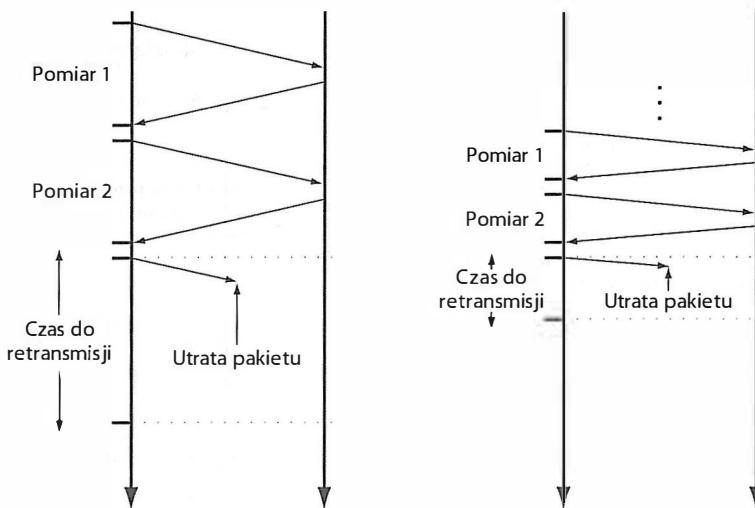
W jaki sposób opóźnienie jest monitorowane? Tak naprawdę, mechanizm TCP nie może dokładnie ustalić wartości opóźnień w poszczególnych obszarach internetu w dowolnie wybranym czasie. Szacuje jedynie **opóźnienie transmisji w obie strony** (ang. *round-trip delay*) na każdym aktywnym połączeniu. Zadanie sprawdza się do pomiaru czasu między wysłaniem komunikatu i otrzymaniem odpowiedzi. Za każdym razem, gdy komputer wysyła wiadomość, na którą spodziewa się odpowiedzi, moduł TCP rejestruje bieżącą wartość czasu. Następnie, w chwili odebrania odpowiedzi, odejmuje czas jej nadęcia od czasu zapisanego wcześniej i otrzymuje opóźnienie w transmisji w obydwu kierunkach. Dzięki temu, że pakiety danych i odpowiedzi są generowane seriami, oprogramowanie TCP może na podstawie funkcji statystycznych wyznaczyć średnią ważoną opóźnienia oraz wariancję. Kombinacja liniowa oszacowanej średniej oraz wariancji jest następnie wykorzystywana do określenia czasu przerwy przed retransmisją.

Testy dowodzą skuteczności mechanizmów adaptacyjnych retransmisji. Wykorzystanie wariancji pozwala protokołowi TCP na szybką reakcję w przypadkach wzrostu opóźnienia w następstwie transmisji zbitki pakietów. Z kolei użycie średniej ważonej gwarantuje resetowanie zegara retransmisji po obniżeniu się wartości opóźnienia w czasie emisji zbitki pakietów. Gdy opóźnienie pozostaje na stałym poziomie, zegar retransmisji jest ustalony na czas nieznacznie dłuższy niż średnie opóźnienie transmisji w dwie strony. W chwili zarejestrowania wahania wartości opóźnienia mechanizm TCP zwiększa czas retransmisji do wartości istotnie większej niż średnia, aby uwzględnić gwałtowne wzrosty.

## 26.10. Porównanie czasów retransmisji

Aby zrozumieć, w jaki sposób algorytm adaptacyjnej retransmisji umożliwia maksymalizowanie przepustowości połączzeń, przeanalizujmy przypadek utraty pakietów w dwóch połączeniach o różnych wartościach opóźnienia. Ruch w każdym z dwóch połączzeń został przedstawiony graficznie na rysunku 26.6.

Jak nietrudno zauważać, oprogramowanie TCP wyznacza czas oczekiwania na potwierdzenie jako nieco dłuższy niż średnie opóźnienie w transmisji w dwie strony. Jeśli opóźnienie ma dużą wartość, czas do retransmisji również jest długi. Jeśli opóźnienie jest niewielkie, retransmisja następuje szybciej. Celem jest obliczenie takiego czasu oczekiwania, aby można było wykryć utratę pakietu, ale bez konieczności przerywania transmisji na czas dłuższy, niż jest to niezbędne.



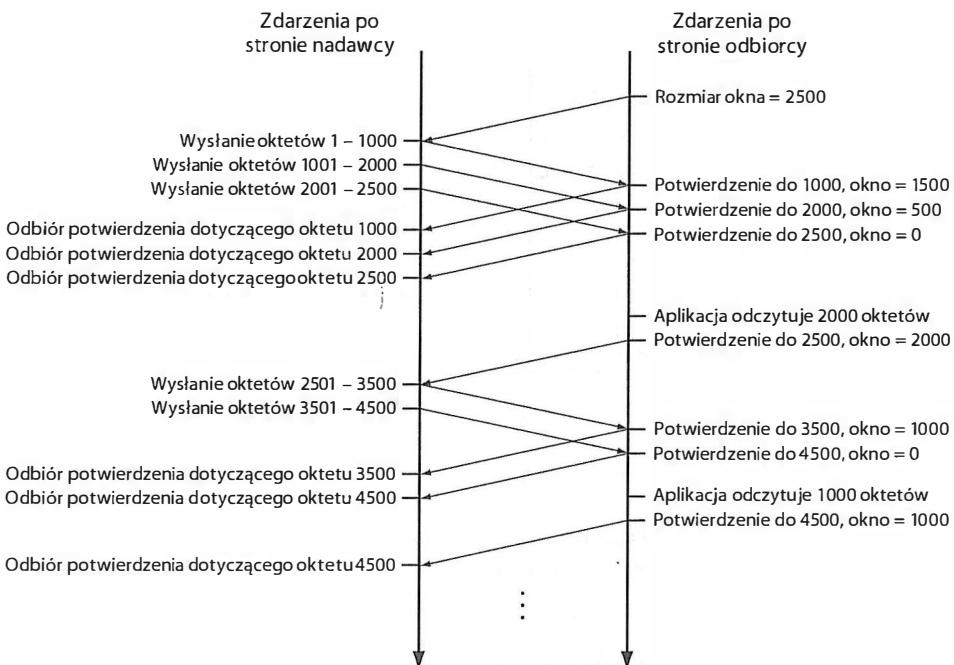
Rysunek 26.6. Czas oczekiwania na potwierdzenie  
i retransmisja w dwóch połączeniach TCP o różnych opóźnieniach

## 26.11. Bufory, sterowanie przepływem i okna

Sterowanie przepływem danych w protokole TCP bazuje na mechanizmie **okna**. Jego rozmiar jest definiowany w bajtach (a nie w pakietach, jak w przedstawionym wcześniej opisie). Po ustanowieniu połączenia obydwie strony rezerwują pamięć potrzebną do przechowywania nadchodzących danych, a następnie wysyłają informacje o rozmiarze bufora do stacji zdalnej. W chwili dostarczenia danych stacja odbiorcza wysyła potwierdzenie, w którym uwzględnia rozmiar wolnego obszaru pamięci pakietów. W specyfikacji TCP ilość wolnej pamięci bufora jest właśnie określana mianem **okna**. Z kolei powiadomienie o rozmiarze okna nazywa się **rozgłoszeniem okna** (ang. *window advertisement*). Odbiorca wysyła rozgłoszenie okna w każdym potwierdzeniu.

Jeśli aplikacja odbiorcza może przyjmować dane z szybkością ich napływanego, komputer odbiorczy wysyła potwierdzenia z niezerowym rozmiarem okna. Jeśli jednak strona nadawcza generuje pakiety szybciej, niż jednostka zdalna może je odbierać (na przykład z uwagi na wydajniejszy procesor), nadchodzące dane zapełniają bufor odbiorczy i mogą doprowadzić do wysłania informacji o **oknie zerowym**. Po otrzymaniu powiadomienia o zerowym rozmiarze okna nadawca musi wstrzymać nadawanie aż do momentu otrzymania pozytywnej aktualizacji okna. Przepływ informacji o rozmiarze okna został pokazany na rysunku 26.7.

W powyższym przykładzie nadawca wykorzystuje segmenty o maksymalnym rozmiarze 1000 bajtów. Transfer rozpoczyna się od przesłania przez odbiorcę informacji o rozmiarze okna, który wynosi 2500 bajtów. Nadawca natychmiast wysyła trzy segmenty — dwa o rozmiarze 1000 bajtów i jeden składający się z 500 bajtów. W chwili odebrania danych stacja odbiorcza generuje potwierdzenie zawierające informację o zmniejszeniu rozmiaru okna o wartość wynikającą z ilości otrzymanych danych.



**Rysunek 26.7.** Sekwencja komunikatów sterujących przepływem danych TCP o maksymalnym rozmiarze segmentu wynoszącym 1000 bajtów

Trzy pierwsze segmenty zajęły bufor odbiorcy szybciej, niż aplikacja mogła je odczytać. Z tego powodu rozmiar okna został zredukowany do zera, a nadawca musiał wstrzymać transmisję. Po odczytaniu 2000 bajtów z bufora jednostka odbiorcza generuje dodatkowe potwierdzenie, które jest powiadomieniem o zmianie rozmiaru okna na 2000 bajtów. Rozmiar okna jest wyznaczany w odniesieniu do danych, które zostały wcześniej odebrane. Odbiorca informuje więc nadawcę, że może przyjąć 2000 bajtów następujących po 2500 bajtach, które odebrał wcześniej. Reakcja nadawcy polega na przekazaniu kolejnych dwóch segmentów. Odebraniu każdego segmentu towarzyszy wygenerowanie przez odbiorcę potwierdzenia z rozmiarem okna mniejszym o 1000 bajtów (odpowiadającym ilości otrzymanych danych).

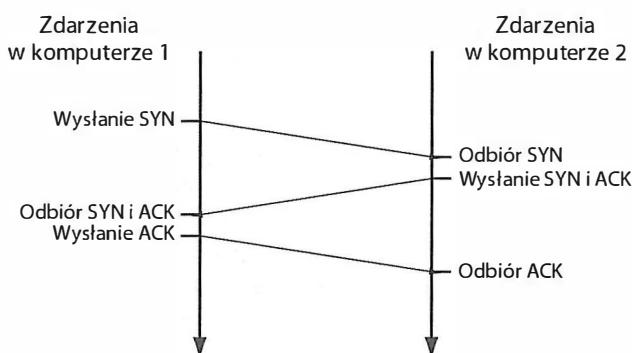
Po raz kolejny rozmiar okna osiąga wartość zerową, co powoduje wstrzymanie transmisji. Po pewnym czasie aplikacja odbiorcza odczyta dane, a oprogramowanie TCP wyśle potwierdzenie z niezerową wartością rozmiaru okna. Jeśli nadawca ma więcej segmentów do wysłania, wznowi transmisję.

## 26.12. Trójetapowe porozumienie

W celu zagwarantowania prawidłowego ustanawiania i rozłączania połączenia w specyfikacji TCP opisano mechanizm **trójetapowego porozumienia** (ang. *3-way handshake*). W czasie ustanawiania połączenia za pomocą wspomnianego mechanizmu obydwie strony wysyłają komunikat sterujący, który zawiera informacje o rozmiarze bufora (do

sterowania przepływem) oraz numer sekwencyjny obowiązujący w danym komputerze. Naukowcy dowiedli, że trójetapowa wymiana komunikatów jest konieczna i wystarczająca do zawarcia jednoznacznego porozumienia między stronami, niezależnie od utraty pakietów, ich opóźniania, duplikowania lub powtarzania z opóźnieniem<sup>75</sup>. Ponadto mechanizm ten daje gwarancję, że moduły TCP nie ustanowią ani nie zerwą połączenia, dopóki obydwie strony nie wyrażą na to zgody.

Komunikat sterujący, który odpowiada za utworzenie połączenia, jest nazywany **segmentem synchronizacyjnym (segmentem SYN [ang. synchronization segment; SYN segment])**. Z kolei komunikat odpowiedzialny za rozłączenie połączenia jest określany jako **segment kończący (segment FIN [ang. finish segment; FIN segment])**. Trójetapowa wymiana komunikatów służąca nawiązaniu połączenia została pokazana na rysunku 26.8.

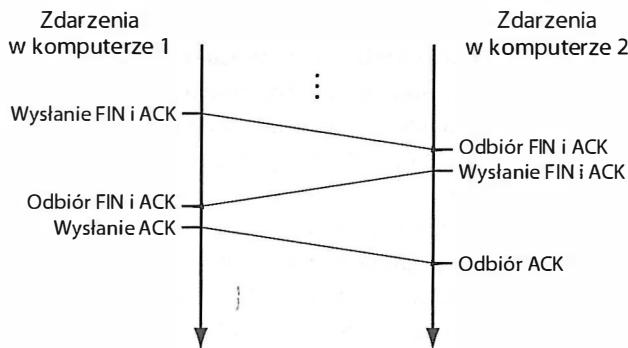


Rysunek 26.8. Trójetapowe porozumienie służące do ustanowienia połączenia

Jedną z ważniejszych operacji towarzyszących nawiązywaniu połączenia jest wymiana numerów sekwencyjnych. Zgodnie ze standardem TCP każda stacja końcowa musi wygenerować 32-bitową wartość, która zostanie wykorzystana jako początkowy numer sekwencyjny w transmisji danych. Próba ustanowienia nowego połączenia po ponownym uruchomieniu komputera wiąże się z koniecznością wygenerowania nowej wartości numeru sekwencyjnego. Ponieważ prawdopodobieństwo wylosowania liczby odpowiadającej numerowi z wcześniejszego połączenia jest bardzo małe, praktycznie nie istnieje ryzyko wystąpienia błędu powtórzeniowego. Jeśli dwie aplikacje korzystające z protokołu TCP przerwą połączenie, a następnie nawiążą nowe, numery sekwencyjne obowiązujące w nowym połączeniu będą się różniły od wykorzystywanych w poprzednim, a to umożliwi modułom TCP odrzucanie opóźnionych pakietów.

Rozłączanie połączenia za pomocą tego samego mechanizmu wymaga użycia segmentów FIN. Wraz z segmentem FIN na drugą stronę dostarczane jest potwierdzenie, które gwarantuje, że przed zakończeniem połączenia wszystkie dane zostały odebrane. Proces wymiany komunikatów pokazano na rysunku 26.9.

<sup>75</sup> Pakiety z trójetapowego porozumienia mogą być retransmitowane w taki sam sposób, jak klasyczne pakiety TCP.



Rysunek 26.9. Trójetapowa wymiana danych podczas rozłączania połączenia

## 26.13. Kontrola przeciążenia

Jednym z najciekawszych elementów protokołu TCP jest mechanizm **kontroli przeciążenia**. Zgodnie z wcześniejszymi informacjami opóźnienia i utrata pakietów wynikają znacznie częściej z przeciążeń niż z awarii sprzętowych, a funkcja retransmisji dodatkowo wzmaga ten problem, wprowadzając kolejne kopie pakietów. Aby uniknąć całkowitego zatoru, w protokole TCP analizuje się wartości opóźnienia i szacuje na ich podstawie skalę przeciążenia. Reakcja na problem polega natomiast na ograniczeniu szybkości wysyłania pakietów.

Ograniczenie szybkości transmisji nie jest takim prostym zadaniem, jak by się mogło wydawać, ponieważ oprogramowanie TCP nie oblicza bieżącej przepływności. Algorytm nadawczy uzależnia transmisję od rozmiaru bufora — odbiorca dostarcza informację o rozmiarze okna, a nadawca dąży do zapełnienia bufora odbiorczego jeszcze przed otrzymaniem segmentu ACK. Sterowanie przepływnością jest jednak możliwe dzięki ograniczeniom nałożonym przez specyfikację TCP na proces wyznaczania rozmiaru okna oraz temu, że chwilowe zmniejszenie rozmiaru okna powoduje efektywne ograniczenie szybkości transmisji danych.

*Protokół transportowy powinien zmniejszać przepływność danych w chwili wystąpienia przeciążenia. Mechanizm TCP realizuje to zadanie dzięki zmiennemu rozmiarowi okna transmisyjnego — chwilowe zmniejszenie rozmiaru okna powoduje spowolnienie transmisji. W szczególnych przypadkach (związanego z utratą pakietów) rozmiar okna może zostać zmniejszony do połowy wcześniejszej wartości.*

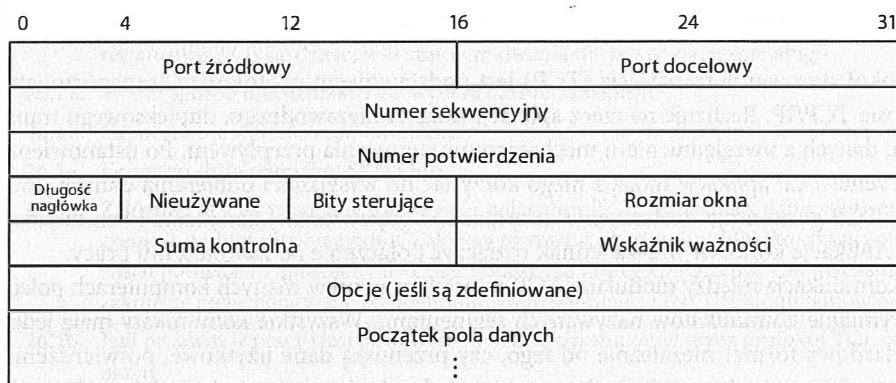
W chwili nawiązania połączenia lub utraty pakietu protokół TCP uruchamia specjalny algorytm kontroli przeciążenia. W wyniku jego działania stacja nadawcza wysyła pojedyncze komunikaty z danymi, a nie grupy komunikatów o rozmiarze odpowiadającym wielkości bufora odbiorczego (rozmiarowi okna odbiorczego). Jeśli pakiet nie zostanie utracony (nadawca otrzyma potwierdzenie dostarczenia pakietu), ilość wysyłanych danych jest podwajana (wysyłane są dwa pakiety). Odebranie dwóch potwierdzeń oznacza, że

nadawca może wyemitować cztery komunikaty itd. Proces wykładniczego zwiększenia liczby pakietów jest kontynuowany do momentu, gdy ilość wysyłanych danych odpowiada połowie rozmiaru okna odbiorczego. Od tej chwili nadawca zwiększa rozmiar okna nadawczego w sposób liniowy, aż do zarejestrowania przeciążenia. Rozwiążanie to jest nazywane algorytmem **powolnego startu**.

Mechanizm kontroli przeciążenia doskonale sprawdza się w sytuacjach zwiększonego natężenia ruchu, a dzięki szybkiej reakcji protokołu możliwe jest złagodzenie skutków przeciążenia. W czasie przeciążenia sieci oprogramowanie TCP unika retransmitowania danych. Ponadto, jeśli wszystkie stacje pracują zgodnie ze standardem, wystąpienie opisywanej sytuacji powoduje wstrzymanie transmisji we wszystkich stacjach, co pozwala na wyeliminowanie ryzyka zatoru.

## 26.14. Format segmentu TCP

Zgodnie ze standardem TCP wszystkie rodzaje wiadomości mają jeden format (zarówno te, które przenoszą dane, jak i potwierdzenia oraz powiadomienia z trójetapowego porozumienia służące do nawiązania bądź przerwania połączenia). W terminologii związanej z protokołem TCP komunikat nazywa się **segmentem**. Format segmentu TCP został przedstawiony na rysunku 26.10.



Rysunek 26.10. Format segmentu TCP  
wykorzystywany do transmisji danych i komunikatów sterujących

Analizując format segmentu, trzeba pamiętać, że na połączenie TCP składają się dwa strumienie danych (po jednym w każdym kierunku). Jeśli dwie aplikacje równocześnie wysyłają dane, moduł TCP może formować segmenty, które będą zawierały dane wychodzące, potwierdzenia danych przychodzących oraz informacje o rozmiarze okna (określające rozmiar wolnej przestrzeni w buforze odbiorczym). Niektóre pola nagłówka odnoszą się więc do strumienia kierowanego w stronę odbiorcy, a inne opisują strumień danych dostarczanych do określonej stacji.

Jednostka wysyłająca segment ustawia pola *numer potwierdzenia* oraz *rozmiar okna* zależnie od ilości odebranych danych. Wartość *numer potwierdzenia* odpowiada numerowi

sekwencyjnemu w następnym segmencie, którego spodziewa się odbiorca. Natomiast *rozmiar okna* wskazuje ilość wolnej pamięci w buforze odbiorczym komputera. Wartość numeru potwierdzenia zawsze odpowiada pierwszej pozycji w brakujących danych. Jeśli segment zostanie dostarczony poza kolejnością, moduł TCP będzie wielokrotnie generował potwierdzenia o takiej samej treści, aż do otrzymania brakującego fragmentu. Pole *numer sekwencyjny* odnosi się do danych wychodzących. Odpowiada numerowi pierwszego bajtu danych przenoszonych przez segment. Odbiorca wykorzystuje numer sekwencyjny do ułożenia w odpowiednim porządku segmentów dostarczanych poza kolejnością oraz do wyliczenia numeru potwierdzenia. Pole *port docelowy* identyfikuje aplikację, która powinna otrzymać dane segmentu. Natomiast pole *port źródłowy* wskazuje program, który wysłał segment. Pole *suma kontrolna* zawiera wartość kontrolną obejmującą nagłówek segmentu oraz dane.

Odnoszenie numerów sekwencyjnych i numerów potwierdzenia warto zapamiętać, że:

*Pole numer sekwencyjny odpowiada numerowi pierwszego bajtu przenoszonego w polu danych segmentu w kierunku do odbiorcy. Pole numer potwierdzenia zawiera numer sekwencyjny pierwszego bajtu spośród brakujących po stronie odbiorczej.*

## 26.15. Podsumowanie

Protokół sterowania transmisją (TCP) jest podstawowym protokołem transportowym w stosie TCP/IP. Realizuje na rzecz aplikacji zadania niezawodnego, dupleksowego transportu danych z uwzględnieniem mechanizmów sterowania przepływem. Po ustanowieniu połączenia TCP aplikacje mogą z niego korzystać do wysyłania i odbierania danych. Protokół TCP gwarantuje dostarczenie informacji w odpowiedniej kolejności i bez duplikatów. Aplikacje końcowe muszą jednak rozłączyć połączenie po zakończeniu pracy.

Komunikacja między modułami TCP zainstalowanymi w różnych komputerach polega na wymianie komunikatów nazywanych segmentami. Wszystkie komunikaty mają jeden standardowy format niezależnie od tego, czy przenoszą dane użytkowe, potwierdzenia, informacje o zmianie rozmiaru okna, czy powiadomienia o ustanowieniu lub rozłączeniu połączenia. Każdy segment TCP jest przenoszony przez datagram IP.

Ogólnie rzecz ujmując, protokoły transportowe wykorzystują różne mechanizmy gwarantujące niezawodność realizowanych przez nie usług. W rozwiązaniu TCP zastosowano bardzo skomplikowane połączenie różnych technik, które ostatecznie okazało się niezwykle udanym przedsięwzięciem. Protokół TCP weryfikuje poprawność danych na podstawie zapisanej w nagłówku wartości sumy kontrolnej, a także retransmituje wszystkie utracone komunikaty. Sam mechanizm retransmisji ma charakter adaptacyjny, dzięki czemu doskonale sprawdza się w internecie, w którym opóźnienia transmisyjne nieustannie się zmieniają. Protokół TCP zawiera mechanizmy pomiaru bieżącego opóźnienia, które realizują zadanie niezależnie w każdym połączeniu. Wyznaczona na podstawie ich działań średnia ważona opóźnienia służy do określenia czasu oczekiwania na potwierdzenie.

## ZADANIA

- 26.1. Zaprojektuj protokół, który w sposób niezawodny będzie wymieniał informacje o nawiązaniu połączenia między dwoma programami. Przyjmij założenie, że wysyłane komunikaty mogą być tracone, duplikowane, opóźniane oraz dostarczane w niewłaściwej kolejności. Przekaż projekt innej osobie i poproś ją o znalezienie takiej sekwencji pakietów utraconych, powielonych lub opóźnionych, która spowoduje błędne działanie protokołu.
- 26.2. Wymień cechy protokołu TCP.
- 26.3. Które warstwy stosu protokołów są implementowane w routerach, a które w komputerach?
- 26.4. Jaki jest główny problem, który musi zostać rozwiązany, aby protokół transportowy mógł przekazywać informacje w sposób niezawodny?
- 26.5. Jaki mechanizm jest stosowany w protokołach transportowych?
- 26.6. Ile pakietów można przesyłać przed odebraniem potwierdzenia, jeśli rozmiar okna przesuwnego wynosi N?
- 26.7. Dlaczego zastosowanie protokołu start-stop w łączu satelitarnym typu GEO o przepustowości 2 Mb/s skutkuje niezwykle małą przepływnością bitową?
- 26.8. Uzupełnij wykres z rysunku 26.3 w taki sposób, aby prezentował transmisję szesnastu kolejnych pakietów.
- 26.9. Jaka jest główna przyczyna opóźnienia i utraty pakietów w internecie?
- 26.10. W jaki sposób protokół TCP obsługuje przypadki utraty pakietów?
- 26.11. Co się dzieje z przepustowością, jeśli protokół transportowy wprowadza zbyt długi czas retransmisji? Co się dzieje, jeśli czas oczekiwania nie jest dostatecznie długi?
- 26.12. W jaki sposób mechanizm TCP wylicza czas retransmisji?
- 26.13. Czym steruje wartość rozmiaru okna?
- 26.14. Do czego służą segmenty SYN i FIN?
- 26.15. Założmy, że dwa programy ustanawiają połączenie TCP, wymieniają dane, rozłączają połączenie, a następnie tworzą nowe. Założmy również, że komunikat FIN (kończący połączenie) został powielony i opóźniony do czasu zestawienia nowego połączenia. Czy mechanizm TCP zakończy nowe połączenie po odebraniu kopii segmentu FIN? Uzasadnij odpowiedź.
- 26.16. Jaki problem w pracy sieci powoduje chwilowe ograniczenie przez protokół TCP rozmiaru okna?
- 26.17. Napisz program komputerowy, który wyodrębnii i wyświetli pola nagłówka segmentu.
- 26.18. Czy suma kontrolna TCP jest niezbędna? Czy o poprawności transmisji nie może decydować suma kontrolna datagramu IP? Uzasadnij odpowiedź.

# Zawartość rozdziału

- 27.1. Wprowadzenie 469
- 27.2. Routing statyczny a routing dynamiczny 469
- 27.3. Routing statyczny w komputerze i trasa domyślna 470
- 27.4. Routing dynamiczny i routery 471
- 27.5. Routing w globalnym internecie 472
- 27.6. Idea systemu autonomicznego 473
- 27.7. Dwa rodzaje protokołów routingu internetowego 473
- 27.8. Trasy i transport danych 476
- 27.9. Protokół bram granicznych (BGP) 476
- 27.10. Protokół informowania o trasach (RIP) 478
- 27.11. Format pakietu RIP 479
- 27.12. Otwarty protokół wyznaczania najkrótszych tras (OSPF) 479
- 27.13. Przykład grafu OSPF 481
- 27.14. Obszary OSPF 482
- 27.15. Protokół systemów pośrednich (IS-IS) 482
- 27.16. Routing w multiemisji 483
- 27.17. Podsumowanie 487

# *Routing internetowy i protokoły routingu*

## 27.1. Wprowadzenie

W poprzednich rozdziałach zostały opisane podstawowe rozwiązania, które zapewniają przesyłanie datagramów oraz wybieranie w tablicy routingu kolejnych węzłów na trasie datagramu IP. Tematem tego rozdziału jest inna niezwykle istotna koncepcja związana z funkcjonowaniem sieci internetowej — rozgłaszenie informacji o trasach pozwalające na automatyczne tworzenie i aktualizowanie tablic routingu. Opisano tutaj mechanizm budowania tablic routingu oraz zasady programowego aktualizowania ich wpisów.

Przedstawione w tym rozdziale zagadnienia dotyczą propagowania informacji o routingu w internecie. Omówienie zostało uzupełnione prezentacją kilku powszechnie stosowanych rozwiązań oraz wyjaśnieniem różnic między protokołami routingu wewnętrznego i zewnętrznego.

## 27.2. Routing statyczny a routing dynamiczny

Zagadnienia związane z routingu IP należą do jednej z dwóch kategorii:

- routing statyczny,
- routing dynamiczny.

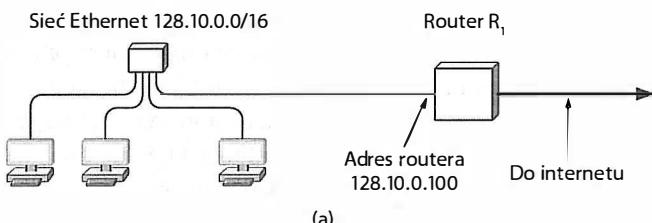
Określenie **routing statyczny** odnosi się do systemów, w których tablice routingu są budowane wraz ze startem systemu i nie podlegają żadnym zmianom, chyba że administrator zdecyduje się osobiście wprowadzić korekty. Z kolei **routing dynamiczny** jest rozwiązaniem, w którym uruchomione oprogramowanie nieustannie aktualizuje tablice routingu,

zapewniając wybór najlepszej trasy dla każdego pakietu. Oprogramowanie odpowiedzialne za routing musi się komunikować z innymi systemami w celu pozyskania informacji o trasach do poszczególnych sieci docelowych oraz wykrycia ewentualnych zmian w sieci wymagających modyfikacji tras. Mechanizmy routingu dynamicznego rozpoczynają swoje działanie tak samo jak procesy routingu statycznego, czyli od załadowania początkowej tablicy routingu.

### 27.3. Routing statyczny w komputerze i trasa domyślna

Zasady routingu statycznego nie są skomplikowane, a ich wdrożenie w systemie operacyjnym nie wymaga szczególnego oprogramowania. Rozwiążanie to nie wiąże się również z zajmowaniem pasma transmisyjnego ani z wykorzystaniem procesora w celu rozpoznanienia informacji o trasach. Wadą routingu statycznego jest jednak brak elastyczności konfiguracji — brak reakcji na awarie sieciowe i zmiany topologii.

W jakich przypadkach routing statyczny jest wykorzystywany? Przede wszystkim w komputerach, które są przyłączone do sieci za pomocą jednego interfejsu sieciowego, oraz gdy sieć jest połączona z internetem za pośrednictwem jednego routera. Jako przykład takiego rozwiązania warto przeanalizować rysunek 27.1. Przedstawiono na nim cztery komputery przyłączone do sieci Ethernet, która sama jest połączona z internetem za pośrednictwem routera  $R_1$ .



Sieć	Maska	Następny skok
128.10.0.0	255.255.0.0	Połączenie bezpośrednie
Domyślna	0.0.0.0	128.10.0.100

(b)

Rysunek 27.1. Typowa sieć przyłączona do internetu (a)  
oraz statyczna tablica routingu każdego z komputerów (b)

Jak wynika z rysunku, statyczna tablica routingu z dwoma wpisami wystarcza komputerowi do pracy w sieci. Pierwszy z wierszy zawiera adres sieci, do której jednostka jest podłączona. Natomiast drugi wpis informuje o tym, że **trasa domyślna** (prowadzi do wszystkich pozostałych sieci docelowych) prowadzi w pierwszym etapie do routera  $R_1$ . Gdy aplikacja musi dostarczyć datagram do jednostki z lokalnej sieci (na przykład do dru-

karki), z pierwszego wiersza uzyska informację, że można ten datagram przekazać bezpośrednio do odbiorcy. Jeśli stacją docelową będzie jakakolwiek jednostka poza siecią lokalną, na podstawie drugiego wpisu komputer dostarczy pakiet do routera  $R_1$ .

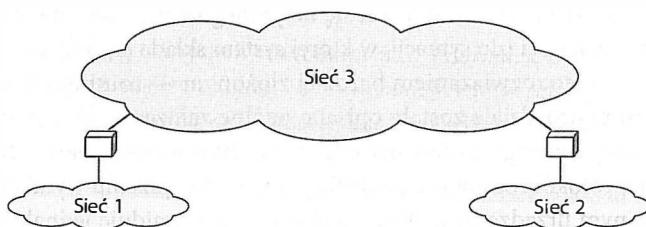
Najważniejszy wniosek to:

*Większość stacji internetowych korzysta z routingu statycznego. Tablica komputera składa się z dwóch wpisów: dotyczącego sieci lokalnej oraz opisującego trasę domyślną, na podstawie którego cały ruch adresowany do jednostek zdalnych jest kierowany do wskazanego routera.*

## 27.4. Routing dynamiczny i routery

Czy routery internetowe mogą korzystać z routingu statycznego w taki sam sposób jak zwykłe komputery? Choć nic nie stoi na przeszkodzie, żeby również na routeraх definiować trasy statycznie, większość urządzeń tego typu używa mechanizmów routingu dynamicznego. Szczególny przypadek, w którym routing statyczny wydaje się wystarczający, został przedstawiony na rysunku 27.1. Przedstawiona konfiguracja jest typowa dla małego przedsiębiorstwa, które jest klientem dostawcy usług internetowych. Cały ruch wychodzący z sieci firmowej jest kierowany przez router  $R_1$  do sieci dostawcy usług (na przykład za pośrednictwem łącza DSL). Ponieważ trasa nigdy nie ulega zmianie, tablica routera  $R_1$  może zawierać wpisy statyczne. Ponadto tablica routingu w urządzeniu  $R_1$  może zawierać wpis o trasie domyślnej, analogicznie jak w przypadku komputera.

Poza kilkoma wyjątkowymi konfiguracjami, routing statyczny i trasy domyślne nie są wystarczające do efektywnej pracy routerów. Jedynymi obszarami ich zastosowania są wówczas sieci o strukturze zbliżonej do opisanej powyżej. W przypadku połączenia sieci dwóch dostawców usług internetowych wymiana informacji o routingu musi być realizowana automatycznie. Przedstawiony na rysunku 27.2 przykład trzech sieci połączonych dwoma routery najlepiej oddaje istotę problemu.



Rysunek 27.2. Przykład architektury sieci, w której wymagany jest routing dynamiczny

Każdy router przechowuje informacje o sieciach bezpośrednio do niego przyłączonych. Router  $R_1$  dysponuje informacjami o sieciach 1 i 3, a router  $R_2$  o sieciach 2 i 3. Jednak router  $R_1$  nie wie o istnieniu sieci 2, a router  $R_2$  o istnieniu sieci 1. Nie są bowiem przyłączone do tych sieci. W tak prostym przypadku routing statyczny byłby wystarczający. Niemniej brak skalalności sprawia, że nie nadaje się on do stosowania w konfiguracji

obejmującej tysiące sieci. Za każdym razem, gdy dostawca usług internetowych wydziela nową sieć, informacje na jej temat muszą zostać rozpropagowane w internecie. Również awarie sieci i przeciążenia wymagają zmian w innych routerach. Wykonywanie tych zadań przez ludzi zajmuje zbyt wiele czasu. Dlatego w celu zapewnienia wszystkim routerom informacji o sposobie dostarczania danych do każdej sieci zdalnej uruchamia się w nich oprogramowanie, które wykorzystuje protokół routingu do wymiany informacji z innymi urządzeniami. Umożliwia ono odbieranie informacji o zmianie tras w innych węzłach i automatyczne wprowadzenie modyfikacji w tablicy routingu routera, który takie powiadomienie otrzyma. Ponieważ routery nieustannie wymieniają między sobą informacje o stanie sieci, tablice routingu każdego z nich są na bieżąco uaktualniane.

W przykładzie przedstawionym na rysunku 27.2 routery  $R_1$  i  $R_2$  mogą wymieniać informacje o routingu za pośrednictwem sieci 3. W wyniku tych działań oprogramowanie zarządzające tablicą routingu węzła  $R_2$  mogłoby wprowadzić wpis o trasie do sieci 1. Natomiast analogiczny moduł routera  $R_1$  dodałby wiersz opisujący trasę do sieci 2. Ewentualna awaria routera  $R_2$  zostałaby wykryta przez mechanizm zarządzania routinem urządzenia  $R_1$ , co z kolei spowodowałoby usunięcie trasy z tablicy routingu. Po ponownym uruchomieniu routera  $R_2$  oprogramowanie routera  $R_1$  zarejestrowałoby dostępność sieci 2 i przywróciło wpis o trasie do tej sieci.

Podsumowując:

*W każdym routerze działa oprogramowanie, które zbiera informacje o sieciach zdalnych oraz węzłach, które pośredniczą w przesyłaniu pakietów do tych sieci. Powiadamia również inne routery o sieciach zdalnych, które są znane danemu routrowi. Informacje nadchodzące od innych węzłów są wykorzystywane do ciągłego aktualizowania lokalnych tablic routingu.*

## 27.5. Routing w globalnym internecie

Przedstawiony wcześniej przykład odnosił się do jednego z najmniej skomplikowanych wariantów konfiguracji sieci (do sytuacji, w której system składa się jedynie z kilku routrów). Zajmijmy się więc rozwiązaniem bardziej złożonym — routingiem w globalnym internecie. W tym podrozdziale zostały opisane ogólne założenia takiego mechanizmu. Konkretnie protokoły routingu zostały natomiast przedstawione w kolejnych punktach.

Jak wiadomo, protokół routingu umożliwia jednemu urządzeniu wymianę informacji na temat tras z innym urządzeniem. Taki mechanizm nie znajduje jednak zastosowania w ogólnoswiatowym internecie. Gdyby każdy router internetowy wymieniał informacje ze wszystkimi pozostałymi urządzeniami działającymi w globalnej sieci, związany z tą działalnością ruch zdominowałby transmisję w łączach rdzeniowych. Aby ograniczyć ilość przesyłanych danych, wprowadzono w routingu pewną formę hierarchii. Routery sieciowe zostały podzielone na grupy. Informacje o połączeniach między sieciami są wymieniane między routerami jednej grupy. W każdej grupie funkcjonuje również specjalny router, który uogólnia zebrane dane i dostarcza je do routerów pracujących w innych grupach.

Jak duża jest pojedyncza grupa? Jaki protokół jest wykorzystywany w danej grupie? Jak zapisywane są informacje na temat tras? Jaki protokół umożliwia wymianę informacji między grupami? Twórcy internetu nie określili rozmiaru grupy. Nie określili również formatu danych ani protokołu, który powinien być wykorzystywany. Celowo zostawili pewną dowolność organizacjom, które zarządzają strukturami poszczególnych sieci. Mogą one samodzielnie określać minimalny i maksymalny rozmiar grupy oraz stosować w nich dowolne protokoły routingu.

## 27.6. Idea systemu autonomicznego

Wspomniane grupy routerów są określane jako **systemy autonomiczne** (AS — ang. *Autonomous System*). Systemy autonomiczne należy postrzegać jako ciągłe zbiory sieci i routerów, pozostające pod kontrolą jednej organizacji nadzorczej. Nie zdefiniowano jednak znaczenia terminu **organizacja nadzorcza**, więc jest on dostatecznie ogólny, by mieć zastosowanie w wielu różnych przypadkach. Na przykład systemem autonomicznym może być sieć dostawcy usług internetowych, sieć przedsiębiorstwa lub uniwersytetu. Korporacje o wielu ośrodkach mogą wyznaczyć jeden system autonomiczny lub oddzielnny system w każdym ośrodku. Większość dostawców usług internetowych utrzymuje jeden system autonomiczny, choć część większych firm tego typu dzieli swoje sieci na kilka systemów autonomicznych.

Ustalenie rozmiaru systemu autonomicznego zależy od uwarunkowań ekonomicznych, technicznych i administracyjnych. Na przykład wielonarodowa organizacja może uznać, że mniej kosztowne okaże się utrzymywanie wielu systemów autonomicznych, z których każdy będzie wykorzystywał łącze zapewnione przez dostawcę usług internetowych działającego w danym kraju. Istotny wpływ na rozmiar systemu ma również zastosowany protokół routingu — część protokołów generuje bardzo duży ruch, jeśli obejmuje swoim działaniem większą liczbę routerów (natężenie ruchu związanego z przesyłaniem informacji o trasach może rosnąć proporcjonalnie do kwadratu liczby routerów).

Podsumowując:

*Internet jest podzielony na systemy autonomiczne. Informacje o trasach wewnętrznych są wymieniane między urządzeniami wchodząymi w skład jednego systemu autonomicznego. Router łączące poszczególne systemy autonomiczne wymieniają między sobą uogólnione informacje o sieciach.*

## 27.7. Dwa rodzaje protokołów routingu internetowego

Znając pojęcie systemu autonomicznego, możemy bardziej szczegółowo przeanalizować zasady działania routingu internetowego. Każdy protokół routingu należy do jednej z dwóch kategorii:

- protokoły routingu wewnętrznego (IGP — ang. *Interior Gateway Protocols*);
- protokoły routingu zewnętrznego (EGP — ang. *Exterior Gateway Protocols*).

Przykłady konkretnych protokołów z każdej kategorii zostały przedstawione po omówieniu samych kategorii.

### 27.7.1. Protokoły routingu wewnętrznego

**Protokoły routingu wewnętrznego** (IGP) są wykorzystywane do wymiany danych na temat tras między routerami wchodzącymi w skład systemu autonomicznego. Istnieje kilka rodzajów mechanizmów IGP. Administrator danego systemu ma pełną dowolność w wyborze jednego z nich. Instalacja i konfiguracja większości protokołów IGP nie jest szczególnym problemem. Trzeba jednak pamiętać, że część z nich może być stosowana w systemach o określonej złożoności i wielkości.

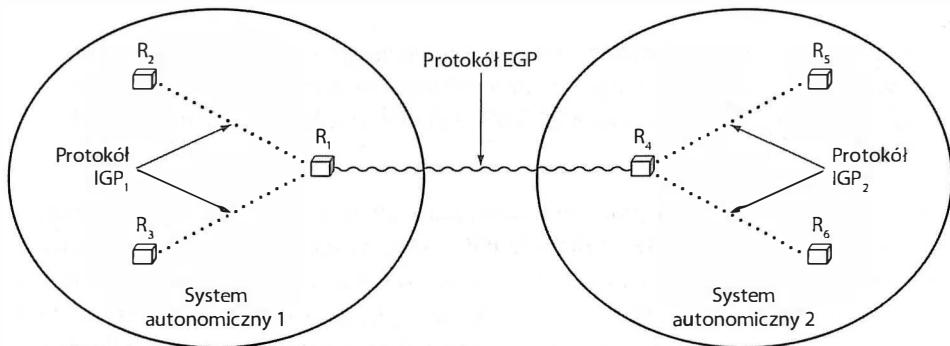
### 27.2.2. Protokoły routingu zewnętrznego

**Protokół routingu zewnętrznego** (EGP) służy routerowi jednego systemu autonomicznego do wymiany informacji o trasach z routerami innego systemu. Rozwiązania EGP są zazwyczaj znacznie trudniejsze w instalacji i utrzymaniu niż mechanizmy IGP. Zapewniają jednak większą elastyczność konfiguracji i wnoszą mniejszy narzut transmisyjny (generują ruch o mniejszym natężeniu). Aby zmniejszyć obciążenie łączy, oprogramowanie EGP agreguje trasy obowiązujące w danym systemie autonomicznym i przekazuje uogólnione informacje na ich temat do innego systemu. Ponadto rozwiązania EGP uwzględniają **politykę ograniczeń**, która umożliwia administratorowi sieci określenie, jakiego rodzaju dane są udostępniane na zewnątrz organizacji.

### 27.7.3. Sposób wykorzystania protokołów IGP i EGP

Dwupięciomowa hierarchia routingu internetowego została zaprezentowana na przykładzie dwóch grup routerów, które pracują w dwóch systemach autonomicznych (zgodnie z rysunkiem 27.3).

W powyższym przykładzie w systemie autonomicznym 1 ( $AS_1$ ) zastosowano protokół  $IGP_1$ , natomiast system autonomiczny 2 ( $AS_2$ ) pracuje pod kontrolą protokołu  $IGP_2$ . Komunikacja między routerami  $R_1$  i  $R_4$  (między dwoma systemami autonomicznymi) jest realizowana za pomocą protokołu EGP. Oznacza to, że jednostka  $R_1$  jest zobowiązana do zagregowania tras obowiązujących w jej systemie i przedstawienia ich w ogólnej formie routerowi  $R_4$ . Ponadto urządzenie  $R_1$  odpowiada za pobieranie uogólnionych informacji o sieciach od routera  $R_4$  i przedstawienie ich za pomocą protokołu  $IGP_1$  innym routerom w grupie  $AS_1$ . W systemie  $AS_2$  analogiczną funkcję pełni router  $R_4$ .



Rysunek 27.3. Wykorzystanie protokołów IGP wewnętrznych systemu autonomicznego oraz EGP między systemami

#### 27.7.4. Najlepsze trasy, metryki routingu i protokoły IGP

Mogłoby się wydawać, że zamiast wyznaczać jedną trasę między sieciami docelowymi, oprogramowanie routingu powinno wyszukiwać wszystkie istniejące trasy i wybierać najlepszą spośród nich. Mimo że w internecie zazwyczaj istnieje kilka sposobów na dostarczenie pakietów do sieci docelowej, trudno jednoznacznie zdefiniować kryteria wyboru optymalnego wariantu. Wystarczy взять pod uwagę wymagania różnych aplikacji komputerowych. W przypadku podłączenia zdalnego pulpu najkorzystniejsze jest połączenie o małym opóźnieniu. Z kolei pobieranie dużych rysunków przez przeglądarkę wymaga jak największej przepływności bitowej. W przekazach dźwięku liczy się przede wszystkim stałość fluktuacji opóźnienia.

Do określania jakości trasy służy **metryka routingu**. Na jej podstawie oprogramowanie routera wybiera optymalną trasę przekazywania pakietów. Choć metrykę można wyrażać za pomocą opóźnienia, przepustowości lub fluktuacji opóźnień, większość mechanizmów routingu bazuje na innych informacjach. Zazwyczaj metrykę wyznaczają **koszt administracyjny** oraz **liczba przeskóków**. W routingu internetowym przeskok odpowiada jednej sieci pośredniej (jednemu routerowi). Zatem liczba przeskóków do określonej sieci wskazuje, ile routerów występuje na trasie do danej sieci. Koszt administracyjny jest zazwyczaj definiowany przez administratora systemu i najczęściej służy do wyróżnienia jednej trasy względem innych. Założymy na przykład, że dział księgowości danej firmy jest połączony za pomocą dwóch łącz z działem płatności. Jedna trasa składa się z dwóch routerów pośrednich i służy do przenoszenia ruchu generowanego przez klientów. Druga natomiast obejmuje trzy przeskoki, ale jest przeznaczona dla ruchu wewnętrznego. Krótsza trasa może być niezgodna z polityką firmy, gdyż korzysta z łącz udostępnionych zewnętrznym użytkownikom sieci. W takim przypadku administrator systemu może zmienić ostateczny koszt dwuetapowej trasy przez nałożenie na nią kary, równej dwóm dodatkowym przeskokom (w ten sposób zastąpi rzeczywisty koszt dostarczania pakietów kosztem administracyjnym, który pozwoli na osiągnięcie zamierzzonego rezultatu). Oprogramowanie routingu wybierze trasę o najniższym koszcie, czyli tą, która przebiega przez trzy routery i jest zgodna z polityką firmy.

*Mimo że większość protokołów routingu internetowego posługuje się metrykami odpowiadającymi liczbie przeskóków, administrator sieci może narzucać inne wartości metryki, które pozwalają mu wdrażać politykę firmy w zakresie transmisji danych.*

Uwzględnianie metryk w pracy protokołu jest jednym z elementów istotnie różniących rozwiązań IGP od EGP. Protokoły IGP wykorzystują metryki w standardowym działaniu. Natomiast mechanizmy EGP z nich nie korzystają. Oznacza to, że wybór tras w ramach systemu autonomicznego bazuje na metrykach — oprogramowanie routera wysyła wartość metryki wraz z parametrami trasy, dzięki czemu urządzenie odbierające informacje może wybrać optymalną trasę pakietu. Jednak poza systemem autonomicznym trasy nie są wybierane. Protokół EGP jedynie odszukuje trasę bez jej wartościowania. Przyczyną jest to, że w każdym systemie autonomicznym metryki mogą być definiowane w inny sposób. Nie można więc ich wykorzystać do porównania różnych tras. Doskonałym uzasadnieniem takiego sposobu postępowania jest przykład dwóch systemów autonomicznych, z których jeden wyraża koszt trasy do sieci docelowej liczbą przeskóków, a drugi przepustowośćią łączy. Moduł obsługi protokołu EGP, dysponując dwoma metrykami, nie może wskazać lepszej trasy, ponieważ nie istnieje mechanizm przekształcania liczby przeskóków w przepustowość łączы. Z tego względu rozwiązanie EGP ogranicza się jedynie do raportowania dostępności tras, ale bez jej wartościowania.

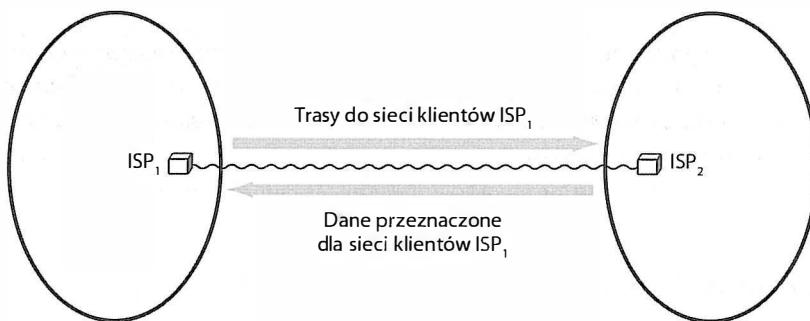
*Wyznaczanie tras w ramach systemu autonomicznego bazuje na protokołach IGP i wartościach metryk. Oprogramowanie EGP odpowiada jedynie za wyszukanie trasy, ale bez określania jej jakości, gdyż metryki różnych systemów autonomicznych nie mogą być ze sobą porównywane.*

## 27.8. Trasy i transport danych

W routingu obowiązuje zasada, zgodnie z którą odpowiedzią na ogłoszenia o trasach są dane. Idea nie jest szczególnie skomplikowana — dane zawsze są przesyłane w odwrotnym kierunku do informacji na temat tras. Jeśli w pewnym systemie autonomicznym, należącym do dostawcy ISP<sub>1</sub>, występuje sieć N, to przed przekazaniem ruchu do sieci N dostawca ISP<sub>1</sub> musi rozgłosić trasy do tej sieci. Oznacza to, że ruch z ogłoszeniami o sieci jest ruchem wchodzącym, a dane dostarczane do sieci mają charakter strumieni przychodzących. Zasadę tę ilustruje rysunek 27.4.

## 27.9. Protokół bram granicznych (BGP)

Szczególną popularność w internecie zdobył tylko jeden protokół routingu zewnętrznego — **protokół bram granicznych** (BGP — ang. *Border Gateway Protocol*). Miał on wiele wersji. Obecna, czwarta wersja jest oficjalnie zapisywana jako **BGP-4**. W praktyce



Rysunek 27.4. Przepływ danych po rozgłoszeniu tras do sieci dostawcy usług internetowych

jednak wersja ta obowiązuje już tak długo, że większość inżynierów posługuje się po prostu skrótem **BGP**, mając na myśli czwartą wersję tego protokołu.

Oto cechy charakterystyczne protokołu BGP:

- **Routing między systemami autonomicznymi.** Rozwiązywanie to zastosowanie w routingu zewnętrznym, więc umożliwia wymianę informacji na poziomie systemów autonomicznych. Trasy są więc określane jako ścieżki między kolejnymi systemami autonomicznymi. Na przykład trasa do określonej sieci może przechodzić przez systemy 17, 2, 56 i 12. Protokół BGP nie przekazuje informacji o metrykach ani o routerach pracujących w ramach systemu autonomicznego.
- **Obsługa polityki ruchu.** Protokół BGP umożliwia nadawcy i odbiorcy zdefiniowanie określonej polityki przenoszenia ruchu. Dzięki niej administrator może wykluczyć z rozpowszechniania niektóre trasy.
- **Obsługa routingu tranzytowego.** W systemie BGP obowiązuje podział systemów autonomicznych na systemy **tranzytowe** (ang. *transit*) oraz systemy **końcowe** (ang. *stub*). Systemy tranzytowe pozwalają na przekazywanie danych do innych systemów autonomicznych. Taki ruch przechodzący jest nazywany ruchem tranzytowym. Wprowadzenie klasyfikacji ruchu gwarantuje odróżnienie systemów autonomicznych należących do dostawców usług internetowych od systemów innych firm. Standard BGP pozwala na zaliczenie sieci korporacyjnej do grupy sieci końcowych, nawet jeśli jest ona przyłączona do internetu za pośrednictwem większej liczby łączy (gdy firma odmawia przenoszenia ruchu tranzytowego).
- **Niezawodny transport.** Komunikacja w ramach protokołu BGP bazuje na protokole TCP. Uruchomiony w jednym routerze program BGP ustanawia połączenie TCP z programem BGP routera innego systemu autonomicznego i przesyła dane w niezawodnym kanale. Protokół TCP gwarantuje bowiem dostarczanie informacji we właściwej kolejności i bez utraty pakietów.

Mechanizm BGP jest pewnego rodzaju spoiwem łączącym poszczególne elementy internetu — funkcjonujący w rdzeniu sieci firmy ISP pierwszego poziomu wykorzystuje protokół BGP do przekazywania sobie nawzajem informacji o klientach.

Protokół bram granicznych (BGP) jest rozwiązaniem z grupy zewnętrznych protokołów routingu, które jest wykorzystywane przez firmy ISP pierwszego poziomu do wymiany informacji o systemach autonomicznych. Obecnie stosowana jest czwarta wersja protokołu BGP.

## 27.10. Protokół informowania o trasach (RIP)

Protokół informowania o trasach (RIP — ang. *Routing Information Protocol*) był jednym z pierwszych rozwiązań routingu wewnętrznego stosowanych w internecie. Oto kilka cech protokołu RIP:

- **Obsługa routingu w ramach systemu autonomicznego.** Mechanizm RIP został zaprojektowany jako protokół routingu wewnętrznego do przekazywania informacji między routerami jednego systemu autonomicznego.
- **Metryka określająca liczbę przeskóków.** Odległość między sieciami jest wyrażana liczbą przeskóków. Jeden przeskok odpowiada jednej sieci znajdującej się na trasie między źródłem a celem pakietu. Odległość między dwoma sieciami przyłączonymi do jednego routera wynosi jeden przeskok.
- **Zawodny mechanizm transportowy.** Komunikaty RIP są przekazywane między routerami za pomocą protokołu UDP.
- **Transmisja rozgłoszeniowa lub multiemisja.** Protokół RIP jest stosowany w sieciach lokalnych, w których dopuszczalne są rozgłoszenia i multiemisje (takich jak Ethernet). W wersji 1. komunikaty były wysyłane rozgłoszeniowo. Natomiast w wersji 2. wykorzystywana jest multiemisja.
- **Obsługa adresowania CIDR i podsieci.** W wersji 2. protokołu RIP wraz z adresami sieci docelowych przekazywane są wartości masek podsieci.
- **Rozpowszechnianie informacji o bramie domyślnej.** Poza rozmieszczaniem tras do konkretnych sieci przekazywane są również dane na temat **trasy domyślnej**.
- **Algorytm wektora odległości.** Protokół RIP jest rozwiązaniem działającym zgodnie z techniką wyznaczania wektorów odległości, opisaną w algorytmie 18.3<sup>76</sup>.
- **Wersja pasywna przeznaczona dla komputerów.** Informacje o routingu są rozgląszane jedynie przez routery. Jednak inne stacje pracujące w sieci mogą nasłuchiwać aktualizacji i uwzględniać odebrane informacje w budowaniu tablicy routingu. Tryb pasywny jest szczególnie użyteczny w konfiguracjach, w których stacje mogą wybierać routery przekazujące dane.

Mechanizm propagowania informacji o trasach jest zgodny z zasadą działania protokołów wektora odległości. Każdy z wychodzących komunikatów zawiera ogłoszenia o sieciach, które nadawca ma w swoim zasięgu. Adresom towarzyszy wartość odległości do

<sup>76</sup> Algorytm 18.3 został przedstawiony na stronie 338.

danej sieci. Oprogramowanie RIP wykorzystuje odbierane powiadomienia do aktualizowania lokalnej tablicy routingu. Każdy wpis w ogłoszeniu RIP składa się z dwóch elementów:

(adres sieci docelowej, odległość)

Wartość odległość jest wyrażona liczbą przeskóków występujących na trasie do sieci docelowej. Odbiorca powiadomienia zapisuje uzyskaną informację o trasie w swojej tablicy routingu, jeśli nie miał wcześniej danej sieci lub wcześniejsza odległość do sieci jest większa niż zawarta w komunikacie.

Największą zaletą mechanizmu RIP jest jego prostota. Skonfigurowanie obsługi protokołu wymaga niewielkich nakładów pracy — zadanie administratora ogranicza się właściwie jedynie do uruchomienia oprogramowania protokołu w poszczególnych routerach i umożliwienia routerom rozgłaszenia aktualizacji. Po krótkim czasie wszystkie routery pracujące w sieci firmowej dysponują informacjami o trasach do wszystkich sieci w ramach systemu autonomicznego.

Protokół RIP uwzględnia również rozgłaszenie informacji o trasie domyślnej. Wystarczy więc, że administrator zdefiniuje trasę domyślną na jednym routerze sieci firmowej (zazwyczaj na routerze przyłączonym do sieci dostawcy usług internetowych), a protokół zajmie się dostarczeniem informacji o niej do wszystkich pozostałych routerów systemu autonomicznego. Dzięki temu wszystkie datagramy kierowane do jednostek spoza sieci firmowej są kierowane do urządzeń dostawcy usług internetowych.

## 27.11. Format pakietu RIP

Analiza formatu pakietu RIP może się okazać bardzo pomocna w zrozumieniu zasady działania protokołów routingu, które wykorzystują algorytm wyznaczania wektora odległości. Struktura komunikatu aktualizacji została pokazana na rysunku 27.5.

Jak wynika z rysunku, każdy wpis zawiera adres IP sieci docelowej oraz odległość do tej sieci. Ponadto, aby umożliwić wykorzystanie protokołu RIP w sieciach zawierających podsieci, do powiadomienia danych na temat sieci dodano 32-bitową wartość maski. W każdym wpisie występuje także adres routera następnego skoku, 16-bitowe pole informujące o tym, że wpis odnosi się do adresu IP, a także 16-bitowy identyfikator ułatwiający grupowanie wpisów. Ostatecznie każdy wpis składa się z dwudziestu oktetów.

*Protokół RIP jest protokołem routingu wewnętrznego, który wykorzystuje algorytm wektora odległości do propagowania informacji o trasach.*

## 27.12. Otwarty protokół wyznaczania najkrótszych tras (OSPF)

Format komunikatu RIP unaoczniła wady protokołów wektora odległości — rozmiar komunikatu jest wprost proporcjonalny do liczby sieci znanych routerowi. Wysyłanie powiadomień RIP nie jest natychmiastowe, a samo przetwarzanie informacji zajmuje wiele

0	8	16	24	31
Polecenie (1-5)	Wersja (2)	Musieć wartość zero		
Identyfikator rodziny adresów 1		Znacznik trasy do sieci 1		
Adres IP sieci 1				
Maska sieci 1				
Adres routera na trasie do sieci 1				
Odległość do sieci 1				
Identyfikator rodziny adresów 2		Znacznik trasy do sieci 2		
Adres IP sieci 2				
Maska sieci 2				
Adres routera na trasie do sieci 2				
Odległość do sieci 2				
...				

Rysunek 27.5. Format komunikatu aktualizacji w drugiej wersji protokołu RIP

cykli procesora. Brak natychmiastowej reakcji oznacza, że informacje o zmianie topologii są propagowane powoli — od routera do routera. Z tego powodu protokół RIP nie nadaje się do zastosowania w dużych sieciach (mimo że w systemach złożonych z kilku węzłów sprawdza się doskonale).

Aby zaspokoić potrzeby dużych przedsiębiorstw, organizacja IETF opracowała **otwarty protokół wyznaczania najkrótszych tras** (OSPF — ang. *Open Shortest Path First*). Nazwa mechanizmu pochodzi od opracowanego przez Edsgera Dijkstrę algorytmu SPF, który wyznacza najkrótsze ścieżki w grafie. Oto kilka cech protokołu OSPF:

- **Obsługa routingu w ramach systemu autonomicznego.** OSPF jest protokołem routingu wewnętrznego działającego w jednym systemie autonomicznym.
- **Obsługa adresowania CIDR.** Każdemu adresowi sieci towarzyszy 32-bitowa wartość maski. Dzięki temu mechanizm OSPF działa poprawnie w systemach z adresacją CIDR.
- **Wymiana komunikatów uwierzytelniających.** Wymiana informacji między dwoma routerami może być poprzedzona wzajemnym uwierzytelnieniem urządzeń.
- **Redystrybucja tras.** Routery OSPF mogą uwzględniać w powiadomieniach trasy pozyskane z innych protokołów routingu (na przykład z BGP).
- **Algorytm stanu łączny.** W działaniu protokołu OSPF uwzględniony jest **algorytm stanu łączny**, który został opisany w rozdziale 18.
- **Użycie metryk.** Administrator sieci może przypisywać każdej trasie dowolną wartość kosztu.
- **Obsługa sieci wielodostępnych.** Tradycyjne mechanizmy routingu na bazie algorytmu stanu łączny są nieefektywne w sieciach wielodostępnych (takich jak Ethernet).

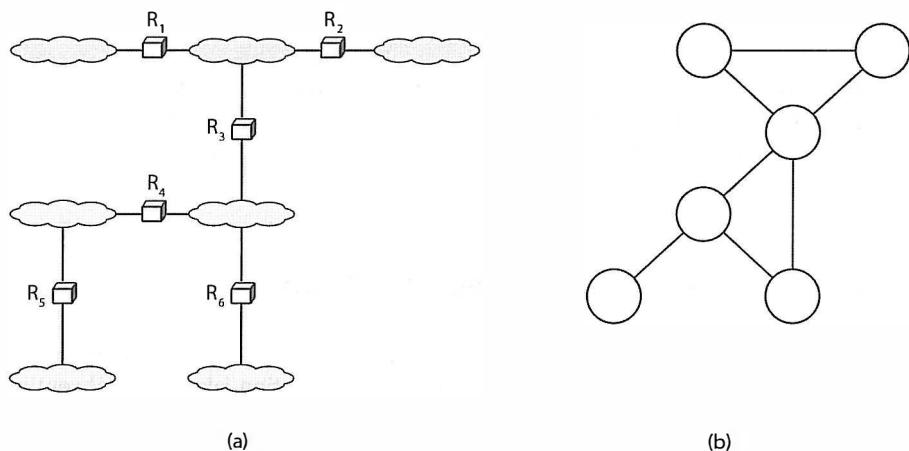
Wynika to z konieczności rozgłaszenia informacji o bieżącym stanie połączenia. W standardzie OSPF problem ten został rozwiązany przez wyznaczenie specjalnego routera, który zajmuje się rozgłaszaniem tego typu informacji w sieci.

Podsumowując:

*OSPF jest protokołem routingu wewnętrznego, który wykorzystuje algorytm stanu łączny do propagowania informacji o trasach oraz algorytm Dijkstry do obliczania najkrótszych tras.*

## 27.13. Przykład grafu OSPF

Zgodnie z informacjami zawartymi w rozdziale 18. algorytmy stanu łączny wykorzystują do obliczeń graf reprezentujący sieć. Choć przedstawienie sieci za pomocą grafu nie powinno stanowić trudności, przeanalizowanie prostego przykładu powinno dodatkowo ułatwić zrozumienie tej idei<sup>77</sup>. Rozważmy więc działanie sieci widocznej na rysunku 27.6.



Rysunek 27.6. Przykładowa topologia sieci (a) oraz odpowiadający jej graf OSPF (b)

Na rysunku zaprezentowano typowy graf OSPF, którego każdy węzeł odpowiada routerowi. Każda krawędź grafu reprezentuje z kolei połączenie między dwoma routery (czyli sieć). Zgodnie z algorytmem stanu łączny para routeraów przyłączona do wspólnej sieci testuje wzajemnie swoją dostępność, a następnie rozgłasza informację o stanie łączna do innych routerów. Każdy z routerów otrzymujących wiadomość aktualizuje na jej podstawie swoją kopię grafu, a w przypadku zmian statusu oblicza najkrótsze trasy.

<sup>77</sup> W praktyce grafy OSPF są znacznie bardziej skomplikowane niż zaprezentowany.

## 27.14. Obszary OSPF

Jedna cecha protokołu OSPF czyni go bardziej skomplikowanym niż inne podobne rozwiązania, ale jednocześnie zwiększa jego użyteczność. Jest nią hierarchiczność routingu. Zgodnie z założeniami mechanizmu OSPF systemy autonomiczne mogą być dodatkowo dzielone w celu optymalizacji routingu. W ten sposób powstają podzbiory sieci i routerów nazywane **obszarami**. W konfiguracji każdego routera zapisane są informacje o granicy obszaru (wymienione są routery wchodzące w skład obszaru), a wszystkie urządzenia należące do jednego obszaru wymieniają okresowe informacje o stanie łączy.

Poza wymianą danych między węzłami obszaru protokół OSPF zapewnia komunikację między obszarami. Jeden router w każdym obszarze jest konfigurowany w taki sposób, aby mógł się komunikować z analogicznymi routerami innych obszarów. Brzegowe routery obszarów agregują informacje o trasach pozyskane od innych routerów w obszarze i w uogólnionej formie przekazują do routerów brzegowych innych obszarów. Dzięki temu zamiast rozmawiania danych do wszystkich routerów systemu autonomicznego emitowanie powiadomień o stanie łączy jest ograniczone do pojedynczych obszarów. W rezultacie powstaje hierarchiczna struktura, która pozwala na obsługę znacznie większych systemów autonomicznych niż w przypadku innych protokołów routingu.

Najważniejsze jest więc to, że:

*Dzięki możliwości podziału routerów i sieci systemu autonomicznego na obszary protokół OSPF nadaje się do stosowania w systemach o znacznie większej liczbie routerów, niż mogłyby obsługiwać inne protokoły IGP.*

## 27.15. Protokół systemów pośrednich (IS-IS)

**Protokół systemów pośrednich**<sup>78</sup> IS-IS (ang. *Intermediate System to Intermediate System*) został zaprojektowany przez firmę Digital Equipment Corporation jako protokół routingu wewnętrznego i jest elementem specyfikacji DECNET V. Jego utworzenie zbiegło się w czasie z powstaniem mechanizmu OSPF. Poza tym rozwiązania te są zblizone funkcjonalnie. Obydwa należą do grupy protokołów stanu łączka i w obydwu zastosowano algorytm Dijkstry do obliczania najkrótszych tras. Działanie obydwu bazuje także na okresowym sprawdzaniu łączka między routerami i rozmawianiu informacji statusowych.

Najważniejsze różnice między protokołem OSPF i pierwotnym rozwiązaniem IS-IS:

- Protokół IS-IS jest standardem firmowym (należącym do DEC), a OSPF został utworzony jako otwarty standard (dostępny dla wszystkich producentów).

<sup>78</sup> Nazwa jest zgodna z terminologią stosowaną przez firmę DEC, w której router był nazywany **systemem pośrednim**, a komputer **systemem końcowym**.

- Protokół OSPF działa z wykorzystaniem mechanizmu IP. Informacje IS-IS są natomiast przenoszone przez usługę CLNS (będącą elementem fernalnego stosu protokołów OSI).
- Protokół OSPF jest przystosowany do przenoszenia informacji o trasach IPv4 (adresów IPv4 i masek podsieci). Natomiast IS-IS został zaprojektowany jako system wspomagający routing z wykorzystaniem protokołów OSI.
- Z czasem protokół OSPF uzupełniano o nowe funkcje, przez co IS-IS wnosi obecnie mniejszy narzut transmisyjny.

Otwarta specyfikacja OSPF i współdziałanie z protokołem IP sprawiły, że rozwiązanie to stało się znacznie bardziej popularne niż mechanizm IS-IS. Obecnie pierwotny standard IS-IS jest niemal całkowicie zapomniany. Popularność OSPF spowodowała, że przez lata organizacja IETF dodawała do niego nowe funkcje. Jednak na początku lat 2000, dziesięć lat po opracowaniu, protokół IS-IS zyskał drugą szansę. Firma Digital Equipment Corporation upadła, więc specyfikacja IS-IS przestała być własnością prywatną. Opracowano wówczas nową wersję protokołu IS-IS, która współdziałała z protokołem IP i internetem. Jednocześnie zauważono konieczność rozpoczęcia prac nad nową wersją specyfikacji OSPF, która obsługiwałaby protokół IPv6 (wcześniejsza była przeznaczona dla sieci IPv4). Poza tym największe firmy ISP rozrosły się do rozmiarów, w których brak dodatkowego narzutu rozwiązania IS-IS stał się istotny. Z tych powodów ponownie zaczęto interesować się mechanizmem IS-IS.

## 27.16. Routing w multiemisji

### 27.16.1. Działanie multiemisji IP

Wszelkie wcześniejsze rozważania na temat routingu dotyczyły propagowania informacji o sieciach docelowych, których adresy są stałe, a lokalizacja nie ulega zmianie. W przypadku emisji pojedynczej celem rozgłaszenia danych o trasach jest zagwarantowanie **stabilności** sieci — nieustanne zmiany tras są niepożądane, ponieważ prowadzą do zwiększenia fluktuacji opóźnień i dostarczania pakietów w niewłaściwej kolejności. Dlatego w przypadku wyznaczenia pewnej trasy mechanizmy routingu starają się z niej korzystać, aż do momentu, gdy stanie się niedostępna z powodu awarii.

Propagowanie informacji na temat **routingu w multiemisji** (ang. *multicast routing*) znacznie różni się od rozgłaszenia danych o trasach emisji pojedynczej. Różnica wynika z tego, że multiemisja zakłada dynamiczne formowanie grup komputerów i transmisję danych z anonimowych jednostek. Dynamiczne formowanie grup oznacza, że aplikacja może się przyłączyć do grupy w dowolnym czasie i pozostać w niej przez dowolnie długi okres. Mechanizm multiemisji działający w systemie komputera pozwala więc aplikacji na:

- Przyłączanie się do grupy multiemisji w dowolnym czasie i odbieranie wszystkich pakietów adresowanych do danej grupy. Przed przyłączeniem się do grupy komputer musi poinformować o tym zamiarze najbliższy router. Jeśli więcej aplikacji działających w tym samym komputerze należą do jednej grupy multiemisji, komputer

otrzymuje jeden pakiet, który jest lokalnie powielany i dostarczany do każdego z programów.

- Opuszczenie grupy multiemisji w dowolnym czasie. Każda jednostka grupy określowo wysyła do routera komunikat o podtrzymaniu swojego uczestnictwa w grupie. Gdy ostatnia aplikacja opuszcza grupę, komputer przesyła do lokalnego routera powiadomienie o wystąpieniu z grupy.

Uczestnictwo w grupie multiemisji jest anonimowe z dwóch względów. Po pierwsze, ani nadawca, ani odbiorca nie znają (i nie mogą ustalić) tożsamości i liczby członków grupy. Po drugie, ani komputery, ani routery nie dysponują informacjami o tym, jakie aplikacje będą wysyłały datagramy, ponieważ każda aplikacja może dostarczać pakiety do dowolnie wybranej grupy multiemisji. Członkostwo w grupie pozwala więc jedynie na zdefiniowanie zbioru odbiorców (nadawca nie musi należeć do grupy).

Podsumowując:

*Członkostwo w grupie multiemisji IP jest dynamiczne. Komputer może przyłączyć się do grupy lub ją opuścić w dowolnym czasie. Przynależność do grupy pozwala na wyznaczenie zbioru odbiorców. Każda aplikacja może wysłać datagram do grupy, nawet jeśli do niej nie należy.*

### 27.16.2. Protokół IGMP

W jaki sposób stacja przyłącza się do grupy lub ją opuszcza? Powiadamianie routera o przystąpieniu do grupy lub jej opuszczeniu należy do zadań **internetowego protokołu grup multiemisji** (IGMP — ang. *Internet Group Multicast Protocol*), który jest wykorzystywany jedynie na połączeniu komputer-router. Protokół ten przyłącza do grupy stacje, a nie aplikacje. Nie dostarcza również żadnych dodatkowych informacji o aplikacjach. Jeśli do grupy multiemisji należy więcej aplikacji jednego komputera, system danej jednostki musi wykonać odpowiednią liczbę kopii każdego odbieranego datagramu i dostarczyć je do lokalnych programów. Wraz z zakończeniem pracy ostatniej aplikacji komputer musi za pomocą protokołu IGMP powiadomić lokalny router o wystąpieniu z grupy.

### 27.16.3. Techniki przesyłania pakietów i wykrywania stacji

Wraz z zarejestrowaniem komputera w grupie multiemisji router musi wyznaczyć do niego trasę i rozpoczęć przekazywanie wszystkich datagramów adresowanych do danej grupy. Zatem odpowiedzialność za rozgłaszanie informacji o trasach w multiemisji spoczywa na routera ch, a nie na komputerach.

Dynamiczne członkostwo w grupie oraz anonimowość nadawcy sprawiają, że routing w multiemisji jest niezwykle trudny do realizacji. Zadania nie ułatwia to, że rozmiar i topologia grupy zmieniają się istotnie w zależności od aplikacji. Na przykład telekonferencje zazwyczaj obejmują kilka stacji (od dwóch do pięciu jednostek), które mogą pracować

w kilku odległych geograficznie miejscach lub w ramach tej samej organizacji. Z kolei aplikacje internetowych przekazów audiowizualnych mogą tworzyć grupy złożone z milionów członków rozmieszczonych na całym świecie.

Aby poprawnie obsługiwać dynamiczną przynależność komputerów do grupy, protokoły routingu w każdej chwili muszą mieć możliwość zmiany tras. Na przykład jeśli użytkownik pracujący we Francji przyłączy się do grupy multiemisji obejmującej stacje z Polski i Japonii, oprogramowanie routingu musi w pierwszym kroku odnaleźć pozostałych członków grupy, a następnie opracować optymalną strukturę tras. Ponadto możliwość wysłania datagramu do grupy przez dowolnego użytkownika powoduje, że informacje o trasach muszą uwzględniać również stacje spoza samej grupy. W praktyce przesyłanie datagramów jest realizowane za pomocą jednej z trzech technik:

- „zalej i odetnij” (ang. *flood-and-prune*);
- konfiguracja i tunelowanie (ang. *configuration-and-tunneling*);
- wyszukiwanie w rdzeniu (ang. *core-based discovery*).

**„Zalej i odetnij”.** Technika ta doskonale nadaje się do stosowania w małych grupach oraz w przypadkach, w których wszystkie stacje członkowskie są przyłączone do ciągłego zbioru sieci LAN (na przykład do sieci korporacyjnej). Początkowo routery przekazują datagramy do wszystkich sieci — gdy router odbierze pakiet z adresem multiemisji, wysyła go do wszystkich bezpośrednio przyłączonych segmentów LAN z wykorzystaniem sprzętowego mechanizmu multiemisji. Aby uniknąć pętli routingu, stosuje się technikę **rozgłaszenia na podstawie tras powrotnych** (RPB — ang. *Reverse Path Broadcasting*). W czasie zalewania sieci pakietami routery wymieniają informacje o przynależności stacji do grupy. Jeśli w danej sieci żaden komputer nie należy do określonej grupy, router przestaje przekazywać ruch multiemisyjny („odcina” sieć).

**Konfiguracja i tunelowanie.** Technika ta znajduje zastosowanie przede wszystkim w konfiguracjach, w których grupa komputerów jest rozproszona geograficznie (tj. składa się z kilku stacji w każdej lokalizacji, a same lokalizacje są odległe od siebie). Router w każdym ośrodku jest skonfigurowany w taki sposób, aby znał umiejscowienie innych ośrodków. Po odebraniu pakietu multiemisji router powiela datagram we wszystkich bezpośrednio przyłączonych sieciach LAN (wykorzystując do tego celu sprzętowe komponenty multiemisji), a następnie sprawdza parametry konfiguracji, aby ustalić, do jakich ośrodków zdalnych powinien przekazać kopie pakietów. Dostarczeniem danych do zdalnych lokalizacji zajmuje się mechanizm tunelowania pakietów IP w pakietach IP.

**Wyszukiwanie w rdzeniu.** Choć techniki „zalej i odetnij” oraz konfiguracji i tunelowania doskonale sprawdzają się w dwóch skrajnych przypadkach, potrzebne jest również rozwiązanie, które zapewni możliwość skalowania systemu multiemisji od niewielkich grup w jednej sieci do rozbudowanych grup w różnych lokalizacjach. Aby zapewnić możliwość rozszerzania się grupy, niektóre protokoły routingu grupowego wyznaczają **rdzeniowy adres emisji pojedynczej**. Za każdym razem, gdy router  $R_1$  odbierze datagram, który powinien zostać dostarczony do grupy, umieszcza go w klasycznym datagramie emisji pojedynczej i przesyła na adres rdzeniowy. Podczas przekazywania pakietu przez internet każdy router może przeanalizować treść datagramu. Gdy datagram dociera do routera  $R_2$ ,

(należącego do grupy), datagram multiemisji jest wyodrębniany z pakietu zewnętrznego i dostarczany do wszystkich członków grupy. Ten sam mechanizm obowiązuje podczas przyłączania komputerów do grupy — gdy router  $R_2$  odbierze żądanie przyłączenia stacji do grupy, dodaje nową trasę do klasycznej tablicy routingu i rozpoczyna przekazywanie do routera  $R_1$  kopii każdego datagramu wysłanego w trybie multiemisji. Zbiór routerów odbierających datagramy multiemisji rozrasta się więc od węzła początkowego nazywanego rdzeniem. W teorii grafów taka konfiguracja zostałaby określona jako **drzewo**.

#### 27.16.4. Protokoły multiemisji

Mimo że opracowano wiele protokołów routingu przeznaczonych do stosowania w multiemisji, żaden nie jest wykorzystywany na skalę globalną. Do najważniejszych propozycji należy zaliczyć:

**DVMRP** (ang. *Distance Vector Multicast Routing Protocol*). Protokół ten jest wykorzystywany przez program mrouted dostępny w systemie UNIX oraz internetową sieć MBONE. Mechanizm DVRMP bazuje na lokalnym rozsyłaniu grupowym oraz wykorzystaniu enkapsulacji IP w IP do przesyłania datagramów multiemisji między zdalnymi sieciami. Więcej informacji na temat sieci MBONE można znaleźć na stronie:

<http://www.lbl.gov/web/Computers-and-Networks.html#MBONE>

**CBT** (ang. *Core Based Trees*). Protokół, którego działanie polega na budowaniu drzew z punktem centralnym każdej grupy. Dostarczanie danych do punktu centralnego jest realizowane zgodnie z klasycznymi zasadami routingu.

**PIM-SM** (ang. *Protocol Independent Multicast — Sparse Mode*). W rozwiązaniu tym wykorzystuje się tę samą technikę budowania drzew multiemisji, która jest stosowana w protokole CBT. Projektanci mechanizmu podkreślają jednak jego **niezależność od protokołu**. Mimo że są w nim wykorzystywane datagramy emisji pojedynczej (do wymiany danych ze zdalnymi ośrodkami w czasie konfigurowania systemu), nie zależy on od żadnego konkretnego protokołu routingu.

**PIM-DM** (ang. *Protocol Independent Multicast — Dense Mode*). Jest to protokół przeznaczony do stosowania wewnętrz organizacji. Routery obsługujące mechanizm PIM-DM rozgłaszą pakiety multiemisji we wszystkich ośrodkach organizacji (zalewają je pakietami). Każdy router, który nie zarejestrował żadnego członka danej grupy, odsyła zwrotnie komunikat o **odcięciu** danej gałęzi drzewa (jest to żądanie wstrzymania przesyłania pakietów). Rozwiązanie to doskonale się sprawdza w przypadku krótkotrwałych sesji (na przykład kilkuminutowych), ponieważ nie wymaga wstępnego konfigurowania urządzeń sieciowych.

**MOSPF** (ang. *Multicast Extensions To The Open Shortest Path First Protocol*). Mechanizm MOSPF nie jest ogólnym protokołem routingu pakietów multiemisji, ale rozwiązaniem, które przekazuje informacje o trasach multiemisji między routerami jednej organizacji. Jego działanie bazuje na grafie OSPF i funkcji LSR.

Wszystkie wymienione powyżej protokoły routingu związanego z multiemisją zostały przedstawione w tabeli 27.1.

**Tabela 27.1.** Protokoły routingu multiemisji i sposób ich działania

Protokół	Sposób działania
DVMRP	Konfiguracja i tunelowanie
CBT	Wyszukiwanie w rdzeniu
PIM-SM	Wyszukiwanie w rdzeniu
PIM-DM	„Zalej i odetnij”
MOSPF	Stan łączy (w ramach organizacji)

Mimo dwudziestu lat badań i eksperymentów multiemisja w internecie nadal nie jest powszechna. Nawet aplikacje pracy grupowej nie zmieniły tego stanu rzeczy. Całe zagadnienie można więc podsumować w następujący sposób:

*Dynamiczny charakter internetowej multiemisji istotnie utrudnia opracowanie mechanizmu propagowania tras służących do przenoszenia tego rodzaju ruchu. Mimo opracowania wielu protokołów funkcje routingu w multiemisji nadal są kwestią przyszłości.*

## 27.17. Podsumowanie

Większość komputerów wykorzystuje routing statyczny. Ich tablice routingu są inicjowane w chwili uruchamiania systemu operacyjnego. Routery na bieżąco aktualizują swoje tablice routingu, używając do tego celu protokołów routingu. Z uwagi na konieczność wyznaczania tras internet został podzielony na systemy autonomiczne. Protokoły odpowiadające za przekazywanie informacji o trasach między systemami autonomicznymi są nazywane protokołami routingu zewnętrznego (EGP). Natomiast protokoły dystrybuujące dane o trasach wewnętrz systemu autonomicznego są określane jako protokoły routingu wewnętrznego (IGP).

Podstawowym protokołem EGP internetu jest protokół bram granicznych (EGP). Dostawcy usług internetowych zaliczani do operatorów pierwszego poziomu wykorzystują mechanizm BGP do informowania siebie nawzajem o sieciach klientów. Do rozwiązań IGP zalicza się: RIP, OSPF oraz IS-IS.

Rozgłaszenie informacji o trasach przeznaczonych do przenoszenia ruchu multiemisji jest wyjątkowo trudne, ponieważ zaimplementowane mechanizmy umożliwiają komputerom dynamiczne formowanie grup, a nawet przesyłanie pakietów do grupy bez wcześniejszego przyłączania się do niej. Mimo że opracowano kilka protokołów routingu multiemisji, żadne z rozwiązań nie zostało wdrożone na skalę globalną.

## ZADANIA

- 27.1. Wymień dwie ogólne kategorie routingu internetowego i opisz każdą z nich.
- 27.2. Jakie dwa wpisy występują w tablicy routingu standardowych jednostek końcowych?
- 27.3. Wykaż, że zdefiniowanie trasy domyślnej w każdym routerze internetowym musiałoby doprowadzić do powstania pętli routingu.
- 27.4. Czym jest system autonomiczny?
- 27.5. Wymień i scharakteryzuj dwa rodzaje protokołów routingu internetowego.
- 27.6. Funkcjonujący w pewnej organizacji protokół routingu dostarcza informacje o tym, że jedna z sieci docelowych jest odległa o dziesięć routerów, mimo że faktycznie jest odległa o trzy routery. Czy taka konfiguracja na pewno jest błędna? Uzasadnij odpowiedź.
- 27.7. Jakie są konsekwencje uwzględnienia danej sieci docelowej w ogłoszeniu generowanym przez router?
- 27.8. Wymień i opisz cechy protokołu BGP.
- 27.9. Jaki jest obszar zastosowań protokołu BGP?
- 27.10. Jaki rodzaj algorytmu routingu został wykorzystany w protokole RIP? Do czego jest wykorzystywany protokół RIP?
- 27.11. Wymień cechy protokołu RIP.
- 27.12. W jaki sposób router dzieli otrzymane w protokole RIP adresy IP na prefiksy i sufiksy?
- 27.13. Napisz program komputerowy, który przeanalizuje komunikat RIP i wyświetli na ekranie zawartość każdego pola.
- 27.14. Maksymalna liczba przeskóków w protokole RIP wynosi 16. Zaproponuj sieć firmową, która będzie złożona z więcej niż 16 routerów i więcej niż 16 sieci, ale jednocześnie będzie się nadawała do zastosowania protokołu RIP.
- 27.15. Wymień cechy protokołu OSPF.
- 27.16. Co oznacza słowo „otwarty” w rozwinięciu skrótu OSPF?
- 27.17. Dlaczego w specyfikacji OSPF zdefiniowano wiele obszarów?
- 27.18. Który protokół wnosi mniejszy narzut: OSPF czy IS-IS? Który ma więcej funkcji?
- 27.19. Jakie jest zasadnicze przeznaczenie protokołu IGMP?
- 27.20. Wymień trzy najważniejsze techniki przesyłania datagramów multiemisji.
- 27.21. Jaki protokół routingu należałby zastosować w przypadku multiemisji związanej z realizacją połączenia telekonferencyjnego między trzema osobami pracującymi w trzech miastach?
- 27.22. Każda grupa multiemisji wykorzystuje niepowtarzalny adres IP. Użycie centralnego serwera przydzielającego adresy wiąże się z ryzykiem powstania zatoru. Opracuj mechanizm umożliwiający grupie komputerów wybranie losowego adresu multiemisji i rozwiązanie konfliktu, jeśli takowy wystąpi.
- 27.23. Ruch generowany przez mechanizm „zalej i odetnij” jest przyczyną ograniczania obszaru stosowania protokołu w sieci. Oszacuj całkowite natężenie ruchu w jednej sieci, jeśli G grup multiemisji generuje dane z częstotliwością P pakietów na sekundę, a każdy pakiet zawiera B bitów. Sieć firmowa składa się z N podsieci, a w każdej podsieci znajduje się co najmniej jeden odbiorca z każdej grupy.
- 27.24. Czy multiemisja jest powszechnie wykorzystywana w internecie? Wyjaśnij zagadnienie.
- 27.25. Który protokół multiemisji pozwala na przesłanie komunikatu przed wyznaczeniem tras?

# CZĘŚĆ V

## Inne aspekty funkcjonowania sieci komputerowych

**Wydajność sieci, QoS, bezpieczeństwo, zarządzanie oraz technologie wschodzące**

### Rozdziały:

Rozdział 28. Wydajność sieci (QoS i DiffServ)	491
Rozdział 29. Multimedia i telefonia IP (VoIP)	513
Rozdział 30. Bezpieczeństwo sieci	531
Rozdział 31. Zarządzanie siecią (SNMP)	557
Rozdział 32. Trendy w technologiach sieciowych i sposobach wykorzystywania sieci	571

# Zawartość rozdziału

28.1. Wprowadzenie	491
28.2. Miary wydajności	491
28.3. Opóźnienie	492
28.4. Przepustowość, pojemność	
i efektywna szybkość dostarczania danych	494
28.5. Zrozumienie przepustowości i opóźnienia	495
28.6. Fluktuacja opóźnienia	496
28.7. Zależność między opóźnieniem a przepustowością	497
28.8. Pomiar opóźnienia, przepustowości i fluktuacji opóźnienia	499
28.9. Pomiar pasywny, małe pakiety i mechanizm NetFlow	500
28.10. Jakość usługi (QoS)	501
28.11. Ogólna i szczegółowa specyfikacja QoS	502
28.12. Implementacja mechanizmów QoS	505
28.13. Internetowe technologie QoS	506
28.14. Podsumowanie	508

# 28

## *Wydajność sieci (QoS i DiffServ)*

### 28.1. Wprowadzenie

W początkowych rozdziałach książki przedstawiono fundamentalne cechy systemów komunikacyjnych oraz zależności między sygnałami, częstotliwościami, szerokością pasma, kodowaniem kanałowym i transmisją danych. Wyjaśniono w nich zasady wartościowania systemów transmisji danych, omówiono rozmiary sieci i podziały sieci na systemy PAN, LAN, MAN i WAN.

Ten rozdział jest kontynuacją wcześniejszych rozważań, ponieważ odnosi się do zagadnienia wydajności sieci. Zawiera omówienie miar efektywności działania oraz sposobów wykorzystania protokołów i określonych technik przekazywania pakietów do zwiększenia priorytetu wybranego rodzaju ruchu.

### 28.2. Miary wydajności

W mowie potocznej zazwyczaj używamy terminu **szynkość** do określenia wydajności sieci. Twierdzimy, że sieć jest **wolna** albo **szynka**. Taki podział bywa jednak zawodny, ponieważ szybki postęp w opracowywaniu technologii sieciowych sprawia, że rozwiązanie sklasyfikowane jako „szynkie” po trzech lub czterech latach staje się „wolne”. Dlatego zamiast opisów jakościowych naukowcy i inżynierowie stosują ilościowe miary, które precyzyjnie określają wydajność systemów. Po omówieniu podstawowych miar wydajności przedstawione zostaną zasady ich wykorzystywania w implementowaniu usług o różnych poziomach jakości obsługi. Mimo że osoby początkujące często wolą posługiwać się nieformalnymi opisami sieci, powinny nauczyć się stosowania miar ilościowych, ponieważ pozwalają one na porównywanie tych samych aspektów funkcjonowania dwóch sieci i budowanie mechanizmów gwarantujących priorytetowe traktowanie określonego rodzaju ruchu. Najważniejsze miary wydajności sieci zostały przedstawione w tabeli 28.1. Każda z opisanych wielkości została wyjaśniona w dalszej części rozdziału.

Tabela 28.1. Najważniejsze miary wydajności sieci

Miara	Opis
Opóźnienie	Czas potrzebny na przesłanie danych przez sieć.
Przepustowość (pojemność)	Ilość danych, które można przesyłać w jednostce czasu.
Fluktuacja opóźnień (jitter)	Zmiany w opóźnieniu oraz czas trwania zmian.

{}

### 28.3. Opóźnienie

Pierwszą z cech sieci, którą można dokładnie pomierzyć, jest **opóźnienie**. Wartość opóźnienia informuje o tym, ile czasu zajmuje przesyłanie danych z jednego komputera pracującego w sieci do innego. Zazwyczaj wartości te odpowiadają ułamkowym częściom sekundy. Opóźnienie w dostarczaniu danych przez internet zależy od infrastruktury połączeń oraz lokalizacji dwóch komunikujących się jednostek. Użytkownicy przykładają dużą wagę do opóźnień sieciowych, dlatego inżynierowie muszą się posługiwać precyzyjnymi miarami tej wielkości. Zazwyczaj określają zarówno maksymalne, jak i średnie opóźnienie, a także rozkładają całkowite opóźnienie na kilka elementów składowych. W tabeli 28.2 przedstawiono różne rodzaje opóźnień.

Tabela 28.2. Różne rodzaje opóźnień wraz z opisem

Rodzaj	Opis
Opóźnienie propagacyjne	Czas potrzebny do przesłania sygnału w medium transmisyjnym.
Opóźnienie dostępu do medium	Czas potrzebny na uzyskanie dostępu do medium transmisyjnego (na przykład do kabla).
Opóźnienie przełączania	Czas potrzebny na przekazanie pakietu przez urządzenie sieciowe.
Opóźnienie kolejkowania	Czas przechowywania pakietu w pamięci przełącznika lub routera przed pobraniem w celu wysłania.
Opóźnienie serwera	Czas potrzebny do przetworzenia żądania w serwerze i wysłania odpowiedzi.

**Opóźnienie propagacyjne.** Niektóre opóźnienia sieciowe wynikają z tego, że przesyłanie sygnału przez medium transmisyjne zajmuje pewien czas. Ogólnie opóźnienia propagacyjne są zależne od długości łącza. Ich wartości w kablach sieci LAN rozciągniętych na dużych odległościach w ramach jednego budynku nie przekraczają milisekundy. Choć mogłoby się wydawać, że są one nieistotne dla użytkownika sieci, trzeba pamiętać, że w czasie jednej milisekundy nowoczesny procesor jest w stanie wykonać ponad sto tysięcy instrukcji. Zatem milisekundowe opóźnienie może się okazać istotne w koordynacji pracy wielu komputerów (na przykład w bankowości, w której dostarczenie na czas zlecenia

zakupu akcji decyduje o tym, czy operacja zostanie wykonana, czy nie). Sieci wykorzystujące satelity GEO charakteryzują się znacznie większymi opóźnieniami — nawet w przypadku transmisji z prędkością światła przesłanie każdego bitu do satelity i z powrotem na Ziemię zajmuje kilkaset milisekund.

**Opóźnienie dostępu do medium.** W wielu sieciach wykorzystywane są współdzielone media transmisyjne. Doskonałym przykładem jest tutaj sieć Wi-Fi, w której dostęp do medium jest regulowany przez algorytm CSMA/CA. Opóźnienia związane z rywalizacją o możliwość wyemitowania sygnału są nazywane **opóźnieniami dostępu do medium**. Ich wartości zależą od liczby stacji oraz natężenia ruchu generowanego przez każdą z jednostek sieciowych. Jeśli medium nie jest przeciążone, opóźnienie utrzymuje się na niskim stałym poziomie.

**Opóźnienie przełączania.** Elektroniczne urządzenia sieciowe (na przykład przełączniki warstwy 2. lub routery) przed wysłaniem pakietu muszą wyznaczyć kolejną jednostkę na jego trasie do komputera docelowego. Określenie trasy wymaga przeanalizowania tablicy przełączania, czyli odwołania do pamięci. W niektórych urządzeniach dodatkowo trzeba uwzględnić czas potrzebny na przesłanie pakietu wewnętrznej magistrali. Łączny czas wyznaczania następnego urządzenia oraz przygotowania danych do transmisji nazywa się **opóźnieniem przełączania**. Dzięki wydajnym procesorom i specjalnie zaprojektowanym układom sprzętowym opóźnienia przełączania są w nowoczesnej sieci najmniej istotnym składnikiem całociowymi opóźnień transmisyjnych.

**Opóźnienie kolejkowania.** Stosowana powszechnie w przełączaniu pakietów technika „zapisz i przekaż” wiąże się z tym, że urządzenia takie jak router muszą gromadzić bity pakietów, zapisywać je w pamięci, wyznaczać kolejne odcinki tras i oczekiwając na możliwość rozpoczęcia emisji. Opóźnienia wprowadzane w ostatniej fazie opisanego procesu nazywa się **opóźnieniami kolejkowania**. W najmniej skomplikowanym przypadku pakiet jest zapisywany w wyjściowej kolejce FIFO, w której oczekuje na zakończenie transmisji innych pakietów emitowanych przed nim. W bardziej wyrafinowanych systemach należy uwzględnić również działania algorytmów, które w pierwszej kolejności wybierają pakiety o wyższym priorytecie. Opóźnienia kolejkowania mają zmienne wartości — długość kolejki zależy od liczby odebranych wcześniej pakietów. Są jednak przyczyną większości opóźnień w transmisji danych przez internet. Gdy wydłużają się nadmiernie, dochodzi do przeciążenia sieci.

**Opóźnienie serwera.** Choć serwery nie są komponentami samej sieci, stanowią ważny element większości przypadków komunikacji. Czas potrzebny na przetworzenie żądania i wysłanie odpowiedzi jest istotnym składnikiem całociowego opóźnienia w wymianie danych. Serwery kolejkują nadchodzące żądania, a to oznacza, że wartość opóźnienia jest zmienna i zależy od bieżącego obciążenia komputera. W większości przypadków opóźnienia zauważane przez użytkowników sieci wynikają z długiego przetwarzania żądań po stronie serwera, a nie z pracy samej sieci.

## 28.4. Przepustowość, pojemność i efektywna szybkość dostarczania danych

Drugim mierzalnym parametrem sieci jest jej **pojemność**, czyli maksymalna **przepustowość**, z jaką sieć może pracować. Przepustowość jest miarą ilości danych przekazywanych przez sieć w jednostce czasu i jest wyrażana w **bitach na sekundę** (b/s). Większość sieci przeznaczonych do transmisji danych zapewnia przepustowość co najmniej na poziomie 1 Mb/s. Najszybsze sieci pracują z przepustowością większą niż 1 Gb/s. Doświadczenie podpowiada jednak, że zdarzają się również sieci o przepustowości mniejszej niż 1 kb/s.

Ponieważ przepustowość można mierzyć na kilka sposobów, konieczne jest zachowanie szczególnej ostrożności w opisywaniu tego, co zostało uwzględnione w przeprowadzonym pomiarze. Oto kilka możliwości:

- pojemność pojedynczego kanału,
- sumaryczna pojemność wszystkich kanałów,
- teoretyczna pojemność łącza fizycznego,
- efektywna szybkość dostarczania danych do aplikacji.

Dostawcy sprzętu często podają teoretyczną przepustowość produkowanych urządzeń, a także przepływność danych w optymalnych warunkach transmisyjnych. Pojemność warstwy sprzętowej służy często jako przybliżenie potencjalnej przepustowości, ponieważ wyznacza ona górną granicę wydajności systemu — aplikacja nie może przesyłać danych szybciej, niż wynika to z konstrukcji urządzenia.

Użytkownicy nie zastanawiają się nad pojemnością warstwy sprzętowej. Są zainteresowani jedynie szybkością, z jaką dane można przesłać do drugiego urządzenia. Posługują się więc wartościami **efektywnej szybkości dostarczania danych**, które są niższe niż wartości przepustowości, ponieważ nie uwzględniają narzutu transmisyjnego — stosowanie różnorodnych protokołów powoduje, że część pojemności systemu nie jest dostępna dla jego użytkowników. Narzut protokołów wynika z:

- dołączania nagłówków, stopek i informacji kontrolnych,
- ograniczania rozmiaru okna transmisyjnego (rozmiaru bufora odbiorczego),
- wykorzystywania protokołów odwzorowania nazw i adresów,
- stosowania specjalnych mechanizmów nawiązywania i rozłączania połączenia,
- ograniczania szybkości transmisji w przypadkach przeciążeń,
- retransmitowania ultraconykh pakietów.

Wadą wykorzystywania efektywnej szybkości dostarczania danych jako parametru sieci jest to, że narzut transmisyjny zależy od rodzaju stosowanych protokołów. Poza protokołem transportowym, internetowym i sieciowym trzeba również uwzględnić protokół warstwy aplikacji. Przeanalizujmy na przykład wykorzystanie aplikacji korzystającej z **protokołu transferu plików** (FTP — ang. *File Transfer Protocol*) jako sposobu na oszacowanie wydajności dostarczania danych w Ether necie. Mechanizm FTP bazuje na protokole TCP, który z kolei korzysta z IP. Działanie aplikacji FTP nie uwzględnia kompresji danych.

Dane dostarczone przez użytkownika są zapisywane w segmentach TCP. Każdy segment TCP podlega enkapsulacji w datagramie IP, który następnie jest umieszczany w polu danych ramki ethernetowej. Ostatecznie więc każda porcja danych jest uzupełniona o nagłówek ramki, pole CRC, nagłówek datagramu IP oraz nagłówek segmentu TCP. Jeśli użytkownik skorzysta z innej aplikacji transferu plików lub z innego stosu protokołów, efektywna wydajność transmisji może ulec istotnej zmianie.

*Mimo że miara efektywnej szybkości dostarczania danych dostarcza informacji o wydajności sieci, jej wartość zależy od rodzaju wykorzystywanej aplikacji.*

## 28.5. Zrozumienie przepustowości i opóźnienia

Terminologia stosowana przez doświadczonych administratorów sieci do opisu przepustowości systemu lub jego pojemności bywa niekiedy dość myląca. Na przykład w rozdziale poświęconym transmisji danych wprowadzone zostało pojęcie szerokości pasma kanału, któremu towarzyszyło wyjaśnienie zależności między szerokością pasma w urządzeniach a maksymalną szybkością transmisji danych. Niestety, inżynierowie często zamiennie używają terminów **szerokość pasma** i **szybkość transmisji**. Nietrudno więc usłyszeć, że określona sieć pracuje z „szerokością 1 Gb/s” albo że jej „pasmo to 1 Gb/s”. Aby móc rozróżnić dwa przypadki stosowania terminu **szerokość pasma**, określenie **szerokość pasma** stosuje się w odniesieniu do **szerokości pasma analogowego**, natomiast **szerokość pasma cyfrowego** jest określana jako **przepustowość**. Choć często można usłyszeć stwierdzenia podobne do przytoczonych, niekiedy mogą się one okazać mylące, ponieważ przepustowość, opóźnienie i szerokość pasma to trzy zupełnie różne wielkości.

Tak naprawdę, przepustowość jest miarą pojemności, a nie szybkości. Aby zrozumieć tę zależność, wystarczy wyobrazić sobie sieć jadącą drogę między dwoma punktami, a przesybane przez sieć pakiety jako samochody przemiesczające się od jednego punktu do drugiego. Przepustowość określa liczbę samochodów, które mogą wjechać na drogę w każdej sekundzie. Opóźnienie propagacyjne odpowiada czasowi, jaki jest potrzebny na przejście z jednego punktu do drugiego. Na przykład droga, na którą co pięć sekund może wjechać jeden samochód, ma przepustowość 0,2 samochodu na sekundę. Jeśli przejazd zajmuje 30 sekund, opóźnienie propagacyjne również wynosi 30 sekund. Zastanówmy się, co się stanie, gdy zostanie otwarty drugi pas jezdni (podwoi się pojemność odcinka). Od tego momentu co pięć sekund na drogę będą mogły wjeżdżać dwa samochody. Przepustowość podwoi się, osiągając wartość 0,4 samochodu na sekundę. Oczywiście, opóźnienie pozostanie niezmienne, gdyż każdy samochód nadal musi pokonać tę samą odległość, a to zajmuje 30 sekund. Dlatego zastanawiając się nad parametrami sieci, trzeba pamiętać, że:

*Opóźnienie propagacyjne określa czas, jaki jest potrzebny na przesłanie jednego bitu przez sieć. Przepustowość odpowiada maksymalnej liczbie bitów, którą można wprowadzić do sieci w jednostce czasu. Przepustowość wyznacza więc pojemność sieci.*

Administratorzy sieci mają takie powiedzenie:

*Zawsze możesz kupić dodatkową przepustowość, ale nie kupisz mniejszego opóźnienia.*

Porównanie do drogi pomaga w zrozumieniu tego powiedzenia. Dodanie kolejnych pasów na jezdni wpływa na zwiększenie liczby samochodów, które mogą wjechać na drogę w danym okresie, ale nie zmniejsza czasu potrzebnego na przejście odcinka. W sieciach obowiązuje ta sama zależność: dodanie równoległych tras przesyłania pakietów zwiększa przepustowość sieci, ale nie obniża opóźnienia propagacyjnego, które zależy od odległości między urządzeniami.

## 28.6. Fluktuacja opóźnienia

Trzecia własność sieci jest niezwykle istotna w przypadku przekazów audiowizualnych. Parametr **fluktuacji opóźnienia**, określany również jako **jitter** (czytaj *dżiter*), opisuje zmienność wartości opóźnienia. Dwie sieci mogą się na przykład cechować takim samym opóźnieniem, ale różną fluktuacją opóźnień. Jeśli wszystkie pakiety przesyłane przez sieć są dostarczane z jednakowym opóźnieniem  $D$ , sieć ma zerową fluktuację opóźnienia. Jeżeli jednak opóźnienie zmienia się w zakresie od  $D+\epsilon$  do  $D-\epsilon$ , średnie opóźnienie jest takie samo, jak w poprzednim przypadku, ale towarzyszy mu niezerowa fluktuacja.

Aby zrozumieć, dlaczego fluktuacja opóźnienia jest tak ważna, wystarczy przeanalizować proces przesyłania sygnału głosowego. Po stronie nadawczej sygnał analogowy podlega próbkowaniu i konwersji do formatu cyfrowego. W wyniku tej operacji co 125 μs generowana jest kolejna 8-bitowa wartość. Próbki są następnie zapisywane w pakiecie, który zostaje wysłany do sieci. Po stronie odbiorczej wartości cyfrowe zostają wyodrębnione z pakietu i przekształcone z powrotem w sygnał analogowy. Jeśli fluktuacja opóźnienia ma wartość zero (każdy pakiet dociera na drugą stronę po takim samym czasie), odtwarzany sygnał dźwiękowy dokładnie odpowiada sygnałowi oryginalnemu. W przeciwnym przypadku zostanie znieksztalcony. Eliminacja fluktuacji opóźnienia wymaga zastosowania jednego z dwóch rozwiązań:

- zaprojektowania sieci izochronicznej o zerowej wartości fluktuacji opóźnień,
- zastosowania protokołu, który uwzględnia występowanie fluktuacji opóźnień.

Tradycyjne systemy telefoniczne bazują na pierwszym rozwiążaniu. Operatorzy sieci telefonicznych budują **sieci izochroniczne**, które gwarantują jednakowe opóźnienie na wszystkich trasach. Zatem jeśli cyfrowa transmisja rozmowy telefonicznej wymaga wykorzystania dwóch tras, obydwie będą miały dokładnie takie samo opóźnienie.

Transmisja sekwencji wizyjnych i dźwięku w internecie bazuje na drugim z wymienionych rozwiązań. Mimo że w sieci transmisyjnej występuje istotna fluktuacja opóźnień, aplikacje audiowizualne bazują na **protokołach strumieniowania w czasie rzeczywistym**,

które zapobiegają występowaniu niepożądanych skutków tego problemu<sup>79</sup>. Ponieważ użycie specjalnych protokołów jest znacznie tańsze od budowania sieci izochronicznych, firmy telekomunikacyjne coraz częściej odstępują od zasady obowiązkowego zachowania izochroniczności systemu transmisyjnego. Oczywiście, działanie protokołu nie pozwala na skompensowanie wyjątkowo dużych wartości fluktuacji, które mogą doprowadzić do zniekształcenia sygnału wyjściowego. Dlatego nawet w przypadku stosowania drugiego rozwiązania dostawcy usług starają się minimalizować jitter we własnych sieciach.

## 28.7. Zależność między opóźnieniem a przepustowością

Teoretycznie opóźnienie i przepustowość sieci są od siebie niezależne. W praktyce jednak bywają powiązane. Przeanalizujmy ponownie przykład drogi między dwoma punktami. Jeśli samochody wjeżdżają na nią w równych odstępach czasu i przemieszczają się z jednakową prędkością, docierają do celu w regularnych interwałach. Jeśli jeden z samochodów zwolni z jakichkolwiek przyczyn, kolejne również będą musiały zwolnić, co w konsekwencji może doprowadzić do zatoru. Kierowcy samochodów wjeżdżających na drogę w czasie, gdy występuje na niej korek, zanotują znacznie większe opóźnienie niż osoby podróżujące w czasie, gdy droga nie była zatłoczona. Analogiczna sytuacja występuje w sieci. Jeśli router zapisuje odbierane pakiety w kolejce, w której są już przechowywane wcześniejsze datagramy, nadchodzące dane będą musiały oczekwać na wyemitowanie aż do czasu przetworzenia wcześniejszych. W przypadku przeciążenia sieci pakiety są opóźniane znacznie bardziej niż w czasie transmisji przez nieobciążoną sieć.

### 28.7.1. Stopień wykorzystania sieci jako estymata opóźnienia

Naukowcy zajmujący się sieciami badali zależność między opóźnieniem a zajętością sieci i ustalili, że w wielu przypadkach wartość opóźnienia można oszacować na podstawie stopnia wykorzystania pojemności łącznej w danej chwili. Jeśli opóźnienie w czasie braku obciążenia oznaczamy jako  $D_o$ , a bieżący **stopień wykorzystania** (ang. *utilization*)  $U$  będzie się zmieniał w przedziale od 0 do 1, efektywne opóźnienie można wyznaczyć za pomocą wzoru:

$$D = \frac{D_o}{(1 - U)} \quad (28.1)$$

W czasie braku jakiegokolwiek obciążenia sieci  $U$  ma wartość zerową, a efektywne opóźnienie odpowiada wartości  $D_o$ . Gdy system pracuje z połową swojego maksymalnego obciążenia, efektywne opóźnienie podwaja się. Natomiast gdy natężenie ruchu zbliża się do całkowitej pojemności sieci ( $U$  zbliża się do 1), opóźnienie dąży do nieskończoności. Mimo że przedstawione równanie jest jedynie sposobem na szacowanie efektywnego opóźnienia, można na jego podstawie stwierdzić, że:

---

<sup>79</sup> Transmisja danych czasu rzeczywistego w internecie jest tematem kolejnego rozdziału.

*Przepustowość i opóźnienie nie są całkowicie niezależne od siebie. Opóźnienie rośnie wraz ze wzrostem natężenia ruchu. Gdy ilość przesyłanych danych zbliża się do całkowitej pojemności sieci, opóźnienie osiąga bardzo duże wartości.*

Administratorzy sieci są świadomi faktu, że duże obciążenie systemu może powodować ogromne opóźnienia. Dlatego większość z nich stara się utrzymać natężenie ruchu na niskim poziomie i nieustannie monitoruje stan sieci. Gdy tylko poziom średniego lub szczytowego wykorzystania sieci przekroczy pewną wartość progową, administrator zwiększa pojemność systemu. Na przykład jeśli obciążenie sieci Ethernet o przepustowości 100 Mb/s utrzymuje się na wysokim poziomie, należałoby pomyśleć o wdrożeniu gigabitowego Ethernetu. Można również podzielić sieć na dwie części, umieszcając w każdej z nich połowę w wcześniejszych komputerów (taka operacja nie nastręcza żadnych trudności, jeśli administrator dysponuje przełącznikami z funkcją VLAN).

Jaki poziom wykorzystania sieci należy uznać za progowy? Na to pytanie nie ma prostej odpowiedzi. Wielu administratorów wybiera bezpieczne wartości. Na przykład jeden z większych dostawców usług internetowych, obsługujący sieć szkieletową, utrzymuje na wszystkich obwodach cyfrowych obciążenie poniżej 50% wartości maksymalnej. Inne firmy, ze względów finansowych, wyznaczają ten próg na poziomie 80%. Większość administratorów zgodzi się jednak z twierdzeniem, że sieć nie powinna pracować z obciążeniem przekraczającym 90% jej pojemności.

### 28.7.2. Iloczyn opóźnienia i przepustowości

Znając opóźnienie i przepustowość sieci, można wyliczyć inną przydatną wartość — **iloczyn opóźnienia i przepustowości**<sup>80</sup>. Znaczenie iloczynu znowu najłatwiej można wyjaśnić przez analogię do drogi. Jeśli samochody wjeżdżają na nią z określona częstotliwością  $T$  aut na sekundę, a przejazd jednego samochodu zajmuje  $D$  sekund, wówczas  $T \times D$  aut może wjechać na drogę, zanim pierwszy samochód zakończy przejazd. W każdej chwili na drodze może się więc znaleźć  $T \times D$  aut. W przypadku sieci oznacza to, że w dowolnie wybranym momencie w sieci przesyłana jest taka liczba bitów, jaka wynika z równania:

$$\text{Bity w sieci} = T \times D \quad (28.2)$$

Czynnik  $D$  odpowiada opóźnieniu mierzonymu w sekundach, a  $T$  reprezentuje przepustowość wyrażaną w bitach na sekundę. Podsumowując:

*Iloczyn opóźnienia i przepustowości wyznacza maksymalną ilość danych przesyłanych w sieci. Sieć o przepustowości  $T$  oraz opóźnieniu  $D$  może w dowolnie wybranym momencie transportować  $T \times D$  bitów.*

<sup>80</sup> W przypadku zastosowania tej miary do opisu warstwy sprzętowej często nazywa się ją **iloczynem opóźnienia i szerokości pasma**.

Wartość iloczynu jest bardzo ważna w sieciach o szczególnie dużym opóźnieniu lub dużej przepustowości, ponieważ ma wpływ na transmisję danych — aplikacja nadawcza może wysłać spory blok danych, zanim stacja odbiorcza zarejestruje pierwszy bit.

## 28.8. Pomiar opóźnienia, przepustowości i fluktuacji opóźnienia

Techniki pomiaru przepustowości i fluktuacji opóźnienia nie należą do szczególnie skomplikowanych. Aby ustalić przepustowość, nadawca wysyła duży blok danych. Odbiorca rejestruje czas od odebrania pierwszego bitu do zakończenia przekazu, a następnie oblicza przepustowość, dzieląc ilość danych przez wartość czasu. Technika pomiaru fluktuacji opóźnienia polega na wysyłaniu serii pakietów z niewielkim stałym opóźnieniem między nimi. Odbiorca rejestruje czas nadchodzenia pakietów, a następnie na podstawie zbioru takich zapisanych wartości oblicza różnice w opóźnieniu.

W przeciwieństwie do przepustowości i fluktuacji opóźnienia precyzyjne obliczenie opóźnienia na trasie ze stacji A do stacji B wymaga wcześniejszego zsynchronizowania zegarów. Ponadto przy pomiarze na krótkich odcinkach (na przykład w sieci LAN) niezbędne są zegary o bardzo dużej precyzji. Aby uniknąć wstępnej synchronizacji, wiele narzędzi pomiarowych wyznacza czas przejścia pakietu w dwie strony i dzieli wynik pomiaru przez dwa. Do wykonania tego zadania można się posłużyć choćby poleceniem ping.

Pomiar wydajności sieci okazuje się bardzo trudny z czterech powodów:

- Trasy bywają asymetryczne.
- Warunki transmisji zmieniają się w sposób gwałtowny.
- Zastosowana technika pomiarowa może mieć wpływ na uzyskiwane wyniki.
- Ruch ma formę zbitek danych.

Z pierwszego punktu wynika, dlaczego do szacowania opóźnień nie można wykorzystywać czasu transmisji pakietu w dwie strony. Asymetryczny routing oznacza, że opóźnienia na trasie z komputera B do komputera A mogą istotnie odbiegać od opóźnień występujących na trasie z A do B. Zwykły podział zmierzonej wartości nie oddaje rzeczywistego stanu sieci.

Drugie stwierdzenie wyjaśnia, dlaczego dokładny pomiar wydajności sieci jest tak trudny do przeprowadzenia — warunki zmieniają się w sposób gwałtowny. Jako przykład przeanalizujmy sieć współdzieloną. Jeśli tylko jedna stacja wysyła dane, z pewnością zarejestruje małe opóźnienie, wysoką przepustowość i małą fluktuację opóźnienia. Jednak gdy również inne komputery rozpoczęną korzystanie z sieci, stopień zajętości łączy wzrośnie, a to z kolei spowoduje zwiększenie opóźnienia i fluktuacji opóźnienia oraz zmniejszenie przepustowości. Ponadto zmiana warunków następuje bardzo szybko, przez co wartość opóźnienia może się istotnie zmienić w czasie krótszym niż sekunda. Zatem nawet wykonywanie pomiarów co dziesięć sekund nie gwarantuje zarejestrowania wszystkich istotnych zmian w wydajności systemu.

Z trzeciego punktu wynika, że generowanie ruchu testowego (potrzebnego do oszacowania parametrów sieci) może mieć wpływ na wydajność. W laboratorium testowym PlanetLab zaobserwowano kiedyś, że tak wielu badaczy korzystało z polecenia ping, że

generowany przez to ruch przeważał nad innym ruchem sieciowym. Problem okazał się na tyle poważny, że administratorzy systemu zdefiniowali politykę stosowania polecenia `ping`.

Czwarty punkt ma kluczowe znaczenie dla problemu — sieci komputerowe przenoszą ruch **zbitkowy**. Oznacza to, że ruch nie jest równomierny. Jeśli zastanowimy się nad sposobem pracy komputera, takie stwierdzenie okaże się oczywiste. Większość stacji pozostałe nieaktywna do czasu, aż użytkownik uruchomi aplikację komunikującą się z innymi jednostkami w internecie. Na przykład wpisanie w przeglądarce adresu URL powoduje, że aplikacja pobiera kolejne elementy strony, po czym komunikacja ustaje, aż do czasu przejścia do następnej strony. W podobny sposób pobierana jest poczta elektroniczna. Komputer komunikuje się z systemem pocztowym, pobiera wiadomości, a następnie oczekuje na reakcję użytkownika.

Co ciekawe, sumaryczny ruch również ma charakter zbitkowy. Można by sądzić, że skoro generowanie paczek danych jest zjawiskiem lokalnym, to praca milionów stacji internetowych powinna spowodować wygładzenie charakterystyk ruchu. Przecież nie wszyscy użytkownicy odbierają pocztę w tym samym czasie, więc gdy jeden czyta list, inny mógłby pobierać swoje wiadomości. Pomiary parametrów sieci telefonicznych rzeczywiście wykazały, że natężenie ruchu generowanego przez miliony rozmówców zmienia się w sposób łagodny. Jednak w internecie takie twierdzenie się nie sprawdza. Sumaryczny ruch internetowy zmienia się dość gwałtownie, co objawia się na wykresie wieloma szczytami i dolinami przebiegu. Osoby zajmujące się statystyczną analizą ruchu twierdzą, że jest on **samopodobny**, czyli że ma przebieg analogiczny do **fraktali** — niezależnie od gradacji pomiaru profil statystyczny pozostaje jednakowy. Jeśli więc ruch generowany przez stację lokalną ma formę zbitek pakietów, również dostawca usług internetowych będzie rejestrował zmienne natężenie ruchu, niezależnie od tego, czy pochodzi on od tysięcy, czy milionów użytkowników. Poziomy natężenia będą znacznie wyższe niż w sieci lokalnej, ale ogólna charakterystyka będzie podobna.

Podsumowując:

*W przeciwieństwie do ruchu telefonicznego, wymiana danych ma charakter zbitkowy. Ruch w transmisji danych jest samopodobny, ponieważ kształt przebiegu natężenia jest taki sam, niezależnie od stopnia agregacji ruchu.*

## 28.9. Pomiar pasywny, małe pakiety i mechanizm NetFlow

Osoby zajmujące się pomiarami parametrów sieci dzielą techniki pomiarowe na:

- aktywne,
- pasywne.

Wady technik **aktywnego** pomiaru zostały opisane w poprzednim punkcie — wprowadzenie dodatkowego ruchu do sieci powoduje, że pomiar może spowodować zmianę wydajności sieci. Jedyną alternatywą jest więc pomiar **pasywny**, czyli monitorowanie sieci

i zliczanie pakietów, ale bez generowania dodatkowego obciążenia. Na przykład oszacowanie stopnia wykorzystania łącza dostawcy usług internetowych sprowadza się do zliczenia pakietów przesyłanych w tym łączu w określonym czasie. Firmy ISP korzystają ze stacji monitorujących, które zliczają bajty we wszystkich pakietach transmitowanych w kolejnych interwałach.

Pomiar może odnosić się zarówno do liczby pakietów, jak i liczby bajtów danych. Stopień wykorzystania łącza jest definiowany jako wartość procentowa pojemności, a pojemność jest określana liczbą bitów na sekundę. Dostawca usług internetowych musi więc rejestrować liczbę bitów danych przesyłanych w ciągu sekundy. Z drugiej strony, wydajność przełączników i routerów określa się za pomocą liczby pakietów na sekundę. Przyczyną jest to, że zadaniem wspomnianych urządzeń jest odnajdywanie tras dla całych pakietów, a ich obciążenie obliczeniowe jest proporcjonalne do liczby pakietów, a nie liczby bitów w pakiecie. Jeśli strumień danych ma przepływność 1 Gb/s i składa się z pakietów o dużym rozmiarze, przełącznik musi wykonać mniej operacji niż w przypadku, gdy strumień jest podzielony na wiele pakietów o małym rozmiarze. Dostawcy sprzętu znają, oczywiście, tę zależność i dlatego niekiedy definiują wydajność swoich urządzeń w bitach na sekundę, a nie w pakietach na sekundę (tj. podają parametry wydajnościowe właściwe dla ruchu pakietów o dużych rozmiarach).

Zagadnienie to można podsumować w następujący sposób:

*Aby oszacować stopień wykorzystania łącza, dostawcy usług internetowych rejestrują całkowitą ilość danych przesyłanych danym łączem w jednostce czasu. Z kolei w celu określenia wpływu ruchu na obciążenie routera lub przełącznika zliczają pakiety przekazane w określonym czasie.*

Najczęściej wykorzystywana technika pomiaru pasywnego została opracowana przez firmę Cisco i jest obecnie standardem IETF o nazwie **NetFlow**. Routery wyposażone w funkcję NetFlow próbkują pakiety zgodnie z parametrami określonymi przez administratora sieci (na przykład pobierają jeden pakiet na tysiąc). Odczytują informacje zawarte w nagłówku, agregując je i w uogólnionej postaci przekazując do systemu zarządzania siecią, gdzie dane są poddawane dalszemu przetwarzaniu (które często sprowadza się do zapisania ich na dysku z przeznaczeniem do późniejszej analizy). Zazwyczaj mechanizm NetFlow wyodrębnia źródłowy i docelowy adres IP, informację o typie datagramu oraz numery portów protokołu. Pasywny sposób działania wymaga, aby informacje gromadzone przez moduł NetFlow były wysyłane do stacji zarządzającej za pośrednictwem osobnego portu, a nie w ramach tej samej sieci, w której przekazywane są dane użytkowników.

## 28.10. Jakość usługi (QoS)

Działaniem towarzyszącym pomiarowi parametrów sieci jest **rezerwacja zasobów sieciowych** (ang. *network provisioning*), czyli takie projektowanie sieci, aby możliwe było zagwarantowanie właściwego poziomu obsługi ruchu. W pozostałej części rozdziału opisano

mechanizmy, które mają na celu zapewnienie odpowiedniej **jakości usługi** (QoS — ang. *Quality of Service*).

Parametr QoS jest elementem kontraktu zawieranego między dostawcą usług i klientem. W najprostszym przypadku definiuje gwarantowaną przez dostawcę szybkość transmisji danych. Na przykład firma telekomunikacyjna oferująca łączą DSL może gwarantować przepustowość 2,2 Mb/s. W bardziej szczegółowych kontraktach często opisuje się **usługi wielopoziomowe** (ang. *tiered services*), w których rodzaj realizowanej usługi zależy od wysokości opłat. Przykładami takiego rozwiązania są systemy **priorytetowania**, bazujące na założeniu, że pakiety klientów, którzy wykupili wyższy poziom usługi, są obsługiwane przed pakietami klientów opłacających usługi niższego poziomu.

Duże korporacje zazwyczaj domagają się bardziej precyzyjnych **gwarancji realizacji usługi**. W branży finansowej typowe jest uwzględnianie w kontraktach dopuszczalnej wartości opóźnienia w transmisji między dwoma punktami. Firma brokerska mogłaby na przykład zażądać wpisania do kontraktu, że pakiety wygenerowane w sieci firmowej będą dostarczane do systemów Giełdy Papierów Wartościowych w Warszawie w czasie krótszym niż 10 milisekund. Z kolei firma wykonująca każdej nocy kopię zapasową swojego centrum danych mogłaby zażądać gwarancji, że przepustowość połączeń TCP nie spadnie poniżej 1 Gb/s.

## 28.11. Ogólna i szczegółowa specyfikacja QoS

W jaki sposób dostawcy usług definiują parametry QoS i jakich technologii używają do zagwarantowania odpowiedniego współczynnika QoS? W tabeli 28.3 wymieniono dwie ogólne propozycje rozwiązań odnoszące się do problemu zapewnienia odpowiedniej jakości usługi. Jak wynika z zestawienia, różnią się one stopniem szczegółowości oraz tym, kto definiuje parametry (dostawca czy klient).

Tabela 28.3. Dwie propozycje sposobu definiowania jakości usług

Rozwiązanie	Opis
Szczegółowa specyfikacja (ang. <i>fine-grain</i> )	Dostawca usług umożliwia klientowi zdefiniowanie parametrów QoS dla określonego połączenia. Klient generuje żądanie podczas każdorazowego ustanawiania połączenia (na przykład przed nawiązaniem komunikacji TCP).
Ogólna specyfikacja (ang. <i>coarse-grain</i> )	Dostawca definiuje kilka ogólnych klas jakości usługi właściwych dla określonych rodzajów ruchu. Ruch generowany przez klienta musi zostać dostosowany do parametrów danej klasy.

### 28.11.1. Szczegółowa specyfikacja QoS i przepływ danych

Większość dotychczasowych opracowań na temat technik QoS jest efektem prac firm telekomunikacyjnych. Projektanci systemów zakładają, że usługi są realizowane zgodnie z modelem transmisji połączeniowych, tak jak w przypadku systemów telefonicznych —

wymiana danych między stacją klienta a zdalnym ośrodkiem (na przykład serwerem WWW) wymaga ustanowienia połączenia. Ponadto zakłada się, że klient będzie określił parametry QoS w odniesieniu do każdego połączenia niezależnie. Z kolei dostawca usług naliczy opłatę w zależności od odległości między stacjami oraz zastosowanych parametrów QoS.

Wiele funkcji QoS zostało uwzględnionych przez firmy telekomunikacyjne w projekcie standardu ATP. Choć ostatecznie technologia ATM nie zyskała spodziewanego zainteresowania, a dostawcy usług nie naliczają opłat za każde połączenie, niektóre pojęcia szczegółowej specyfikacji QoS są nadal wykorzystywane w nieznacznie zmienionych wersjach. Zamiast wyznaczania parametrów QoS q w odniesieniu do połączenia, definiuje się je dla strumienia (ang. *flow*) danych. Strumień zazwyczaj opisuje przepływ danych na poziomie warstwy transportowej, czyli połączenia TCP lub zbioru pakietów UDP przekazywanych między dwoma aplikacjami (bądź generowanych w ramach rozmowy VoIP). W tabeli 28.4 przedstawiono cztery główne kategorie usług standardu ATM wraz z opisem ich powiązań ze strumieniami danych.

Tabela 28.4. Cztery główne kategorie parametrów QoS

Skrót	Rozwiniecie	Znaczenie
CBR	<i>Constant Bit Rate</i> (stała przepływność bitowa)	Dane zawarte w strumieniu są przesyłane z ustaloną szybkością. Na przykład rozmowa telefoniczna zawsze wymaga zagwarantowania przepływności 64 kb/s.
VBR	<i>Variable Bit Rate</i> (zmienna przepływność bitowa)	Dane zawarte w strumieniu są przesyłane ze zmienną szybkością, ale w ustalonych statystycznie granicach.
ABR	<i>Available Bit Rate</i> (dostępna przepływność bitowa)	Do przenoszenia danych wykorzystywana jest cała dostępna w danej chwili przepustowość sieci.
UBR	<i>Unspecified Bit Rate</i> (niezdefiniowana przepływność bitowa)	Dane są przenoszone w sieci bez określania ich przepływności. W działaniu aplikacji wykorzystywana jest usługa typu best-effort.

Jak wynika z zestawienia, usługa CBR jest odpowiednia do przesyłania danych ze stałą przepływnością. Doskonałym przykładem jej zastosowania jest, oczywiście, cyfrowa transmisja głosu. Rozwiązania VBR znajdują zastosowanie w przekazach, w których wykorzystuje się kodowanie o zmiennej przepływności bitowej. Część koderów wideo bazuje na technice kodowania różnicowego, co oznacza, że ilość danych potrzebnych do odzwierciedlenia jednej klatki filmu jest proporcjonalna do różnicy między bieżącą klatką a obrazem wcześniejszym. W takich przypadkach klient może określić średnią szybkość transmisji danych, a także maksymalną przepływność bitową i czas utrzymywania się maksymalnej przepływności. Wybierając usługę VBR, trzeba określić:

- trwałą przepływność bitową (SBR — ang. *Sustained Bit Rate*);
- szczytową przepływność bitową (PBR — ang. *Peak Bit Rate*);

- trwały rozmiar zbitki (SBS — ang. *Sustained Burst Size*);
- szczytowy rozmiar zbitki (PBS — ang. *Peak Burst Size*).

Usługa ABR umożliwia współdzielenie zasobów. Klient płaci za taką część usługi, jaką w danej chwili można zapewnić. Jeśli inni klienci przesyłają dane w tym samym czasie, jakość usługi jest niższa (co zazwyczaj oznacza również niższe opłaty). Ostatnie rozwiązanie (UBR) znajduje zastosowanie w przypadkach, w których klient nie chce płacić za wyższą przepustowość i jest zadowolony z usługi realizowanej na zasadzie best-effort.

Pierwszym rozwiażaniem na temat zasadności wprowadzania mechanizmów QoS do rozwiązań internetowych towarzyszyło twierdzenie firm telekomunikacyjnych, że szczegółowa specyfikacja jakości usług jest niezbędna do realizacji połączeń głosowych. W rezultacie oprócz prac nad standardem ATM rozpoczęto badania nad wdrożeniem systemu **zintegrowanych usług** (IntServ — ang. *Integrated Services*).

### 28.11.2. Ogólna specyfikacja QoS i klasy usług

Alternatywą dla szczegółowej specyfikacji QoS jest ogólna specyfikacja, w której ruch podlega podziałowi na **klasy**, a parametry QoS są definiowane w odniesieniu do klas, a nie strumieni. Aby uświadomić sobie potrzebę definiowania ogólnej specyfikacji QoS, trzeba najpierw przeanalizować implementację mechanizmów QoS w routerze rdzeniowym. Interfejsy routera pracują z przepustowością 10 Gb/s, co oznacza, że pakiety nadchodzą z bardzo dużą częstotliwością. Do ich przekazywania potrzebne są specjalne urządzenia, ponieważ klasyczny procesor nie jest dostatecznie wydajny. Poza tym routery tego typu przenoszą ruch od głównych firm ISP, który obejmuje miliony jednoczesnych strumieni. Zapewnienie odpowiednich parametrów QoS wymaga dostępności dodatkowych zasobów sprzętowych urządzenia. Router musi utrzymywać informacje o stanie milionów strumieni i wykonywać skomplikowane operacje w odniesieniu do każdego przekazywanego pakietu, a jak wiadomo, odwołania do pamięci spowalniają pracę urządzenia. Trzeba również uwzględnić czas potrzebny na zaalokowanie zasobów nowego strumienia oraz zwolnienie ich po zakończeniu transmisji danego strumienia.

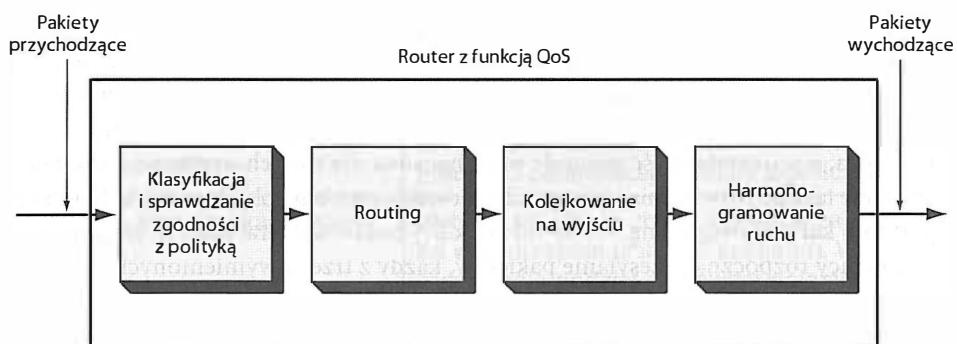
Po wielu latach badań nad zintegrowanymi usługami i opracowaniu kilku protokołów społeczność naukowa doszła do wniosku, że szczegółowa specyfikacja QoS jest niepraktyczna i niepotrzebna. Zwykły użytkownik nie będzie miał dostatecznie dużej wiedzy, aby właściwie dobrać parametry usługi (bo jacy wymagania odnośnie przepustowości należałoby zdefiniować, aby pobrać typową stronę internetową?). Z drugiej strony, routery rdzeniowe nie są dość wydajne, aby implementować w nich mechanizmy QoS operujące strumieniami danych.

Z tego powodu większość rozwiązań QoS koncentruje się na wyznaczaniu kilku ogólnych klas usługi, a nie na zapewnianiu określonej charakterystyki ruchu na połączeniach między dwoma punktami końcowymi.

*Mimo wieloletnich prac badawczych i standaryzacyjnych zastosowanie szczegółowej specyfikacji QoS ogranicza się do kilku szczególnych przypadków transmisji danych.*

## 28.12. Implementacja mechanizmów QoS

Na rysunku 28.1 przedstawiono cztery etapy przetwarzania pakietów w routerze lub przełączniku zawierającym funkcję QoS.



Rysunek 28.1. Cztery etapy przetwarzania pakietów w mechanizmie QoS

**Klasyfikacja i sprawdzanie zgodności z polityką.** Dobierane przez router pakietы są **klasyfikowane** przez przypisywanie im identyfikatorów strumienia. W przypadku systemu ze szczegółową specyfikacją QoS identyfikator odpowiada jednemu połączeniu. W systemach ogólnych wyznacza on klasę ruchu. Następna operacja polega na **sprawdzeniu zgodności pakietu z polityką**. Router sprawdza wówczas, czy odebrany pakiet jest zgodny z parametrami danego strumienia. Na przykład jeśli klient generuje dane z większą częstotliwością, niż przewidziano w kontrakcie, router odrzuca pakiet. Jedną z technik wymuszania zdefiniowanej polityki jest zastosowanie algorytmu **wczesnego losowego odrzucania** (RED — ang. *Random Early Discard*), który usuwa pakiet zgodnie z pewną funkcją statystyczną. Prawdopodobieństwo odrzucenia wynika z długości kolejki, jaka została przygotowana do obsługi danego strumienia. Gdy kolejka jest całkowicie zapełniona, prawdopodobieństwo usunięcia pakietu wynosi 1. W pozostałych przypadkach wartość prawdopodobieństwa jest liniowo zależna od liczby pakietów zapisanych w buforze. Wykorzystanie algorytmu RED pozwala na uniknięcie problemu **odrzucania ogona** (ang. *tail drop*), który powoduje usuwanie wszystkich pakietów, które nadeszły po zapełnieniu kolejki. Wiele sesji TCP rozpoczęta wówczas procedurę powolnego startu i stopniowe zwiększenie natężenia ruchu, aż do ponownego zapełnienia kolejki i rozpoczęcia nowego cyklu.

**Routing.** Identyfikator strumienia może zostać wykorzystany również w fazie wyznaczania kolejnego węzła na trasie pakietu. W niektórych przypadkach trasa zależy wyłącznie od identyfikatora (na przykład urządzenie może zapisać informację o tym, że cały ruch głosowy należy przekazywać wybranym portem do przełącznika telefonii internetowej). W innych jest on ignorowany, a o wyborze kolejnego węzła decyduje adres docelowy zapisany w pakiecie. Dokładne działanie opisywanego bloku zależy od przeznaczenia określonego przełącznika lub routera, a także od parametrów polityki QoS.

**Kolejkowanie na wyjście.** Większość mechanizmów QoS wykorzystuje zbiory kolejek powiązane z portami wyjściowymi. Po wybraniu w module routingu odpowiedniego portu

pakiet (na podstawie identyfikatora) zostaje zapisany w jednym z buforów skojarzonych z tym portem. W systemach ogólnej specyfikacji QoS jedna kolejka odpowiada jednej klasie ruchu. Jeśli więc administrator zdefiniuje osiem klas ruchu, do każdego portu zostanie przypisanych osiem buforów pakietów. W systemach szczegółowej specyfikacji QoS zazwyczaj każdemu połączeniu odpowiada oddzielną kolejkę. Jeden z procesorów sieciowych umożliwia wykorzystanie 256 000 kolejek o hierarchicznej organizacji.

**Harmonogramowanie ruchu.** Stosowanie polityki QoS w module harmonogramowania ruchu polega na wybieraniu pakietów do wysłania przez niezajęty port. Administrator sieci może na przykład ustalić, że trzech klientów otrzyma po 25% całkowitej pojemności łączą, a pozostała część zostanie przeznaczona dla innych użytkowników sieci. Wdrożenie takiego rozwiązania wymaga zdefiniowania czterech kolejek oraz uruchomienia **algorytmu karuzelowego** (ang. *round-robin*), który będzie wybierał pakiety. Jeśli wszyscy użytkownicy rozpoczną przesyłanie pakietów, każdy z trzech wymienionych klientów otrzyma kwartę pojemności łączą.

Możliwe jest również wdrożenie techniki proporcjonalnego podziału zasobów sieci, ale wymaga to zaimplementowania bardziej skomplikowanych algorytmów selekcji. Większa złożoność rozwiązania wynika z tego, że moduł harmonogramowania musi stosować długookresowe parametry wynikające z polityki, mimo przenoszenia ruchu zbitkowego. Router musi poprawnie działać w sytuacjach, w których chwilowa ilość danych przekracza rozmiar pamięci przydzielonej buforowi, ale długookresowa średnia przepływność bitowa mieści się w wyznaczonych granicach. Analogicznie, mechanizm zarządzania pracą modułu powinien uwzględnić fakt, że kolejki będą okresowo opróżniane i będzie można wykorzystać nieużywaną pamięć w innych kolejkach.

Opracowano i przeanalizowano wiele algorytmów harmonogramowania. Nie istnieje jednak taki, który można by uznać za idealny. Każdy z nich jest pewnym kompromisem między zgodnością z założeniami i narzutem obliczeniowym. W tabeli 28.5 wymieniono wszystkie z zaproponowanych algorytmów zarządzania ruchem.

### 28.13. Internetowe technologie QoS

Organizacja IETF opracowała kilka technologii i protokołów związanych z mechanizmami QoS. Trzy najważniejsze z nich to:

- RSVP i COPS,
- DiffServ,
- MPLS.

**RSVP i COPS.** W czasie prac nad systemem IntServ organizacja IETF opracowała dwa protokoły przeznaczone do wykorzystania w mechanizmach QoS. Są to **protokół rezerwacji zasobów** (RSVP — ang. *Resource ReSerVation Protocol*) oraz **otwarty protokół informowania o politykach** (COPS — ang. *Common Open Policy Services*). Standard RSVP znajduje zastosowanie w systemach o szczegółowej specyfikacji QoS, gdyż jest wykorzystywany w ustalaniu każdej sesji TCP lub UDP. Działanie mechanizmu RSVP rozpoczyna się od tego, że aplikacja wysyła żądanie, w którym określa parametry QoS.

Tabela 28.5. Przykłady algorytmów harmonogramowania ruchu

Algorytm	Opis
Algorytm cieknącego wiadra	Umożliwia wysyłanie pakietów z kolejki w określonych interwałach przez okresowe zwiększanie licznika pakietów i uzależnienie transmisji od bieżącej wartości tego licznika.
Algorytm wiadra z żetonami	Umożliwia wysyłanie danych z kolejki w określonych interwałach przez okresowe zwiększanie licznika bajtów i uzależnienie transmisji od bieżącej wartości tego licznika.
Ważony algorytm karuzelowy	Wybiera pakiety ze zbioru kolejek w zależności od wag przypisanych tym kolejkom. Wagi wyznaczają podział całkowitej pojemności łączna na odpowiednie wartości procentowe. W działaniu mechanizmu zakłada się stałą długość pakietu.
Deficytowy algorytm karuzelowy	Odmiana ważonego algorytmu karuzelowego, w której przetwarzaniu podlegają bajty, a nie całe pakiety. Dopuszcza się również chwilowy deficyt pojemności powodowany przez pakiety o dużym rozmiarze.

Każdy router na trasie między jednostką źródłową i docelową rezerwuje wskazane zasoby, a następnie przekazuje żądanie do następnego routera. Ostatnią jednostką, która musi się zgodzić na proponowane warunki, jest komputer docelowy. Jeśli wszystkie urządzenia na trasie pakietów są gotowe do obsługi ruchu z podanymi parametrami, generowany jest identyfikator strumienia, który jest następnie przekazywany do jednostki źródłowej. Od tego momentu można przesyłać dane wzduż zarezerwowanej trasy. Protokołowi RSVP towarzyszy protokół COPS, który odpowiada za ustalanie i wdrażanie polityki przenoszenia ruchu. Routery obsługujące ten protokół komunikują się z serwerem polityk, aby pobrać parametry danego strumienia danych. System RSVP nie jest jednak często stosowany, ponieważ należy do grupy mechanizmów, które odnoszą się do każdego strumienia oddzielnie.

**DiffServ.** Po wstrzynaniu prac nad systemami IntServ organizacja IETF skoncentrowała się na programie **zróżnicowanych usług** (DiffServ — ang. *Differentiated Services*), który obejmuje mechanizmy ogólnych specyfikacji QoS. W wyniku tych działań powstał dokument opisujący zasady definiowania klas QoS oraz wykorzystywania pola *rodzaj usługi* w nagłówkach pakietów IPv4 i IPv6. Mimo że wielu dostawców usług internetowych testowało rozwiązania DiffServ, technologia ta nie spotkała się z powszechną akceptacją.

**MPLS.** Technika **wieloprotokołowego przełączania etykiet** (MPLS — ang. *Multi-protocol Label Switching*) została opisana w rozdziale 19. jako połączeniowy mechanizm komunikacyjny zbudowany na bazie protokołu IP. Jego działanie wymaga wstępnego skonfigurowania tras w routerach zgodnych ze standardem MPLS. Komputer pracujący na jednym końcu trasy wprowadza do sieci pakiety z nagłówkiem MPLS, a stacja działająca

po drugiej stronie połączenia usuwa nagłówek MPLS i dostarcza dane do odbiorcy. W wielu przypadkach trasom MPLS przypisuje się pewną politykę, która powoduje, że podczas wysyłania pakietów przez określony port są uwzględniane parametry polityki QoS. Dzięki temu dostawca usług internetowych może oddzielić trasę przeznaczoną na połączenia głośowe od tras innego ruchu.

## 28.14. Podsumowanie

Dwie podstawowe miary wydajności sieci to opóźnienie (czyli czas potrzebny na przekazanie bitu z jednego komputera do drugiego) i przepustowość (czyli liczba bitów, które można wprowadzić do sieci w ciągu jednej sekundy). Mimo że przepustowość jest często określana jako szybkość transmisji, jest ona miarą pojemności sieci. Iloczyn przepustowości i opóźnienia służy do określania ilości danych transmitowanych w wybranym momencie. Opóźnienie i przepustowość są od siebie zależne. Gdy przepustowość zbliża się do 100% pojemności, opóźnienie gwałtownie rośnie.

Coraz większe znaczenie w transmisji danych ma fluktuacja opóźnienia (jitter). Niską wartość fluktuacji zapewniają sieci izochroniczne oraz protokoły przeznaczone do transmisji w czasie rzeczywistym głosu i sekwencji wizyjnych. W internecie stosowane jest rozwiązanie bazujące na wspomnianym protokole.

Pomiar wydajności sieci bywa bardzo trudnym zadaniem. Aby oszacować opóźnienie w sieciach o asymetrycznych trasach, konieczne jest wstępne zsynchronizowanie zegarów obydwu stacji końcowych. Nierównomierne natężenie ruchu sprawia, że wydajność zmienia się gwałtownie w krótkich odstępach czasu. Dodatkowo generowany w czasie pomiaru ruch zmienia parametry sieci. Dlatego wielu administratorów korzysta z rozwiązań pasywnych, takich jak NetFlow.

Omówione zostały dwie specyfikacje QoS — szczegółowa i ogólna. Rozwiązań implementujących szczegółowe polityki QoS nie są stosowane praktycznie, choć nadal w użyciu są akronimy kategorii usług definiowanych przez protokół ATM (CBR, VBR, ABR i UBR).

Przełączniki i routery działające zgodnie z mechanizmem QoS klasyfikują nadchodzące dane, weryfikują je na podstawie polityki QoS, przekazują do odpowiednich interfejsów, zapisują w kolejkach wyjściowych i wybierają zgodnie z odpowiednim algorytmem w chwili zwolnienia portu wyjściowego. Wybór pakietów należy do zadań kilku przedstawionych algorytmów. Każdy z nich jest kompromisem między zgodnością z polityką a narzutem obliczeniowym.

Organizacja IETF opracowała protokoły RSVP i COPS, wchodzące w skład specyfikacji IntServ. Jednak prace nad mechanizmami szczegółowego definiowania parametrów QoS zostały zastąpione projektem DiffServ. Ponadto organizacja IETF przygotowała standard MPLS, który umożliwia przypisywanie określonych parametrów QoS pakietom sklasyfikowanym jako należące do określonego tunelu MPLS.

## ZADANIA

- 28.1. Wymień i opisz trzy podstawowe miary wydajności sieci.
- 28.2. Wymień i opisz pięć rodzajów opóźnień.
- 28.3. Czy opóźnienia w dostępie do medium są większe w sieciach LAN, czy WAN? A opóźnienia kolejkowania? Uzasadnij odpowiedź.
- 28.4. W jaki sposób można zmierzyć przepustowość?
- 28.5. Jakiego terminu używa się do określenia przepustowości, aby był on zrozumiały dla użytkownika?
- 28.6. Podaj przykłady przetwarzania danych, w których efektywna wydajność transmisji jest mniejsza od pojemności kanału.
- 28.7. Wyjaśnij opóźnienie i przepustowość na przykładzie transmisji bitów.
- 28.8. Który parametr (opóźnienie czy przepustowość) w największym stopniu ogranicza wydajność sieci?
- 28.9. Użyj polecenia ping do ustalenia opóźnienia w komunikacji z lokalnymi i zdalnymi jednostkami. Sprawdź, jaką najmniejszą i największą wartość opóźnienia w internecie możesz zarejestrować.
- 28.10. Podczas korzystania z polecenia ping w odniesieniu do adresu 127.0.0.1 opóźnienie jest bardzo małe. Wyjaśnij dlaczego.
- 28.11. Pobierz z internetu program ttcp i wykorzystaj go do pomiaru przepustowości lokalnej sieci Ethernet. Jaka jest efektywna wydajność transmisji? Oszacuj stopień wykorzystania łącza.
- 28.12. Porównaj przepustowości sieci 100 Mb/s i 1 Gb/s.
- 28.13. Czym jest fluktuacja opóźnienia i jakie dwie techniki można zastosować do ograniczenia skutków fluktuacji opóźnienia?
- 28.14. Osoby zajmujące się sieciami często mówią o zagięciu charakterystyki opóźnienia. Aby sprawdzić, co to oznacza, sporządź wykres efektywnego opóźnienia odpowiadającego poziomowi wykorzystania łącza w przedziale od 0 do 0,95. Czy można wskazać taką wartość stopnia wykorzystania łącza, od której krzywa zaczyna gwałtownie piąć się w górę?
- 28.15. Oblicz, ile danych jest „w locie” pomiędzy naziemną stacją nadawczą, satelitą i naziemną stacją odbiorczą. Wyznacz iloczyn opóźnienia i przepustowości w sieciach satelitarnych o przepustowości 3 Mb/s. Przyjmij założenie, że orbita satelity znajduje się 32 000 kilometrów nad Ziemią, a fale radiowe przemieszczają się z prędkością światła.
- 28.16. Dlaczego pomiar wydajności sieci jest trudny?
- 28.17. Czy transmisja danych różni się od transmisji głosu?
- 28.18. Wyjaśnij, dlaczego firmy ISP zliczają pakiety odbierane w jednostce czasu, a nie bajty?
- 28.19. Wymień dwa rodzaje mechanizmów QoS.
- 28.20. Oszacuj obciążenie obliczeniowe wynikające z zastosowania szczegółowego mechanizmu QoS w rdzeniu internetu. Przyjmij założenie, że w łączu o przepustowości 10 Gb/s przesyłane są pakiety o rozmiarze 1000 bajtów, a do przetworzenia każdego z nich potrzeba N operacji arytmetycznych. Oblicz, ile operacji procesor musi wykonać w każdej sekundzie.
- 28.21. Wymień cztery kategorie QoS zdefiniowane w standardzie ATM. Opisz każdą z nich.

- 28.22. Jaki rodzaj mechanizmu QoS będzie odpowiedni do pobierania stron internetowych za pomocą przeglądarki? Uzasadnij odpowiedź.
- 28.23. Jaki rodzaj mechanizmu QoS będzie odpowiedni do obsługi czatu między dwoma użytkownikami?
- 28.24. Wymień cztery parametry, które charakteryzują mechanizm VBR.
- 28.25. Opisz cztery etapy przetwarzania pakietów w routerze z funkcją QoS.
- 28.26. Jeśli dostawca usług internetowych wykorzystuje algorytm cieknącego wiadra, czy większą przepływność danych zapewni duży, czy mały rozmiar pakietów?
- 28.27. Co to jest DiffServ?
- 28.28. Czy routing MPLS różni się od routingu IP?



# *Zawartość rozdziału*

- 29.1. Wprowadzenie 513
- 29.2. Transmisja w czasie rzeczywistym 513
- 29.3. Opóźnione odtwarzanie i bufory fluktuacji opóźnienia 514
- 29.4. Protokół transportowy czasu rzeczywistego (RTP) 515
- 29.5. Enkapsulacja RTP 516
- 29.6. Telefonia IP 517
- 29.7. Sygnalizacja i standardy sygnalizacji VoIP 518
- 29.8. Elementy składowe systemu telefonii IP 519
- 29.9. Podsumowanie protokołów i podział na warstwy 523
- 29.10. Charakterystyka protokołu H.323 523
- 29.11. Warstwy systemu H.323 524
- 29.12. Charakterystyka protokołu SIP 524
- 29.13. Przebieg sesji SIP 525
- 29.14. Odwzorowanie numerów telefonicznych i routing 525
- 29.15. Podsumowanie 527

# Multimedia i telefonia IP (VoIP)

## 29.1. Wprowadzenie

Rozdziały tej części książki są poświęcone różnorodnym technologiom sieciowym oraz ich zastosowaniom. W poprzednim rozdziale omówiono zagadnienia związane z wydajnością sieci oraz mechanizmami QoS. Przedstawiono w nim dwa sposoby projektowania sieci, które umożliwiają korzystanie z aplikacji czasu rzeczywistego. Wspomniane rozwiązania to budowanie sieci izochronicznych i stosowanie protokołów, które eliminują negatywny wpływ fluktuacji opóźnienia.

Przekazywanie danych multimedialnych w internecie jest również tematem bieżącego rozdziału. Zawarte tutaj informacje pozwalają na zrozumienie zasad przekazywania danych multimedialnych w systemach typu best-effort. Opis obejmuje także ogólne protokoły aplikacji czasu rzeczywistego oraz transmisję danych w ramach połączeń telefonicznych.

## 29.2. Transmisja w czasie rzeczywistym

Termin **multimedia** oznacza dane dźwiękowe lub wideo, które dodatkowo mogą zawierać tekst. Określenie **multimedia czasu rzeczywistego** odnosi się do danych multimedialnych, które muszą być odtwarzane z dokładnie taką samą częstotliwością, z jaką zostały zarejestrowane (przykładem może być przekaz telewizyjny dotyczący bieżącego wydarzenia).

Nasuwa się pytanie, w jaki sposób można wykorzystać internet do transmisji treści multimedialnych w czasie rzeczywistym. Jest w tym pewna trudność, wynikającą z braku gwarancji dostarczania danych. Wiadomo, że pakiety z danymi są narażone na utratę, opóźnienie lub dostarczenie w niewłaściwej kolejności. Zwykłe przekształcenie dźwięku lub obrazu do postaci cyfrowej i odtwarzanie ich w takt nadchodzących pakietów jest niedopuszczalne. W początkowych systemach multimedialnych problem był rozwiązywany przez projektowanie specjalnych sieci przeznaczonych do transmisji dźwięku i wideo.

Do zapewniania wysokiej jakości dźwięku w sieciach telefonicznych wykorzystuje się systemy izochroniczne. Natomiast sieci telewizji kablowej są projektowane w taki sposób, aby strumienie wideo były przekazywane wieloma równoległymi kanałami bez zakłóceń i utraty informacji.

Zamiast nakładać na sieć obowiązek obsługi transmisji w czasie rzeczywistym, internet wykorzystuje specjalne protokoły. Co ciekawe, największą trudnością w ich działaniu jest odpowiednie reagowanie na fluktuację opóźnienia, a nie na utratę pakietów. Aby się o tym przekonać, przeanalizujmy przykład transmisji „na żywo” sygnału wideo. Jeśli protokół transportowy wykorzystuje mechanizm oczekiwania i retransmisji, powtórzone pakiety nadchodzą zbyt późno, by można ich było użyć. Odbiornik zdąży już odtworzyć dźwięk i sekwencję wizyjną z kolejnych pakietów, więc przekazywanie przez sieć utraconego fragmentu strumienia nie ma sensu.

Należy więc zapamiętać, że:

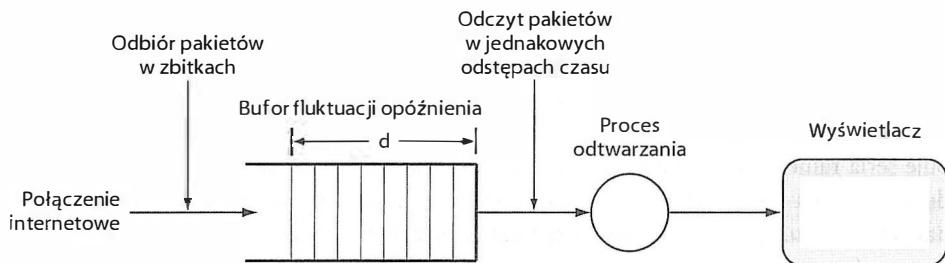
*W przeciwieństwie do konwencjonalnych protokołów transportowych, rozwiązania przeznaczone do transmisji danych czasu rzeczywistego eliminują jedynie skutki fluktuacji opóźnień, a nie retransmitują utraconych pakietów.*

### 29.3. Opóźnione odtwarzanie i bufory fluktuacji opóźnienia

Aby rozwiązać problem fluktuacji opóźnienia i odtwarzać dane w czasie rzeczywistym w sposób niezakłócony, stosuje się dwie metody postępowania:

- **Dołączanie znaczników czasu.** Nadawca dodaje do każdej porcji danych wartość znacznika czasu, która pozwala odbiorcy ustalić odpowiednią kolejność pakietów oraz odtworzyć sekwencję z zachowaniem właściwych zależności czasowych.
- **Wykorzystanie buforów fluktuacji opóźnienia.** Aby wyeliminować skutki fluktuacji opóźnienia, odbiorca buforuje dane i odtwarza je z opóźnieniem.

Implementacja buforów fluktuacji nie jest szczególnie skomplikowana. Odbiorca musi jedynie przechowywać listę odebranych elementów i posługiwać się znacznikami czasu jako wyznacznikami kolejności elementów na liście. Odtwarzanie jest opóźniane o  $d$  jednostek czasu względem nadchodzących danych. Jeśli więc którykolwiek z pakietów zostanie wstrzymany podczas transmisji na czas krótszy niż  $d$ , jego zawartość przed prezentacją zostanie zapisana w buforze. Innymi słowy, odbierane dane są wprowadzane do bufora ze zmienną częstotliwością, ale ich odczyt następuje w równomiernych odstępach czasu. Budowa systemu odtwarzania w czasie rzeczywistym została pokazana na rysunku 29.1.



Rysunek 29.1. Bufory fluktuacji opóźnienia o przesunięciu czasowym d

## 29.4. Protokół transportowy czasu rzeczywistego (RTP)

W stosie protokołów internetowych za przekazywanie danych czasu rzeczywistego odpowiada **protokół transportowy czasu rzeczywistego** (RTP — ang. *Real-time Transport Protocol*). Określenie *transportowy* jest w tym przypadku nieco myjące, ponieważ mechanizm RTP jest implementowany w warstwach powyżej transportowej. Niezależnie od nazwy należy go postrzegać jako protokół transferu danych.

Protokół RTP nie gwarantuje dostarczenia informacji na czas i nie uwzględnia buforów fluktuacji opóźnienia lub mechanizmów odtwarzania informacji. Uzupełnia jednak każdy pakiet o trzy elementy, które umożliwiają odbiorcy użycie wspomnianych buforów:

- **Numer sekwencyjny.** Pozwala on na zapisywanie nadchodzących pakietów w odpowiedniej kolejności i wykrywanie przypadków utraty pakietów.
- **Znacznik czasu.** Umożliwia on odtworzenie treści pakietu w odpowiednim momencie.
- **Identyfikator źródła.** Dzięki niemu odbiorca może rozróżnić źródła danych.

Rozmieszczenie pól numeru sekwencyjnego, znacznika czasu oraz identyfikatora źródła w nagłówku pakietu RTP zostało pokazane na rysunku 29.2.

0	1	3	8	16	31
Wersja	P	X	CC	M	Typ danych
Znacznik czasu					
Identyfikator źródła synchronizacji					
Identyfikator źródła informacji					
...					

Rysunek 29.2. Podstawowy nagłówek rozpoczętyjący każdy pakiet RTP

Pole **WERSJA** przechowuje numer wersji protokołu RTP (obecnie jest to wersja 2). Pole **P** zawiera informację o tym, czy pole danych jest dopełnione zerami (w niektórych technikach kodowania istotne jest przekazywanie bloków danych o jednakowym rozmiarze).

Pole *X* wskazuje, czy transmitowane jest rozszerzenie nagłówka, a pole *CC* zawiera informację o liczbie źródeł, które zostały wykorzystane do wygenerowania strumienia. Wartość *M* jest znacznikiem umożliwiającym wyróżnianie niektórych ramek. Niektóre mechanizmy kodowania sekwencji wizyjnych wysyłają pełne ramki obrazu, po których następuje seria ramek z informacjami różnicowymi. Wartość *M* jest wówczas wykorzystywana do oznaczania ramek pełnych. Pole *TYP DANYCH* służy do określenia zawartości pola danych pakietu. Na jego podstawie program odbiorczy wie, w jaki sposób powinien zinterpretować pozostałą część pakietu.

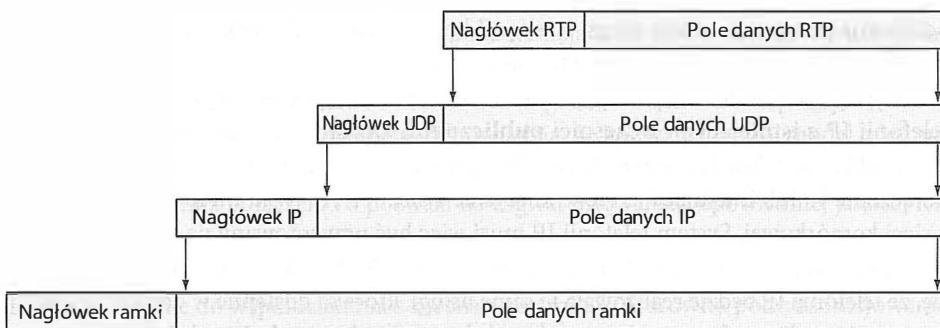
Każdy pakiet zawiera *NUMER SEKWENCYJNY*, zwiększany o jeden w czasie wysyłania informacji. Podobnie jak w przypadku mechanizmu TCP, nadawca losowo wybiera wartość początkową sekwencji, aby uniknąć problemu powtórzeń. Pole *ZNACZNIK CZASU*, niezależnie od numeru sekwencyjnego, dostarcza odbiorcy informację o czasie odtwarzania zawartości pakietu. Uniezależnienie znacznika czasu od numeru sekwencyjnego jest szczególnie istotne w przypadkach, gdy zależność między czasem a liczbą generowanych pakietów nie jest liniowa (kodery generują niekiedy mniej ramek, gdy obraz nie zmienia się gwałtownie).

Wartość *ZNACZNIKA CZASU* w protokole RTP nie odpowiada dacie i czasowi. Mechanizm RTP wybiera początkową wartość w sposób losowy i generuje w odniesieniu do niej pozostałe wartości znacznika. Specyfikacja RTP nie określa również, czy pomiar czasu jest wykonywany w sekundach, milisekundach, czy innych jednostkach. O rozdzielczości zegara decyduje rodzaj danych. Zresztą, niezależnie od rozdzielczości, nadawca musi zwiększać wartość w sposób liniowy, nawet jeśli nie wysyła żadnych pakietów (na przykład w przypadkach, gdy koder dźwięku wykryje ciszę).

Źródła danych opisują dwa pola — *IDENTYFIKATOR ŹRÓDŁA SYNCHRONIZACJI* oraz *IDENTYFIKATOR ŹRÓDŁA INFORMACJI*. Potrzeba identyfikowania źródeł wynika z tego, że dostarczanie pakietów często jest realizowane na zasadzie multiemisji. Oznacza to, że stacja może otrzymywać strumienie z różnych źródeł oraz że pakiety mogą być powielane. Z kolei występowanie wielu identyfikatorów źródeł w jednym strumieniu jest spowodowane tym, że mechanizm RTP umożliwia **miksowanie** (ang. *mixing*) strumieni, czyli łączenie strumieni pochodzących z kilku nadajników w jeden nowy strumień. Mikser może na przykład łączyć oddzielne strumienie dźwięku i wideo, a następnie emitować je w formie strumienia sumarycznego.

## 29.5. Enkapsulacja RTP

Do transportu komunikatów RTP wykorzystywany jest protokół UDP. Przed wysłaniem do internetu każda informacja RTP musi więc zostać zapisana w polu danych segmentu UDP. Na rysunku 29.3 przedstawiono trzy poziomy enkapsulacji, które są konieczne do przesłania komunikatu RTP w ramach pojedynczej sieci.



Rysunek 29.3. Trzy poziomy enkapsulacji komunikatu RTP

Dzięki zastosowaniu protokołu UDP do przenoszenia informacji RTP komunikaty protokołu można rozsyłać w sposób rozgłoszeniowy lub na zasadzie multiemisji. Multiemisja jest niezwykle użyteczna w przypadku transmitowania sygnału wideo do większej grupy odbiorców. Jeśli operator telewizji kablowej udostępnia film lub przekaz z wydarzenia sportowego, musi się liczyć z tym, że więcej osób zechce go obejrzeć. Wówczas zamiast wysyłać oddzielne kopie pakietów do każdego odbiorcy, protokół RTP będzie generował pojedyncze komunikaty, które zostaną przekazane do wszystkich członków logicznej podsieci. W przypadku grupy multiemisji składającej się z  $N$  odbiorców natężenie ruchu zostanie więc zredukowane  $N$ -krotnie.

## 29.6. Telefonia IP

Określenia **telefonia IP** lub transmisja **Voice over IP**<sup>81</sup> (VoIP) są wykorzystywane do opisu jednej z najpowszechniej wykorzystywanych aplikacji multimedialnych. Operatorzy telekomunikacyjni na całym świecie zastępują tradycyjne przełączniki telefoniczne routery IP. Uzasadnieniem jest, oczywiście, ekonomia. Routery są znacznie tańsze niż tradycyjne przełączniki telefoniczne. Przyczyny ekonomiczne sprawiły również, że telefonią IP zainteresowały się przedsiębiorstwa. Przesyłanie danych i głosu w datagramach IP obniża koszty, ponieważ pozwala na wykorzystanie jednej sieci. Jeden zestaw urządzeń, jedna instalacja kablowa i te same połączenia sieciowe zaspakaja wszystkie potrzeby komunikacyjne firmy, również związane z połączeniami telefonicznymi.

Zasada działania telefonii IP jest bardzo prosta. Należy nieustannie próbować sygnał audio, przekształcać każdą próbkę do formatu cyfrowego i wysyłać powstający strumień danych cyfrowych przez sieć IP w formie pakietów. Po drugiej stronie połączenia wystarczy przekształcić odebrane dane do postaci analogowej i odtworzyć. Wykonanie tego zadania komplikuje jednak wiele dodatkowych czynników. Nadajnik nie może czekać na skompletowanie danych, które wypełnią standardowy pakiet, ponieważ wprowadziłoby to wielosekundowe opóźnienia w transmisji. Poza tym system musi dodatkowo obsługiwać proces ustanawiania połączenia. Wybranie numeru oznacza konieczność przekształcenia go na adres IP i odszukania wskazanego rozmówcy. Rozmowa rozpoczyna się w chwili

<sup>81</sup> Czytaj *vojs over aj-pi*.

odebrania połączenia przez drugą stronę. Z kolei zakończenie rozmowy może nastąpić dopiero wtedy, gdy obydwie strony to zaakceptują.

Największe trudności wynikają jednak z konieczności zapewnienia współdziałania sieci telefonii IP z istniejącymi wcześniej **publicznymi sieciami telefonicznymi** (PSTN — ang. *Public Switched Telephone Network*). Zgodnie z tym założeniem osoba odbierająca połączenie IP lub inicjująca je może korzystać z sieci PSTN (w kraju lub za granicą) lub z sieci komórkowej. System telefonii IP musi więc być przygotowany na to, że połączenia będą przebiegały pomiędzy siecią IP i siecią PSTN. Poza tym użytkownicy spodziewają się, że telefonia IP będzie realizowała tejsame usługi, które są dostępne w dotychczasowych systemach, czyli **przekazywanie rozmów, obsługę rozmów oczekujących i skrzynki pocztowej, ustanawianie połączeń konferencyjnych i pokazywanie numeru rozmówcy**. Firmy korzystające z **central abonenckich** (PBX — ang. *Private Branch Exchange*) często wymagają od systemu telefonii IP również realizacji zadań właściwych dla wspomnianych central.

## 29.7. Sygnalizacja i standardy sygnalizacji VoIP

Standardy komunikacji telefonicznej w ramach sieci IP są opracowywane przez dwie organizacje — **Międzynarodową Unię Telekomunikacyjną** (ITU — ang. *International Telecommunications Union*) nadzorującą przygotowywanie standardów telefonicznych oraz organizację IETF, która nadzoruje prace związane ze standardami TCP/IP. Dlatego po zaprezentowaniu ogólnych zasad funkcjonowania telefonii IP przedstawione zostaną rozwiązania zaproponowane przez każdą z wymienionych grup.

Na szczęście, obydwie grupy doszły do porozumienia w kwestii kodowania i transmisji sygnału audio:

- Dźwięk jest kodowany za pomocą **modulacji impulsowo-kodowej** (PCM).
- Do przesyłania dźwięku w formie cyfrowej służy protokół RTP.

Największa trudność w implementacji telefonii IP (i jednocześnie przyczyna powstania wielu standardów) wynika z konieczności ustanawiania połączenia i zarządzania nim w trakcie trwania rozmowy. W telekomunikacji wymiana informacji w czasie ustanawiania i rozłączania połączenia nazywa się **sygnalizacją**. Sygnalizacja obejmuje lokalizację rozmówcy o danym numerze telefonu, utworzenie obwodu z aparatem działającym po drugiej stronie, a także realizację dodatkowych zadań, jak przekazywanie rozmów. W tradycyjnych systemach telefonii stosuje się zazwyczaj mechanizm nazywany **systemem sygnalizacji 7** (SS7).

Jednym z kluczowych problemów telefonii IP jest wybór mechanizmu sygnalizacji. Czy powinien on mieć charakter scentralizowany, tak jak w przypadku klasycznej telefonii, czy rozproszony, jak w przypadku systemu odwzorowania nazw na adresy IP? Zwolennicy systemu rozprozonego argumentują, że telefony IP powinny mieć możliwość odnajdywania urządzeń rozmówcy niezależnie od tego, w którym miejscu w internecie pracują w danej chwili. Komunikacja powinna przypominać tę, która jest obecnie wykorzystywana w internecie (telefon IP miałby pracować jak serwer, który odbiera rozmowy, lub jak klient, który żąda ustanowienia połączenia). W rozwiązaniu rozproszonym nie potrzeba żadnych

dodatkowych elementów infrastruktury poza istniejącymi serwerami DNS i usługami transportu pakietów IP. System tego typu jest szczególnie użyteczny w sieciach lokalnych (umożliwia bowiem wykonywanie rozmów bez angażowania firm zewnętrznych). Zwolennicy rozwiązania scentralizowanego odpowiadają, że konwencjonalny model telefonii najlepiej sprawdza się w praktyce, ponieważ nałada na firmy telekomunikacyjne obowiązek ustanawiania połączeń, co pozwala na zagwarantowanie odpowiedniej jakości ich obsługi.

Aby zachować zgodność z istniejącymi systemami telefonicznymi, nowe protokoły muszą być zdolne do współdziałania z mechanizmami SS7 (zarówno podczas inicjowania połączenia, jak i w czasie jego odbierania). W czasie trwania debat na temat wyboru odpowiedniego rozwiązania dla telefonii IP zaproponowano cztery zestawy protokołów sygnalizacyjnych. Organizacja IETF przedstawiła **protokół inicjowania sesji** (SIP — ang. *Session Initiation Protocol*) oraz **protokół sterowania bramami mediów** (MGCP — ang. *Media Gateway Control Protocol*). Organizacja ITU dostarczyła natomiast obszerny zbiór protokołów pod wspólną nazwą **H.323**. Obydwie grupy opracowały również wspólny standard **Megaco** (**H.248**).

*Proces ustanawiania i kończenia połączenia jest nazywany sygnalizacją. Do obsługi sygnalizacji w systemie telefonii IP zaproponowano wiele specjalistycznych protokołów.*

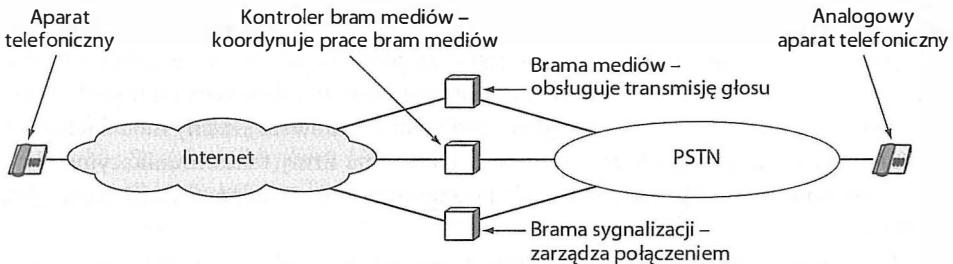
## 29.8. Elementy składowe systemu telefonii IP

Cztery główne komponenty systemu telefonii IP zostały wymienione w tabeli 29.1. Z kolei na rysunku 29.4 przedstawiono sposób wykorzystania tych elementów.

Tabela 29.1. Cztery najważniejsze komponenty systemu telefonii IP

Komponent	Opis
Aparat telefoniczny	Działa jak klasyczny aparat telefoniczny, ale wykorzystuje protokół IP do przesyłania dźwięku w formacie cyfrowym.
Kontroler bramy mediów	Koordynuje komunikację między aparatami telefonicznymi a urządzeniami sieciowymi w czasie ustanawiania, kończenia i przekazywania połączenia.
Brama mediów	Zapewnia połączenie między dwoma sieciami, które wykorzystują różne techniki kodowania, i dokonuje translacji danych wymienianych między tymi sieciami.
Brama sygnalizacji	Łączy dwie sieci o różnych systemach sygnalizacji. Tłumaczy żądania i odpowiedzi związane z zarządzaniem rozmową.

**Aparat telefoniczny.** Jest przyłączony do sieci komputerowej, wykorzystuje w komunikacji protokół IP, ale ma budowę klasycznego aparatu telefonicznego, który umożliwia



Rysunek 29.4. Połączenia między komponentami sieci telefonicznej

użytkownikowi inicjowanie połączeń lub odbieranie rozmów przychodzących. Telefon IP może być niezależnym urządzeniem (tj. aparatem telefonicznym) lub programem, który działa w komputerze wyposażonym w głośniki i mikrofon. Połączenie między telefonem IP a pozostałą częścią systemu może być przewodowe lub bezprzewodowe (na przykład w standardzie Ethernet lub 802.11b).

**Kontroler bram mediów.** Koordynuje wymianę danych związaną z połączeniem między dwoma telefonami IP. Umożliwia użytkownikowi dzwoniącemu zlokalizowanie rozmówcy lub skorzystanie z innych usług, takich jak przekazywanie rozmów.

**Brama mediów.** Odpowiada za translację danych audio podczas ich przekazywania pomiędzy siecią IP i siecią PSTN lub pomiędzy dwoma sieciami IP o różnych technikach kodowania sygnału. Na przykład brama mediów funkcjonująca na styku sieci PSTN i internetu przenosi dane między systemem TDM (wykorzystywanym w klasycznych obwodach rozmownych) a siecią pakietową.

**Brama sygnalizacji.** Działa na połączeniu dwóch niezależnych sieci i odpowiada za tłumaczenie operacji sygnalizacyjnych. Umożliwia inicjowanie połączenia urządzeniom pracującym w jednej z obsługiwanych sieci (na przykład uczestniczy w ustanawianiu połączenia między telefonem IP przyłączonym do internetu a aparatem sieci PSTN). Pracę bram mediów i sygnalizacji koordynuje kontroler bram mediów.

Powysze rozwiązania i terminologia odnoszą się do nieco uproszczonego modelu systemu telefonii IP, który został przedstawiony w opracowaniach IETF i ITU dotyczących protokołów Megaco i MGCP. Praktyczne wdrożenia omawianych systemów są znacznie bardziej skomplikowane, czego potwierdzeniem są następne punkty podrozdziału.

### 29.8.1. Terminologia i zasady działania protokołu SIP

Mechanizm SIP został zaprojektowany w taki sposób, aby ograniczyć potrzebę stosowania dodatkowych protokołów i, jeśli to możliwe, wykorzystywać istniejące wcześniej rozwiązania. Na przykład odwzorowanie numeru telefonu na adres IP jest realizowane za pośrednictwem usług DNS. W specyfikacji SIP zdefiniowano trzy nowe elementy składające się na system sygnalizacji:

- moduł użytkownika,
- serwer lokalizacji,

- serwery wspomagające (pośredniczące, przekierowujące połączenia i rejestrujące urządzenia).

**Moduł użytkownika** (ang. *user agent*). W dokumentacji SIP komponent ten jest określany jako urządzenie, które jest punktem końcowym połączenia telefonicznego. Jego rolę może pełnić telefon IP, komputer lub brama PSTN, która obsługuje połączenia między telefonami IP i siecią PSTN. Moduł użytkownika składa się z dwóch części — **modułu klienckiego** inicjującego połączenia telefoniczne oraz **modułu serwerowego** odpowiedzialnego za odbieranie rozmów przychodzących.

**Serwer lokalizacji** (ang. *location server*). Serwer lokalizacji utrzymuje bazę danych informacji o użytkownikach systemu, odpowiadających im adresach IP, wykupionych usługach oraz preferencjach. Serwer dostarcza informacji na temat lokalizacji użytkownika, z którym ma zostać nawiązane połączenie.

**Serwer pośredniczący** (ang. *proxy server*). W specyfikacji SIP przewidziano serwery pośredniczące, które mają przekazywać żądania od użytkowników do innych lokalizacji. Odpowiadają one za optymalny routing pakietów i przestrzeganie zasad określonej polityki (na przykład sprawdzają, czy użytkownik jest uprawniony do zainicjowania rozmowy).

**Serwery przekierowań** (ang. *redirect server*). Serwery przekierowań służą do przekazywania rozmów oraz do realizowania połączeń z numerami z grupy 800. Po odebraniu żądania od modułu użytkownika serwer odsyła informację o alternatywnej lokalizacji rozmówcy.

**Serwer rejestrujący** (ang. *registrar server*). Serwery rejestrujące odbierają żądania rejestracji i uaktualniają bazę danych lokalizacji użytkowników, z którymi później kontaktują się serwery lokalizacji. Jednostki tego typu są odpowiedzialne za uwierzytelnianie żądań i dbanie o spójność bazy danych.

## 29.8.2. Terminologia i zasady działania protokołu H.323

Opracowany przez organizację ITU standard H.323 odnosi się przede wszystkim do interakcji z systemami PSTN. Mimo że jest niezwykle rozbudowany i szczegółowo definiuje różnorodne funkcje, jego analizę można sprowadzić do przedstawienia kilku wymienionych poniżej elementów.

**Terminal.** Terminal H.323 zapewnia użytkownikowi dostęp do funkcji telefonii IP, a także do mechanizmów transmisji audio i wideo.

**Nadzorca** (ang. *gatekeeper*). Jednostka nadzorcza protokołu H.323 realizuje zadania związane z lokalizacją użytkowników i sygnalizacją oraz koordynuje działania bramy, która zapewnia połączenia z systemem PSTN.

**Brama** (ang. *gateway*). W rozwiązaniach H.323 wykorzystywana jest pojedyncza brama służąca do łączenia systemów telefonii IP i PSTN. Moduł ten odpowiada za translację komunikatów sygnalizacji i treści.

**Moduł sterowania połączaniami wielopunktowymi** (MCU — ang. *Multipoint Control Unit*). Moduł MCU jest wykorzystywany podczas korzystania z takich usług, jak konferencje z wieloma rozmówcami.

### 29.8.3. Terminologia i funkcje systemu ISC

Ponieważ organizacje ITU i IETF przygotowały kilka rozbieżnych specyfikacji, producenci sprzętu sieciowego utworzyli konsorcjum o nazwie ang. *International Softswitch Consortium* (ISC), którego zadaniem jest opracowanie spójnego modelu funkcjonalnego, który połączy wszystkie pozostałe modele w jedną platformę telefonii IP. W tym celu zdefiniowano niezbędne funkcje systemu, w tym sygnalizację między rozwiązaniami różnego typu, translację kodowania, realizację usług dodatkowych (takich jak przekazywanie połączeń) oraz zarządzanie (operowanie kontami użytkowników i obsługę płatności). Ostateczna lista funkcji opracowanych przez konsorcjum ISC zawiera następujące pozycje:

**Funkcja kontrolera bram mediów** (MGC-F — ang. *Media Gateway Controller Function*). Jej zadanie polega na przechowywaniu informacji na temat punktów końcowych oraz sterowaniu przebiegiem połączenia.

**Funkcja modułu połączenia** (CA-F — ang. *Call Agent Function*). Zadania CA-F są podzbiorem zadań MGC-F. Dotyczą zarządzania informacjami o stanie połączenia. W skład grupy CA-F wchodzą protokoły SIP, H.323 i Q.931.

**Funkcja współpracy z innymi sieciami** (IW-F — ang. *InterWorking Function*). Zadania IW-F są podzbiorem zadań MGC-F i dotyczą obsługi sygnalizacji w sieciach heterogenicznych, takich jak SS7 i SIP.

**Funkcja routingu i funkcja obsługi kont** (R-F — ang. *Routing Function*; A-F — ang. *Accounting Function*). Mechanizmy R-F odpowiadają za wybór tras dla połączeń MGC-F. Natomiast moduły A-F zbierają informacje związane z kontem i rozliczeniami danego użytkownika.

**Funkcja bram sygnalizacji** (SG-F — ang. *Signaling Gateway Function*). Komponenty SG-F odpowiadają za przekazywanie sygnalizacji między sieciami IP i PSTN.

**Funkcja sygnalizacji w bramie dostępowej** (AGS-F — ang. *Access Gateway Signaling Function*). Moduł ten przekazuje sygnalizację między sieciami IP i siecią dostępową z przełączaniem obwodów (taką jak PSTN).

**Funkcja serwera aplikacji** (AS-F — ang. *Application Server Function*). Komponent AS-F obsługuje usługi takie jak poczta głosowa.

**Funkcja sterowania usługami** (SC-F — ang. *Service Control Function*). Funkcje SC-F są wywoływanie w przypadkach, w których moduł AS-F musi zmienić logikę działania usługi (na przykład zdefiniować nowe odwzorowanie numeru).

**Funkcja bramy mediów** (MG-F — ang. *Media Gateway Function*). Komponent MG-F odpowiada za zamianę jednego formatu cyfrowego sygnału dźwięku na inny format. Do jego zadań może również należeć reagowanie na takie zdarzenia, jak odkładanie słuchawki lub generowanie sygnałów DTMF.

**Funkcja serwera mediów** (MS-F — ang. *Media Server Function*). Moduł MS-F przetwarza strumień pakietów na potrzeby aplikacji AS-F.

## 29.9. Podsumowanie protokołów i podział na warstwy

Ponieważ wiele grup projektowych promuje własne rozwiązania z dziedziny telefonii IP, w każdej warstwie stosu protokołów występuje wiele konkurujących ze sobą rozwiązań. Część z proponowanych mechanizmów została przedstawiona w tabeli 29.2, w której uwzględniono również informacje na temat powiązań protokołów z odpowiednimi warstwami internetowego modelu odniesienia.

Tabela 29.2. Zestawienie protokołów telefonii IP

Warstwa	Obsługa rozmowy	Multimedia użytkownika	Dane użytkownika	Wsparcie	Routing	Transport sygnału
5	H.323, Magaco, MGCP, SIP	RTP	T.120	RTCP, RTSP, NTP, SDP	ENUM, TRIP	SIGTRAN <sup>82</sup>
4	TCP, UDP	UDP	TCP	TCP, UDP		SCTP
3	IP, RSVP i IGMP					

## 29.10. Charakterystyka protokołu H.323

Standard H.323 nie opisuje pojedynczego protokołu, ale zbiór rozwiązań, które współpracując ze sobą, odpowiadają za każdy aspekt komunikacji telefonicznej. Najważniejsze cechy rozwiązania H.323 to:

- Obsługa wszystkich elementów cyfrowej rozmowy telefonicznej.
- Uwzględnienie sygnalizacji, która umożliwia ustanawianie i rozłączanie połączeń.
- Możliwość przekazywania strumieni wideo i danych w czasie prowadzenia rozmowy.
- Generowanie komunikatów binarnych zgodnie z notacją ASN.1 i kodowaniem BER.
- Uwzględnienie protokołów zwiększających bezpieczeństwo rozwiązania.
- Wykorzystanie specjalnych modułów sprzętowych (MCU) do obsługi połączeń konferencyjnych.
- Wyznaczenie serwerów do realizacji takich zadań jak **odwzorowanie adresów** (tj. tłumaczenie numeru telefonu rozmówcy na adres IP jego urządzenia), **uwierzytelnianie, autoryzacja** (tj. ustalanie praw dostępu użytkownika do określonej usługi), **rejestrowanie zdarzeń** oraz **usługi dodatkowe** (na przykład przekazywanie rozmów).

<sup>82</sup> Protokół SIGTRAN umożliwia przekazywanie sygnałów PSTN (np. SS7 i DTMF) w sieciach IP. Z kolei protokół SCTP odpowiada za multipleksację wielu strumieni wejściowych w ramach jednego strumienia warstwy transportowej.

## 29.11. Warstwy systemu H.323

Protokoły wchodzące w skład systemu H.323 korzystają zarówno z protokołu TCP, jak i UDP. Przekaz głosu jest realizowany na bazie protokołu UDP. Natomiast protokół TCP służy do transmisji danych. Zdefiniowany w standardzie H.323 podział na warstwy został pokazany w tabeli 29.3.

Tabela 29.3. Rozmieszczenie najważniejszych protokołów H.323 w warstwach modelu odniesienia

Warstwa	Sygnalizacja	Rejestracja	Audio	Wide o	Dane	Bezpieczeństwo
5	H.225.0-Q.931, H.250-Dodatek G, H.245, H.250	H.225.9-RAS	G.711, H.263, G.722, G.723, G.728	H.261, H.323	T.120	H.235
4	TCP, UDP	UDP			TCP	TCP, UDP
3	IP, RSVP i IGMP					

## 29.12. Charakterystyka protokołu SIP

Najważniejsze cechy **protokołu inicjowania sesji** (SIP) to:

- Praca w warstwie aplikacji.
- Uwzględnienie wszystkich aspektów sygnalizacji, w tym lokalizacji rozmówcy, powiadamiania o rozmowie i konfiguracji połączenia (uruchamianie dzwonka telefonu), ustalania dostępności rozmówcy (informowania o tym, czy może odbierać rozmowy) i rozłączania połączenia.
- Realizacja usług dodatkowych, takich jak przekazywanie rozmów.
- Wykorzystanie multiemisji w połączeniach konferencyjnych.
- Umożliwienie urządzeniom negocjowania parametrów transmisji i rodzaju medium<sup>83</sup>.

Do wyszukania rozmówcy w systemie SIP służy adres URI, który składa się z nazwy konta użytkownika systemu oraz nazwy domenowej. Na przykład osoba o nazwisku *Kowalski* pracująca w zakładzie *SuperFirma sp. z o.o.* mogłaby posługiwać się adresem:

sip:kowalski@superfirma.pl

W standardzie SIP zdefiniowano sześć podstawowych rodzajów komunikatów i siedem rozszerzeń. Komunikaty podstawowe są nazywane **metodami** (ang. *methods*). Najważniejsze metody SIP zostały wymienione w tabeli 29.4.

<sup>83</sup> Do opisu parametrów oraz dostępnych funkcji urządzenia wykorzystywany jest **protokół opisu sesji** (SDP — ang. *Session Description Protocol*).

Tabela 29.4. Sześć podstawowych metod protokołu SIP

Metoda	Przeznaczenie
INVITE	Utworzenie sesji. Rozmówca jest zapraszany do wzięcia udziału w danej sesji.
ACK	Zgoda na zaproszenie (odpowiedź na wywołanie INVITE).
BYE	Zakończenie sesji (zakończenie rozmowy).
CANCEL	Przerwanie wykonywanego żądania (bez efektu, jeśli żądanie zostało zrealizowane).
REGISTER	Zarejestrowanie lokalizacji użytkownika (adresu URL, pod którym użytkownik jest dostępny).
OPTIONS	Zapytanie o funkcje obsługiwane przez zdalne urządzenie.

## 29.13. Przebieg sesji SIP

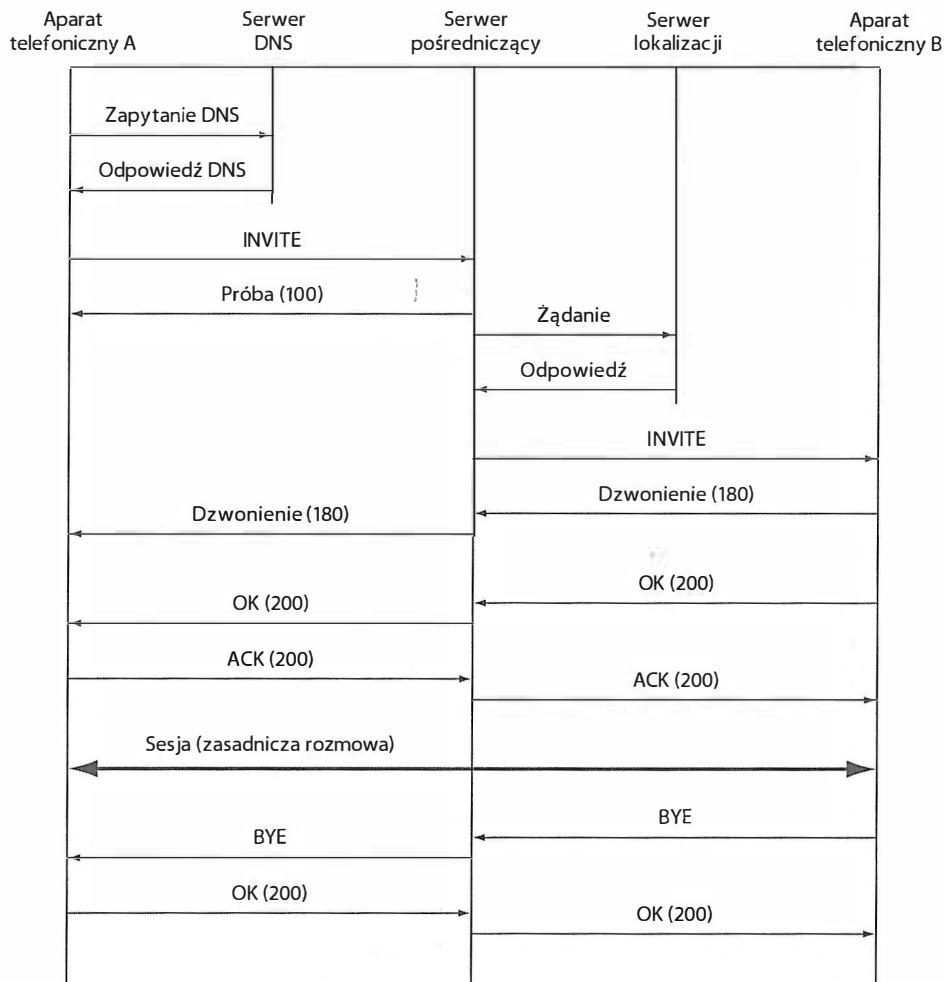
Analiza przykładowych komunikatów wymienianych w ramach sesji SIP powinna pomóc w zrozumieniu ogólnej zasady działania telefonii IP. Na rysunku 29.5 przedstawiono sekwencję pakietów generowanych przez urządzenie A w czasie odwołań do serwera DNS, a później do serwera pośredniczącego, który z kolei komunikuje się z serwerem lokalizacji<sup>84</sup>. Po ustanowieniu połączenia telefony IP wymieniają dane bezpośrednio między sobą. Protokół SIP jest ponownie wykorzystywany do zakończenia rozmowy.

Telefon IP ma zazwyczaj zapisany adres IP co najmniej jednego serwera DNS (który odpowiada za przekształcenie nazwy domenowej z identyfikatora URI na adres IP) oraz adresy serwerów pośredniczących. Każdy serwer pośredniczący dysponuje natomiast adresem lub większą liczbą adresów serwerów lokalizacji. Dzięki temu może szybko zrealizować żądanie, nawet jeśli jeden z serwerów jest niedostępny.

## 29.14. Odwzorowanie numerów telefonicznych i routing

W jaki sposób określa się nazwy i lokalizację użytkowników telefonii IP? W sieciach PSTN obowiązuje standard E.164, opisujący numerację telefoniczną. W rozwiązaniach IP wykorzystywane są natomiast adresy IP. Problem lokalizacji użytkowników jest dość skomplikowany, ponieważ wymaga uwzględnienia sieci działających na bazie różnych technologii. Przykładem takiej konfiguracji jest system złożony z dwóch sieci PSTN rozdzielonych siecią IP. Z perspektywy projektantów trudności są dwie — lokalizacja użytkowników w zintegrowanej sieci oraz ustalenie wydanej trasy do użytkownika. Organizacja IETF opracowała dwa protokoły, które służą do odwzorowywania adresów i jednocześnie rozwiążają obydwa problemy:

<sup>84</sup> Standard SIP dopuszcza **rozgałęzianie rozmów** (ang. *call forking*), co oznacza, że serwer lokalizacji może zwrócić kilka informacji o lokalizacji użytkownika (na przykład w domu i w biurze). To z kolei prowadzi do ustanowienia kilku jednocześnie połączzeń.



Rysunek 29.5. Przykład wymiany komunikatów protokołu SIP w czasie rozmowy telefonicznej

- ENUM — mechanizm odwzorowania numeru telefonu na adres URI;
- TRIP — mechanizm wyszukiwania użytkowników w zintegrowanej sieci.

**ENUM.** Protokół ENUM rozwiązuje problem przekształcania numerów telefonicznych zgodnych ze specyfikacją E.164 na **ujednolicone identyfikatory zasobów** (URI — ang. *Uniform Resource Identifier*). Do przechowywania informacji o powiązaniach między wartościami mechanizm ENUM wykorzystuje system nazw domenowych (DNS). Numer telefonu jest w nim przekształcany na specjalną nazwę w domenie:

e164.arpa

Konwersja jest wykonywana w następujący sposób. Numer jest traktowany jak ciąg tekstowy, w którym poszczególne cyfry są zapisywane w odwrotnej kolejności w sposób właściwy dla segmentów nazwy domenowej. Na przykład numer 32 231 22 19 zostałby zapisany w serwerze DNS jako:

### 9.1.2.2.1.3.2.2.3.e164.arpa

Powiązania w systemie ENUM mogą mieć charakter jeden-do-jednego (podobnie jak w klasycznej numeracji telefonicznej) lub jeden-do-wielu. Oznacza to, że telefon stacjonarny i komórkowy mogą mieć przypisany ten sam numer. Jeśli dany numer odpowiada wielu stacjom, serwer DNS zwraca listę wartości wraz z informacją o tym, jakiego protokołu należy użyć, aby połączyć się z określona jednostką. Moduł użytkownika próbuje tak długo nawiązać połączenie ze stacjami wymienionymi na liście, aż jedno z urządzeń odpowie.

**TRIP.** Protokół **routingu telefonicznego w sieciach IP** (TRIP — ang. *Telephone Routing over IP*) rozwiązuje problem wyszukania użytkownika w zintegrowanej sieci. Z mechanizmu TRIP korzystają serwery lokalizacji oraz inne urządzenia sieciowe, które dostarczają informacji o trasach. Jest on również wykorzystywany w komunikacji między dwoma serwerami lokalizacji, które chcą się poinformować o znanych sobie trasach. Ponieważ protokół TRIP jest niezależny od protokołów sygnalizacji, współdziała zarówno z systemem SIP, jak i z innymi mechanizmami sygnalizacji.

W rozwiązaniu tym zastosowano podział ogólnoświatowej sieci na **domeny administracyjne telefonii IP** (ITAD — ang. *IP Telephone Administrative Domains*). Zgodnie z założeniami standardu generowane przez serwery lokalizacji ogłoszenia TRIP identyfikują punkty wejścia do domeny (serwery lokalizacji informują siebie nawzajem o trasie do bramy sygnalizacji, która łączy różne domeny ITAD). Sam protokół został zaprojektowany jako rozwojowy, gdyż idea routingu w telefonii IP jest pewną nowością i może w przyszłości wymagać wprowadzenia zmian.

## 29.15. Podsumowanie

Protokół transportowy czasu rzeczywistego (RTP) odpowiada za przesyłanie danych multimedialnych w internecie. Komunikat RTP zawiera numer sekwencyjny oraz znacznik czasu, a także wartości identyfikujące źródło lub źródła danych. Znacznik czasu umożliwia właściwe ułożenie danych w buforze przed odtworzeniem ich po stronie odbiorczej. Transmisja informacji RTP wymaga uprzedniego zapisania ich w polu danych segmentu UDP, co pozwala na stosowanie multiemisji i rozgłaszenia. Nie stosuje się retransmisji, ponieważ pakiety dostarczone po odtworzeniu dalszych fragmentów przekazu nie są do niczego potrzebne.

Terminy „telefonia IP” oraz „VoIP” odnoszą się do przesyłania przez internet cyfrowej postaci dźwięku, rejestrowanego podczas rozmowy telefonicznej. Jednym z największych wyzwań związanych z budowaniem systemu telefonii IP jest zapewnienie zgodności z wcześniejszymi rozwiązaniami. Konieczne jest więc stosowanie bram, które łączą sieci telefonii IP z tradycyjnymi systemami PSTN. Wspomniane bramy odpowiadają za translację formatu danych (zamianę kodowania cyfrowego sygnału głosu) oraz sygnalizacji (dopasowanie mechanizmów zarządzania połączeniem).

Standardy telefonii IP zostały opracowane zarówno przez organizacje ITU, jak i IETF. Zaproponowany przez Międzynarodową Unię Telekomunikacyjną standard H.323 obejmuje wiele protokołów przeznaczonych do ustanawiania połączeń oraz zarządzania połączeniami, uwierzytelniania użytkowników i monitorowania ich poczynań, a także za realizację

dodatkowych usług, takich jak przekazywanie rozmów czy transmisja sekwencji wizyjnych i danych w czasie prowadzenia rozmowy. Standard SIP (przygotowany przez organizację IETF) obsługuje sygnalizację, która odpowiada za lokalizację użytkowników, ustalanie połączeń oraz wymianę informacji o funkcjach obsługiwanych przez urządzenia końcowe. W systemie SIP pracuje wiele serwerów wykonujących różnorodne zadania. Są to serwery nazw domenowych, serwery pośredniczące w przenoszeniu rozmów oraz serwery lokalizacji użytkowników. Standaryzacją telefonii IP zajęło się również konsorcjum ISC, które opracowało projekt platformy obejmującej wszystkie wcześniejsze modele technologii VoIP.

Organizacja IETF przygotowała dodatkowo dwa protokoły, które realizują zadania pomocnicze. Mechanizm ENUM wykorzystuje system nazw domenowych do odwzorowywania numerów telefonicznych zgodnych ze specyfikacją E.164 na identyfikatory URI (stosowane przede wszystkim w rozwiązańach SIP). Protokół TRIP odpowiada za routing między domenami administracyjnymi telefonii IP. Jest on wykorzystywany przez serwery lokalizacji SIP do wymiany informacji na temat bram wejściowych w danych sieciach.

## Do samodzielnego studiowania

Protokół SIP został zdefiniowany w dokumencie RFC 3216. Dokument RFC 2916 opisuje numerację E.164 oraz system DNS. Specyfikacja protokołu TRIP znajduje się w dokumencie RFC 3219. Protokół RTP i towarzyszący mu protokół RTCP są przedstawione w dokumencie RFC 1889. Inne rozwiązania z tego zakresu są opisane w dokumentach RFC o numerach 2915, 2871, 3015, 3435 oraz 3475.

## ZADANIA

- 29.1. Scharakteryzuj dane multimedialne. Jakie dwie techniki eliminują negatywne skutki fluktuacji opóźnienia?
- 29.2. Wyjaśnij, dlaczego buforowanie danych gwarantuje poprawne odtworzenie sygnału dźwiękowego, mimo występowania fluktuacji opóźnienia w transmisji internetowej.
- 29.3. Czy po przechwycieniu komunikatu RTP przesyłanego przez internet można określić czas nadania pakietu na podstawie znacznika czasu? Jeśli tak, to w jaki sposób? Jeśli nie, to dlaczego?
- 29.4. Komunikat RTP jest przesyłany w segmencie UDP. Może więc zostać powielony. Czy odbiorca musi przechowywać kopie wszystkich wcześniej odebranych komunikatów, aby ustalić, czy nadchodząca wiadomość została zduplikowana? Uzasadnij odpowiedź.
- 29.5. Protokołowi RTP towarzyszy protokół RTCP, dzięki któremu odbiorca informuje nadawcę o jakości otrzymywanych komunikatów. W jaki sposób można wykorzystać te informacje w adaptacyjnym mechanizmie kodowania wideo?
- 29.6. Ile bitów zostanie wygenerowanych przez koder w ciągu pół sekundy, jeśli do konwersji sygnału dźwiękowego na format cyfrowy zostanie wykorzystana modulacja PCM?
- 29.7. Na podstawie informacji z poprzedniego zadania oszacuj rozmiar (w oktetach) datagramu IP, który przenosi dane z 250 ms transmisji dźwięku. Przyjmij, że do kodowania sygnału zastosowano modulację PCM, a komunikaty RTP są enkapsulowane w segmentach UDP. Podpowiedź: rozmiar nagłówka RTP opisano w dokumencie RFC 1889.

- 29.8. Które elementy telefonii IP są obsługiwane w systemach H.323?
- 29.9. Jaki protokół transportowy jest wykorzystywany do przesyłania w standardzie H.323 danych, które towarzyszą przekazowi audio lub wideo?
- 29.10. Wymień sześć metod SIP.
- 29.11. Zapoznaj się z dokumentem RFC opisującym protokół SIP i popraw rysunek 29.5 tak, aby przedstawał przypadek przekazania rozmowy. Podpowiedź: szukaj komunikatów REDIRECTION.
- 29.12. Jakie jest przeznaczenie protokołów ENUM i TRIP?
- 29.13. Zastanów się na sposobem działania telefonii IP i telefonii analogowej. Które rozwiązanie lepiej sprawdziłoby się podczas wojny? Uzasadnij odpowiedź.
- 29.14. Sprawdź domenę *e164arpa*. Która organizacja jest za nią odpowiedzialna?

# Zawartość rozdziału

- 30.1. Wprowadzenie 531
- 30.2. Działalność przestępco i ataki sieciowe 531
- 30.3. Polityka bezpieczeństwa 534
- 30.4. Odpowiedzialność za dane i nadzór nad nimi 536
- 30.5. Technologie związane z bezpieczeństwem 536
- 30.6. Generowanie skrótów — weryfikacja spójności danych i uwierzytelnianie 537
- 30.7. Kontrola dostępu i hasła 538
- 30.8. Szyfrowanie — podstawowa technika zabezpieczeń 538
- 30.9. Szyfrowanie z użyciem klucza prywatnego 539
- 30.10. Szyfrowanie z użyciem klucza publicznego 539
- 30.11. Uwierzytelnianie z wykorzystaniem podpisów cyfrowych 540
- 30.12. Organa zarządzające kluczami i certyfikaty cyfrowe 541
- 30.13. Zapory sieciowe 543
- 30.14. Zapory sieciowe z filtrowaniem pakietów 544
- 30.15. Systemy wykrywania włamań 545
- 30.16. Skanowanie treści i szczegółowa inspekcja pakietów 546
- 30.17. Wirtualne sieci prywatne (VPN) 547
- 30.18. Wykorzystanie technologii VPN w pracy zdalnej 549
- 30.19. Szyfrowanie pakietów a tunelowanie 550
- 30.20. Rozwiązania z zakresu bezpieczeństwa sieci 552
- 30.21. Podsumowanie 553

# 30

## Bezpieczeństwo sieci

### 30.1. Wprowadzenie

W poprzednich rozdziałach opisano urządzenia i oprogramowanie składające się na internet oraz zasady wykorzystywania istniejącej infrastruktury sieciowej do komunikowania się. Celem tego rozdziału jest przedstawienie najważniejszych zagadnień związanych z bezpieczeństwem sieci. Omówiono tutaj typowe przestępstwa komputerowe, rodzaje zagrożeń oraz techniki podnoszenia poziomu zabezpieczeń systemu.

### 30.2. Działalność przestępca i ataki sieciowe

Każda nowa technologia jest obiektem zainteresowania przestępco. Internet nie jest w tym względzie wyjątkiem. Większość jego użytkowników zdaje sobie z tego doskonale sprawę, gdyż prasa na bieżąco informuje o wykorzystywaniu sieci do działalności kryminalnej. Choć przestępstwa takie jak kradzież tożsamości dotyczą pojedynczych osób, większość operacji prowadzonych na poważną skalę zagraża firmom. Korporacje nie obawiają się jedynie strat wynikających z samej kradzieży dóbr lub usług, ale także trwałe utraty wiarygodności, zniszczenia reputacji, utraty zaufania klientów oraz kradzieży własności intelektualnych.

Podczas rozważania zagadnień związanych z bezpieczeństwem nasuwa się kilka pytań:

- Jakie są główne problemy i zagrożenia internetowe?
- W jaki sposób przestępcy wykorzystują luki w protokołach?
- Jakie są najważniejsze funkcje systemu zabezpieczeń?
- Jakie technologie umożliwiają podniesienie poziomu zabezpieczeń?

Najważniejsze zagrożenia internetowe zostały wymienione w tabeli 30.1.

Tabela 30.1. Najważniejsze zagrożenia internetowe

Problem	Opis
Phishing	Podszywanie się pod znaną organizację (na przykład bank) w celu wyłudzenia poufnych informacji, zazwyczaj numeru rachunku bankowego i hasła dostępowego.
Naciąganie	Przedstawianie nieprawdziwych lub wyobrzymionych informacji na temat produktów lub usług oraz dostarczanie uszkodzonych lub niepełnowartościowych produktów.
Wyłudzenia	Różne formy oszustw, które mają na celu nakłonienie naiwnych użytkowników sieci do zainwestowania pieniędzy.
Odmowa obsługi	Celowe zablokowanie określonej witryny internetowej, które skutkuje przerwaniem lub spowolnieniem działania serwisu.
Utrata kontroli	Przejście kontroli nad systemem użytkownika i wykorzystanie tego systemu w działalności przestępcej.
Utrata danych	Utrata dóbr intelektualnych lub ważnych informacji biznesowych.

Analizując zabezpieczenia systemu, trzeba umieć odróżniać klasyczne przestępstwa dokonane przy użyciu internetu oraz działania wymierzone przeciw systemom internetowym. Wyobraźmy sobie na przykład wykorzystanie przez przestępcoów telefonii VoIP do komunikacji czy zakup w internecie narzędzi użytych później w przestępstwie albo wyłudzenie polegające na celowym oszukaniu niczego nieodejrzewającej ofiary. Choć każdy z wymienionych przypadków jest przedmiotem zainteresowania policji, nie ma nic wspólnego z technologiami sieciowymi. Równie skutecznie można by wykonać te same zadania z użyciem innych form komunikacji. Dwa rodzaje konwencjonalnych przestępstw, które są najczęściej popełniane w internecie, to naciąganie klientów i dostarczanie wadliwych produktów. Pierwsze działanie polega na prezentowaniu nieprawdziwych informacji na temat oferowanych towarów, co jest formą fałszywej reklamy. Drugi rodzaj przestępstwa nie różni się niczym od oszustw popełnianych w innych systemach sprzedaży wysyłkowej.

W dalszej części rozdziału opisane zostaną działania polegające na wykorzystaniu luk w technologiach sieciowych oraz mechanizmy wdrażane w celu utrudnienia lub podniesienia kosztów popełnienia przestępstwa. W tabeli 30.2 została przedstawiona lista najczęściej stosowanych technik ataków.

Techniki **podслушаń i powtarzania pakietów** nie wymagają szczególnych wyjaśnień. Ciekawe, że jednym z najpopularniejszych sposobów wykorzystania słabości systemu jest zastosowanie techniki **przepelnienia bufora**. Skuteczność tego rodzaju ataku jest wynikiem kiepskiej jakości oprogramowania. Programiści nie sprawdzają, czy podczas wprowadzania danych nie dochodzi do przekroczenia rozmiaru bufora. Typowy atak polega na przesłaniu zbyt dużego pakietu (większego, niż wynika ze standardu) lub wygenerowaniu sekwencji pakietów, które przepełnią zarezerwowaną dla nich pamięć.

**Fałszowanie adresu** ma na celu podszywanie się pod zaufaną stację. Najprostszy sposób podmiany adresu polega na wysłaniu rozgłoszenia ARP, które powiąże wskazany adres IP (A) z adresem MAC jednostki atakującej. Jeśli którykolwiek z komputerów wyśle pakiet

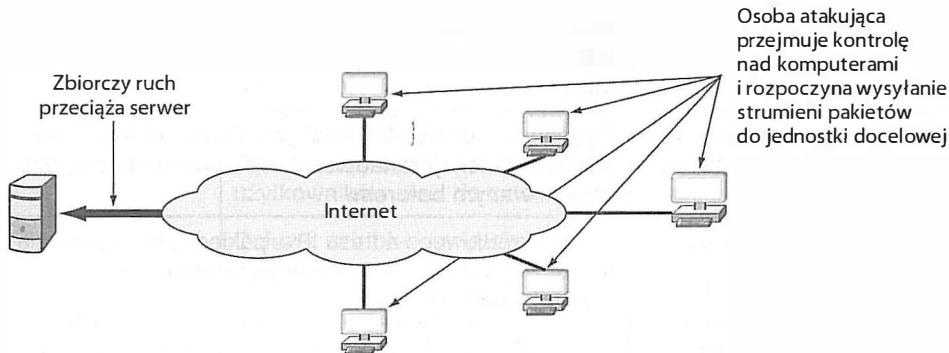
Tabela 30.2. Techniki ataków na systemy informatyczne

Technika	Opis
Podsłuchiwanie	Sporządzanie kopii pakietów przesyłanych przez sieć w celu uzyskania zawartych w nich informacji.
Powtarzanie pakietów	Powtórne przesyłanie pakietów z wcześniejszych sesji komunikacyjnych (na przykład pakietu logowania do systemu).
Przepelnienie bufora	Przestanie większej ilości danych, niż użytkownik może odebrać. Celem jest nadpisanie obszarów pamięci poza zaalokowanym buforem.
Fałszowanie adresu	Zmiana źródłowego adresu IP w pakiecie, która pozwala na oszukanie odbiorcy i wymuszenie na nim przetworzenia pakietu.
Fałszowanie nazwy	Użycie błędnie zapisanej nazwy popularnego serwisu lub zatrucie serwerów DNS niepoprawnymi odwzorowaniami adresów.
DoS i DDoS	Zalewanie serwerów pakietami prowadzące do wstrzymania ich normalnej pracy.
Zalewanie pakietami SYN	Wysyłanie generowanych losowo segmentów TCP SYN w celu wyczerpania zasobu dostępnych połączeń TCP odbiorcy.
Łamanie kluczy	Proces automatycznego odgadywania kluczy szyfrowania lub haseł w celu uzyskania dostępu do poufnych danych.
Skanowanie portów	Próba nawiązania połączenia z każdym dostępnym portem komputera zdalnego i wykrycia ewentualnych luk w zabezpieczeniach.
Przejmowanie pakietów	Przechwytywanie pakietów w internecie, które umożliwia podmianę treści (atak typu man-in-the-middle).

na adres A, pakiet ten zostanie dostarczony do komputera atakującego. Inne techniki uwzględniają wykorzystanie protokołów routingu do rozsyłania nieprawdziwych informacji o trasach, dostarczanie do serwerów DNS błędnych definicji odwzorowania adresów oraz używanie nieznacznie zmodyfikowanych adresów domenowych znanych serwisów, aby sprawić wrażenie osoby mającej dostęp do zaufanej sieci.

Atak **DoS** (odmowy obsługi; ang. *Denial of Service*) polega na zalaniu komputera (zazwyczaj serwera WWW) pakietami żądań. Choć serwer działa nieprzerwanie, atak prowadzi do wyczerpania jego zasobów. To z kolei oznacza, że realizacja żądań innych użytkowników jest znacznie opóźniona, a niektóre połączenia są zrywane. Ustalenie źródła strumienia żądań oraz zablokowanie takiego strumienia nie stanowi dla administratora większego problemu. Dlatego często stosowany jest **rozproszony atak DoS** (DDoS — ang. *Distributed Denial of Service*), w którym uczestniczy duża liczba komputerów generujących

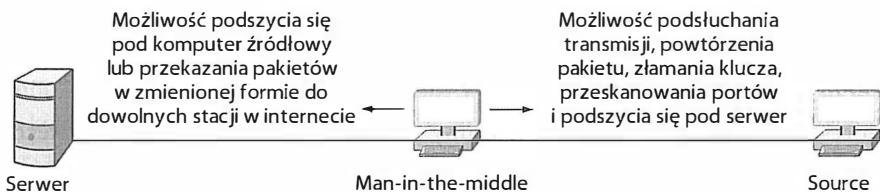
strumienie pakietów. Zasadę działania ataku DDoS przedstawiono na rysunku 30.1. Osoba atakująca najpierw przejmuje kontrolę nad komputerami przyłączonymi do internetu, następnie instaluje w nich specjalne oprogramowanie i wykorzystuje w ataku na wybrany serwer. Żaden z pakietów DDoS nie pochodzi wówczas z komputera osoby atakującej.



Rysunek 30.1. Atak DDoS

**Zalewanie pacjentami SYN** jest pewną wersją ataku DoS wykorzystującą protokół TCP. Każdy dostarczany do odbiorcy pakiet zawiera bowiem komunikat TCP SYN i jest żądaniem ustanowienia nowego połączenia TCP. Odbiorca alokuje pewne zasoby potrzebne do nawiązania takiego połączenia, wysyła segment SYN + ACK i oczekuje na odpowiedź jednostki zdalnej. Po pewnym czasie wszystkie zasoby połączeniowe są wyczerpane i stacja nie może nawiązywać nowych połączeń.

**Przejmowanie pacjentów** pozwala na rozpoczęcie ataku **man-in-the-middle** (człowiek pośrodku), w którym stacja pośrednia może modyfikować treść pakietów podczas przesyłania ich do jednostki docelowej. Choć ten rodzaj ataku jest najtrudniejszy do wykonania, niesie największe zagrożenie. Zasada działania mechanizmu została przedstawiona na rysunku 30.2.



Rysunek 30.2. Włączenie jednostki pośredniczącej w przesyłaniu danych i możliwe formy ataków

### 30.3. Polityka bezpieczeństwa

Jaka sieć jest bezpieczna? Choć pojęcie bezpiecznej sieci jest zrozumiałe dla większości użytkowników, sieci nie można klasyfikować po prostu jako bezpiecznych i niegwarantujących bezpieczeństwa, ponieważ określenie to nie jest precyzyjne. Każda organizacja definiuje własny poziom zabezpieczeń, który wyznacza działania dozwolone i zabronione. Na przykład firma przechowująca wartościowe informacje handlowe może chronić dostęp

do komputerów lokalnych z systemów zewnętrznych. W przedsiębiorstwie publikującym pewne informacje na stronach serwisu internetowego za sieć bezpieczną może zostać uznana ta część systemu, w której dostęp do danych nie jest w żaden sposób ograniczony, ale ich modyfikowanie jest zarezerwowane jedynie dla komputerów wewnętrznych. Organizacje koncentrujące się na zapewnieniu poufności wymiany informacji jako sieci bezpiecznej definiują te, w których nikt oprócz nadawcy i odbiorcy nie może przechwycić lub przeczytać wiadomości. W dużych korporacjach często definicja bezpiecznego systemu jest znacznie bardziej złożona i precyzyjnie określa, które obszary sieci są ogólnie dostępne, a w których dostęp do danych, ich modyfikacja oraz korzystanie z usług są ograniczone.

Ponieważ nie ma jednoznacznej definicji **bezpiecznej sieci**, pierwszym etapem zabezpieczania systemów informatycznych przedsiębiorstwa jest określenie **polityki bezpieczeństwa**. Nie opisuje ona wdrażanych mechanizmów bezpieczeństwa, ale jasno preczyzuje elementy, które podlegają ochronie.

Polityka bezpieczeństwa jest dość skomplikowanym dokumentem, ponieważ musi uwzględniać wszystkie działania ludzi związane z funkcjonowaniem sieci oraz działanie samych urządzeń sieciowych (na przykład postępowanie z pamięciami Flash przynoszonymi przez osoby spoza firmy, dostępność sieci bezprzewodowej poza budynkami firmowymi czy praca zdalna). Szacowanie kosztów i zysków związanych z zastosowaniem określonych rozwiązań dodatkowo zwiększa złożoność dokumentu. Nie można bowiem wdrożyć odpowiedniej polityki bezpieczeństwa, jeśli przedsiębiorstwo nie zna wartości własnych informacji. W wielu przypadkach jest ona jednak trudna do oszacowania. Wyobraźmy sobie kadrową bazę danych, w której każdy rekord odpowiada jednemu pracownikowi. Są w nim zapisane informacje o przepracowanej liczbie godzin oraz stawce za godzinę pracy. Jeśli dostęp do zasobów bazy danych będą mieli wszyscy pracownicy, część z nich może zażądać podniesienia wynagrodzenia. Z kolei udostępnienie tego typu informacji przedstawicielom konkurencyjnej firmy może spowodować, że zaoferują oni lepsze warunki pracy wybranym osobom. Dane te mogą również zostać wykorzystane w inny sposób (na przykład do oszacowania czasu poświęcanego na pracę nad określonym projektem).

Podsumowując:

*Wdrożenie polityki bezpieczeństwa sieciowego może być bardzo złożonym procesem, ponieważ wymaga zdefiniowania bezpieczeństwa sieci i komputerów w odniesieniu do poczynań użytkowników systemu oraz wartości zgromadzonych informacji.*

Definiowanie polityki bezpieczeństwa jest skomplikowane również z tego powodu, że organizacje muszą samodzielnie decydować o tym, które formy zabezpieczeń są najistotniejsze, a często muszą również wybierać między bezpieczeństwem rozwiązania i łatwością korzystania z niego. Oto kilka aspektów pracy sieciowej, które trzeba uwzględnić:

- **Spójność danych.** Dbanie o spójność danych oznacza ochronę przed zmianami. Czy dane dostarczane do odbiorcy są identyczne z wysłanymi przez nadawcę?
- **Dostępność danych.** Dostępność wiąże się z ochroną przed dezorganizacją pracy usługi. Czy dane są udostępniane osobom uprawnionym?

- **Poufność danych.** Poufność oznacza ochronę przed nieuprawnionym dostępem do danych (na przykład przed podsłuchiwaniem lub podszywaniem się pod osoby uprawnione). Czy dane muszą być zabezpieczone przed dostępem osób nieuprawnionych?
- **Prywatność.** Prywatność oznacza zachowanie anonimowości nadawcy. Czy dane nadawcy powinny być ujawniane?

### 30.4. Odpowiedzialność za dane i nadzór nad nimi

Poza ustaleniem odpowiedzi na wcześniejsze pytania organizacja będąca właścicielem sieci musi również określić zasady przekazywania odpowiedzialności za informacje oraz nadzorowania tego procesu. Z odpowiedzialnością za dane wiążą się dwa pojęcia:

- **Rejestrowanie zdarzeń.** Rejestrowanie zdarzeń oznacza prowadzenie dziennika nadzoru. Które grupy pracowników są odpowiedzialne za poszczególne wpisy? W jaki sposób grupa przechowuje informacje o dostępie do danych i wprowadzonych zmianach?
- **Autoryzacja.** Autoryzacja definiuje odpowiedzialność za każdą porcję danych oraz zasady przekazywania tej odpowiedzialności innym osobom. Kto jest odpowiedzialny za wybór miejsca składowania danych i w jaki sposób osoba za nie odpowiedzialna przydziela prawa dostępu i zmian?

Najbardziej newralgicznym elementem zarówno rejestracji zdarzeń, jak i autoryzacji jest **kontrola** — firma musi kontrolować dostęp do informacji w podobny sposób, jak kontroluje dostęp do fizycznych zasobów przedsiębiorstwa (biur, sprzętu, towarów). Kluczową operacją związaną z kontrolą jest **uwierzytelnienie**, czyli potwierdzenie tożsamości użytkownika. Założymy na przykład, że zgodnie z polityką bezpieczeństwa pracownik ma więcej uprawnień niż gość. Mechanizmy autoryzacji okażą się w takim przypadku bezużyteczne, jeśli system uwierzytelniania nie będzie mógł odróżnić gościa od pracownika. Uwierzytelnianie nie musi się ograniczać jedynie do sprawdzania użytkowników, może dotyczyć również komputerów, urządzeń sieciowych oraz aplikacji.

*Polityka autoryzacji jest bezużyteczna, jeśli nie istnieją mechanizmy uwierzytelniania, które jednoznacznie zweryfikują tożsamość osoby bądź jednostki przesyłającej żądania.*

### 30.5. Technologie związane z bezpieczeństwem

W sprzedaży jest dostępnych wiele rozwiązań, które realizują przeróżne zadania związane z zabezpieczaniem zarówno pojedynczych systemów, jak i grup komputerów organizacji. Mechanizmy implementowane w tego typu produktach zostały wymienione w tabeli 30.3. Każdy z nich został natomiast opisany w dalszych punktach rozdziału.

Tabela 30.3. Podstawowe rozwiązania umożliwiające wdrażanie polityki bezpieczeństwa

Rozwiązańe	Przeznaczenie
Generowanie skrótów	Zagwarantowanie spójności danych
Szyfrowanie	Zagwarantowanie poufności danych
Podpisy cyfrowe	Uwierzytelnienie wiadomości
Certyfikaty cyfrowe	Uwierzytelnienie użytkownika
Zapory sieciowe	Ochrona sieci
Systemy wykrywania włamań	Ochrona sieci
Szczegółowa inspekcja pakietów i skanowanie treści	Ochrona sieci
Wirtualne sieci prywatne (VPN)	Zagwarantowanie poufności danych

## 30.6. Generowanie skrótów — weryfikacja spójności danych i uwierzytelnianie

We wcześniejszych rozdziałach opisane zostały techniki generowania **bitów parzystości**, **sum kontrolnych** oraz wartości **CRC**, które służą do zabezpieczania danych przed przypadkowym przekłamaniem. Takie działania nie gwarantują jednak całkowicie spójności informacji. Po pierwsze, błąd w działaniu urządzenia lub oprogramowania może doprowadzić do zmiany zarówno danych, jak i sumy kontrolnej obejmującej te dane. Po drugie, jeśli zmiana jest efektem zaplanowanego ataku, osoba atakująca wygeneruje dla zmienionych informacji poprawną wartość kontrolną. Konieczne jest więc wdrożenie dodatkowych mechanizmów, które zagwarantują spójność danych i uniemożliwią ich celową modyfikację.

Jednym ze sposobów jest wygenerowanie **kodu uwierzytelniającego wiadomość** (MAC — ang. *Message Authentication Code*), którego nie można złamać lub podmienić. Typowe mechanizmy kodowania informacji bazują na **kryptograficznych funkcjach skrótu**. W rozwiązaniach tych często wykorzystuje się **tajny klucz**, znany jedynie nadawcy i odbiorcy. Uruchomiony przez nadawcę algorytm pobiera wiadomość i na podstawie klucza generuje wartość skrótu ( $H$ ). Następnie przesyła skrót  $H$  wraz z wiadomością. Wartość  $H$  jest krótkim ciągiem bitowym, którego długość nie zależy od długości wiadomości. Odbiorca wykorzystuje ten sam klucz do obliczenia wartości skrótu odebranej informacji, a następnie porównuje obydwa skróty. Jeśli są jednakowe, wiadomość nie została zmodyfikowana. Osoba atakująca nie zna klucza, więc ewentualne wprowadzone przez nią zmiany zostałyby wykryte. Wartość skrótu uwierzytelnia więc wiadomość, ponieważ poprawna wartość  $H$  pozwala na stwierdzenie, że informacja nie została zmieniona.

## 30.7. Kontrola dostępu i hasła

Mechanizmy **kontroli dostępu** weryfikują uprawnienia użytkowników i programów do operowania określonymi danymi. W niektórych systemach operacyjnych implementuje się **listy kontroli dostępu** (ACL — ang. *Access Control List*), które wskazują, kto jest uprawniony do korzystania z określonego obiektu. W innych systemach dostęp do chronionych zasobów wymaga wprowadzenia **hasła**. Gdy użytkownik chce odwołać się do takiego zasobu, musi najpierw podać właściwe hasło.

Stosując listy kontroli dostępu i hasła w systemach sieciowych, trzeba pamiętać o wykonyaniu dodatkowych działań, które zapobiegą przypadkowemu ujawnieniu danych. Na przykład jeśli użytkownik pracujący w sieci zdalnej prześle przez sieć niezaszyfrowane hasło, każdy, kto przechwytuje dane sieciowe, może je bez trudu odczytać. Podслушаивание jest szczególnie łatwe w sieciach bezprzewodowych, ponieważ nie ma potrzeby tworzenia fizycznego połączenia między urządzeniami (osoby pracujące w zasięgu działania nadajnika mogą przechwytywać kopie wszystkich pakietów).

Same hasła muszą być trudne do odgadnięcia. Włamywacze często bowiem korzystają ze zautomatyzowanych narzędzi do łamania haseł. Z tego powodu administratorzy definiują pewne reguły doboru hasła — określają jego minimalną długość bądź zabraniają wykorzystywania typowych słów (tj. słów, które można znaleźć w słowniku).

## 30.8. Szyfrowanie — podstawowa technika zabezpieczeń

**Kryptografia** jest podstawową dziedziną nauki, która znajduje zastosowanie w zabezpieczaniu sieci. Narzędzia kryptograficzne umożliwiają zapewnienie poufności danych (nazywanej czasami **prywatnością**), wiarygodności oraz spójności. Eliminują również zagrożenie związane z powtarzaniem komunikatów. Ogólna idea operacji kryptograficznych sprowadza się do tego, że nadawca wykorzystuje algorytm szyfrujący do randomizacji bitów danych w sposób, który umożliwi ich odtworzenie jedynie w komputerze wybranego odbiorcy. Osoba, która przechwyci taką wiadomość, nie będzie mogła odczytać zawartych w niej informacji. Zaszyfrowany komunikat może również zawierać dane na temat długości wiadomości, dzięki czemu podzielenie informacji nie pozostanie niezauważone.

Szyfrowanie nieroźłącznie wiąże się z czterema następującymi pojęciami:

- tekst jawnny — wiadomość oryginalna (przed zaszyfrowaniem);
- szyfrogram — wiadomość po zaszyfrowaniu;
- klucz szyfrujący — krótki串 bitów wykorzystywany do zaszyfrowania wiadomości;
- klucz deszyfrujący — krótki串 bitów wykorzystywany do rozszyfrowania wiadomości.

W niektórych rozwiązaniach klucze szyfrujące i deszyfrujące są takie same. W innych są różne.

W ujęciu matematycznym szyfrowanie jest funkcją (*szyfr*), która pobiera dwa parametry: klucz ( $K$ ) oraz tekst jawnny ( $T$ ). Wynikiem jej działania jest szyfrogram ( $Sz$ ), czyli zaszyfrowana wersja wiadomości:

$$Sz = \text{syfr}(K_1, T)$$

Funkcja rozszyfrowująca odwraca tę zależność i generuje pierwotną wiadomość<sup>85</sup>:

$$T = \text{deszyfr}(K_1, Sz)$$

Z matematycznego punktu widzenia funkcja *deszyfr* jest odwrotnością funkcji *syfr*:

$$T = \text{deszyfr}(K_1, \text{syfr}(K_1, T))$$

## 30.9. Szyfrowanie z użyciem klucza prywatnego

Liczną grupę technik szyfrowania można podzielić na zasadnicze grupy, w zależności od sposobu wykorzystania kluczy szyfrujących:

- algorytmy klucza prywatnego,
- algorytmy klucza publicznego.

W systemie **klucza prywatnego** każda para komunikujących się ze sobą jednostek dysponuje kluczem, który pełni rolę **klucza szyfrującego** i **klucza deszyfrującego**. Klucz musi być tajny, gdyż ujawnienie go trzeciej jednostce oznaczałoby, że będzie ona mogła rozszyfrowywać wszystkie wiadomości wymieniane między uprawnionymi do tego stacjami. Techniki szyfrowania z użyciem klucza prywatnego są mechanizmami **symetrycznymi** (każda strona może odbierać i wysyłać wiadomości z zastosowaniem klucza). W czasie wysyłania wiadomości klucz służy do utworzenia szyfrogramu, który jest później przesyłany przez sieć. Po dostarczeniu komunikatu odbiorca wykorzystuje tajny klucz do przekształcenia szyfrogramu w wiadomość oryginalną (tekst jawnny). W systemie klucza prywatnego nadawca i odbiorca używają tego samego klucza  $K$ , co oznacza, że prawdziwa jest zależność:

$$T = \text{deszyfr}(K, \text{syfr}(K, T))$$

## 30.10. Szyfrowanie z użyciem klucza publicznego

Alternatywą dla mechanizmu klucza prywatnego jest **szyfrowanie z użyciem klucza publicznego**. W rozwiązaniu tym każdej jednostce przypisywane są dwa klucze. Na potrzeby dalszych rozważań możemy przyjąć, że wspomnianą jednostką jest określony użytkownik systemu. Jeden z kluczy tego użytkownika jest nazywany **kluczem prywatnym** (niejawnym), a drugi **kluczem publicznym** (udostępnianym bez ograniczeń wraz z nazwą użytkownika). Stosowana w tym przypadku funkcja szyfrująca ma tę własność, że tekst jawny zaszyfrowany za pomocą klucza publicznego może zostać rozszyfrowany jedynie za pomocą klucza prywatnego, a tekst jawny zaszyfrowany z użyciem klucza prywatnego nie może zostać rozszyfrowany jedynie za pomocą klucza publicznego.

---

<sup>85</sup> Do rozszyfrowania wiadomości może być potrzebny ten sam klucz, który został użyty do zaszyfrowania tekstu, lub inny.

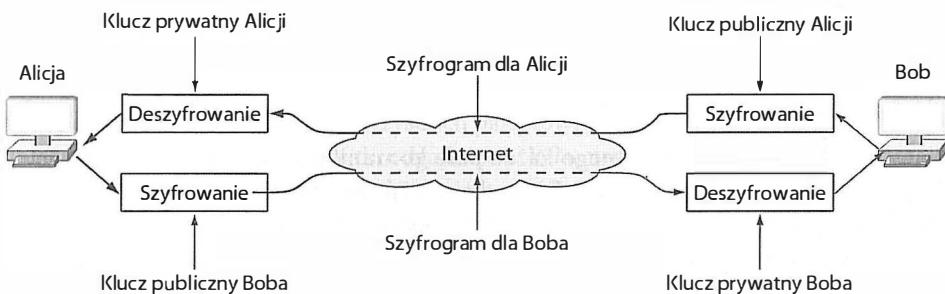
Zależność między funkcjami szyfrującymi i rozszyfrowującymi można wyrazić matematycznie. Oznaczmy tekst jawnny jako  $T$ , klucz publiczny pierwszego użytkownika jako  $pub\_u1$ , a klucz prywatny tego samego użytkownika jako  $pryw\_u1$ . Prawdziwe byłyby wówczas następujące równania:

$$T = deszyfr(pub\_u1, \ szyfr(pryw\_u1, T))$$

oraz

$$T = deszyfr(pryw\_u1, \ szyfr(pub\_u1, T))$$

Na rysunku 30.3 przedstawiono wymianę zaszyfrowanych komunikatów. Z ilustracji wynika również, dlaczego systemy klucza publicznego są klasyfikowane jako algorytmy **asymetryczne**.



Rysunek 30.3. Asymetria szyfrowania w użyciu klucza publicznego

Ujawnienie klucza publicznego jest zupełnie bezpieczne, ponieważ funkcje szyfrujące i rozszyfrowujące mają własność **jednokierunkowości**. Zatem udostępnienie komuś klucza publicznego nie oznacza, że osoba ta będzie mogła przejąć informację zaszyfrowaną za pomocą klucza prywatnego.

Szyfrowanie z użyciem kluczy publicznych sprawdza się w rozwiązaniach wymagających poufności. Użytkownik chcący przesłać wiadomość w sposób bezpieczny wykorzystuje do jej zaszyfrowania klucz publiczny odbiorcy. Ewentualne przechwycenie komunikatu w czasie transportowania go przez sieć nie wiąże się z ryzykiem ujawnienia treści, ponieważ do jego rozszyfrowania potrzebny jest klucz prywatny odbiorcy. Dane zostaną więc przekazane w sposób poufnny, gdyż tylko odbiorca może rozszyfrować wiadomość.

### 30.11. Uwierzytelnianie z wykorzystaniem podpisów cyfrowych

Algorytmy szyfrujące znajdują zastosowanie również w procesie uwierzytelniania nadawcy wiadomości. Technika ta jest nazywana generowaniem **podpisu cyfrowego**. Aby podpisać wiadomość, nadawca szyfuje ją z użyciem klucza znanego jedynie sobie<sup>86</sup>. Odbiorca wie, kto wysłał wiadomość, ponieważ jedynie nadawca ma klucz potrzebny do zaszyfro-

<sup>86</sup> Zachowanie poufności nie jest obowiązkowe. Wiadomość nie musi być zaszyfrowana. Można również zastosować wydajniejszy wariant generowania podpisu cyfrowego, w którym szyfrowana jest jedynie sama wartość skrótu.

wania komunikatu. Aby uniemożliwić kopiowanie i powtarzanie wiadomości, w oryginalnym komunikacie można zapisać datę i czas utworzenia wiadomości.

Zastanówmy się nad zastosowaniem mechanizmu klucza publicznego w przetwarzaniu podpisów cyfrowych. Aby podpisać wiadomość, nadawca szyfruje ją z użyciem własnego klucza prywatnego. Z kolei odbiorca, chcąc zweryfikować podpis, odczytuje klucz publiczny nadawcy i wykorzystuje go do rozszyfrowania komunikatu. Tylko nadawca zna swój klucz prywatny. Zatem rozszyfrowanie wiadomości za pomocą klucza publicznego nadawcy jest możliwe tylko wtedy, gdy on sam ją zaszyfrował.

Wiadomość można również zaszyfrować dwukrotnie, co ją uwierzytelnia i zapewnia zachowanie poufności. W pierwszym kroku wiadomość jest podpisywana i szyfrowana za pomocą klucza prywatnego nadawcy. W drugim etapie dane są szyfrowane z użyciem klucza publicznego odbiorcy. Wykonanie obydwu operacji można zapisać jako:

$$X = \text{szysfr}(\text{pub\_u2}, \text{szysfr}(\text{pryw\_u1}, T))$$

W powyższym równaniu  $T$  odpowiada przesyłanemu tekstowi jawnemu,  $X$  reprezentuje szfyrogram powstały po dwukrotnym zaszyfrowaniu informacji, a zmienne  $\text{pryw\_u1}$  i  $\text{pub\_u2}$  symbolizują odpowiednio klucz prywatny nadawcy oraz klucz publiczny odbiorcy.

Po stronie odbiorczej realizowany jest odwrotny proces. Najpierw odbiorca używa własnego klucza prywatnego do rozszyfrowania dostarczonego bloku danych. W wyniku tego działania uzyskuje wiadomość podpisany cyfrowo. W kolejnym kroku odbiorca ponownie rozszyfrowuje wiadomość, ale tym razem za pomocą klucza publicznego nadawcy. Cały proces można opisać następująco:

$$T = \text{szysfr}(\text{pub\_u1}, \text{szysfr}(\text{pryw\_u2}, X))$$

W równaniu tym  $X$  reprezentuje szfyrogram przesłany przez sieć,  $T$  odpowiada oryginalnemu tekstowi jawnemu, a zmienne  $\text{pryw\_u2}$  i  $\text{pub\_u1}$  symbolizują odpowiednio klucz prywatny odbiorcy oraz klucz publiczny nadawcy.

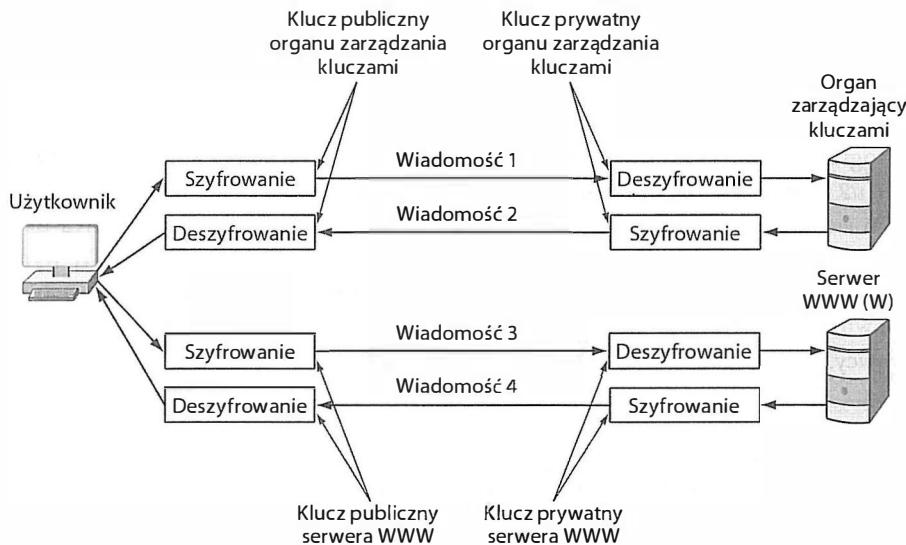
Jeśli w wyniku tych operacji wiadomość ma czytelną postać, jest pewne, że została dostarczona w sposób poufny i że jest autentyczna. Informacje z pewnością zostały odczytane tylko przez właściwego odbiorcę, ponieważ jedynie on dysponuje kluczem prywatnym potrzebnym do przejścia pierwszego etapu rozszyfrowywania. Wiadomo również, że pochodzą od zaufanego nadawcy, ponieważ tylko on ma klucz prywatny, który musiał posłużyć do zaszyfrowania wiadomości, skoro udało się ją rozszyfrować z użyciem klucza publicznego tego nadawcy.

## 30.12. Organa zarządzające kluczami i certyfikaty cyfrowe

Jedną z ważniejszych kwestii dotyczących stosowania technologii klucza publicznego jest sposób pozyskiwania kluczy publicznych. Choć nic nie stoi na przeszkodzie, aby udostępnić je w standardowy sposób (podobnie jak informacje w książce telefonicznej), takie postępowanie jest podatne na błędy, ponieważ użytkownicy musieliby samodzielnie wpisywać odpowiednie wartości kluczów. Czy istnieje więc automatyczny system dystrybucji kluczy publicznych? Oczywiście, cały proces musi być odpowiednio zabezpieczony. Udostępnienie użytkownikowi klucza publicznego o niewłaściwej treści spowodowałoby

bowiem zerwanie relacji zaufania i odrzucenie kolejnych zaszyfrowanych wiadomości. Zagadnienie jest znane pod nazwą **problemu dystrybucji kluczy**, a utworzenie niezawodnego mechanizmu dystrybucji kluczy jest główną przeszkodą w upowszechnianiu się systemów klucza publicznego.

Opracowano kilka mechanizmów dystrybucji kluczy, włącznie z rozwiązańem bazującym na systemie nazw domenowych. W każdym przypadku stosowana jest pewna zasada — znajomość jednego klucza (klucza organu zarządzającego kluczami) pozwala na pobranie innych kluczy w sposób bezpieczny. Administrator musi więc zdefiniować tylko jeden klucz. Na rysunku 30.4 zaprezentowana została wymiana danych podczas pobierania odwołania do nowego serwera WWW (*W*).



Rysunek 30.4. Wykorzystanie organu zarządzającego kluczami do pobrania klucza publicznego serwera

W przedstawionym przykładzie użytkownik dąży do zabezpieczenia wymiany danych z serwerem *W*. Każda z czterech przesyłanych wiadomości jest poufna. Wiadomość 1 może zostać odczytana jedynie przez komputer organu zarządzającego kluczami, ponieważ została zaszyfrowana jego kluczem publicznym (ogólnie dostępnym). Wiadomość 2 musiała zostać wygenerowana przez ten sam komputer, gdyż tylko on ma klucz prywatny odpowiadający danemu kluczowi publicznemu. Po uzyskaniu klucza publicznego serwera *W* użytkownik może wysłać poufne żądanie z przekonaniem, że tylko wybrany serwer może na nie odpowiedzieć (tylko on dysponuje kluczem prywatnym).

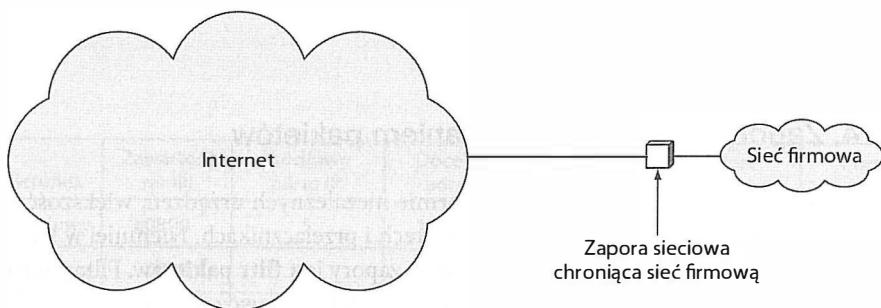
Choć istnieje wiele podobnych rozwiązań, wszystkie mają wspólną cechę:

*Utworzenie bezpiecznego systemu dystrybucji kluczy wymaga ręcznego wprowadzenia tylko jednego klucza publicznego.*

### 30.13. Zapory sieciowe

Szyfrowanie rozwiązuje wiele problemów z bezpieczeństwem pracy sieciowej, ale nie wszystkie. Aby chronić sieci przed niepożądany ruchem internetowym, konieczne jest zastosowanie **internetowych zapór sieciowych** (firewalli internetowych). Podobnie jak klasyczne zapory sieciowe firewalli internetowe uniemożliwiają rozprzestrzenienie się zagrożeń internetowych wśród komputerów firmowych.

Zapora sieciowa jest instalowana pomiędzy siecią przedsiębiorstwa a pozostałą częścią internetu. Wszystkie pakiety wychodzące z sieci firmowej lub do niej przesyłane muszą zostać przekazane przez zaporę sieciową. Budowa opisanego systemu została przedstawiona na rysunku 30.5.



Rysunek 30.5. Instalacja zapory sieciowej pomiędzy siecią firmową i internetem

Jeśli dana organizacja jest przyłączona do internetu za pośrednictwem wielu łącz, zapory sieciowe muszą zostać zainstalowane na każdym takim łączu. Ponadto wszystkie muszą zostać skonfigurowane w taki sposób, aby wymuszały stosowanie postanowień zawartych w polityce bezpieczeństwa. Sama zaporę musi być odporna na ingerencję osób trzecich. A zatem:

- Cały ruch wchodzący do sieci firmowej musi być przekazywany przez zaporę sieciową.
- Cały ruch opuszczający sieć firmową musi być przekazywany przez zaporę sieciową.
- Zaporę sieciową wdraża postanowienia polityki bezpieczeństwa poprzez odrzucanie pakietów nieodpowiadających tej polityce.
- Zaporę sieciową sama musi być odporna na ataki zewnętrzne.

Zapory sieciowe są najważniejszymi komponentami systemu zabezpieczeń, które stosuje się na połączeniu dwóch nieufających sobie organizacji. Zainstalowanie zapory sieciowej na jednym końcu połączenia z siecią zewnętrzną skutkuje wyznaczeniem **granicy systemu zabezpieczeń** i zapobiega zakłócaniu pracy komputerów wewnętrznych. Dzięki jej stosowaniu użytkownicy sieci zewnętrznych nie mogą odnajdywać komputerów organizacji, zalewając ich niepożdanymi pakietami oraz atakować komputerów datagramami IP, które powodują błędne działanie jednostek (na przykład ich zawieszanie). Zapory sieciowe uniemożliwiają również wysyłanie chronionych danych (na przykład w przypadku zainfekowania komputera wirusem, który wysyła zawartość dysku poza sieć organizacji).

Zapory sieciowe mają jeszcze jedną przewagę nad innymi systemami zabezpieczeń — centralizują kontrolę ruchu, a tym samym istotnie podnoszą poziom bezpieczeństwa. Ich brak oznaczałby konieczność zainstalowania odpowiedniego oprogramowania zabezpieczającego na każdym komputerze wewnętrznym. Nie bez znaczenia jest także konieczność przestrzegania na każdym komputerze wspólnej polityki bezpieczeństwa. Koszt zatrudnienia informatyków odpowiedzialnych za administrowanie wszystkimi komputerami byłby bardzo duży, a nie można polegać na tym, że pracownicy sami właściwie skonfigurują swoje komputery. Zapory sieciowe pozwalają administratorom sieci na ograniczenie ruchu internetowego jedynie do niewielkiej liczby stacji i zlecenie odpowiedniemu personelowi dbania o konfigurację i monitorowania wybranej grupy. W szczególnych przypadkach dostęp do sieci zewnętrznej może zostać ograniczony do jednego komputera wewnętrznego. Dzięki temu firma może zaoszczędzić pieniądze i zapewnić większe bezpieczeństwo danych.

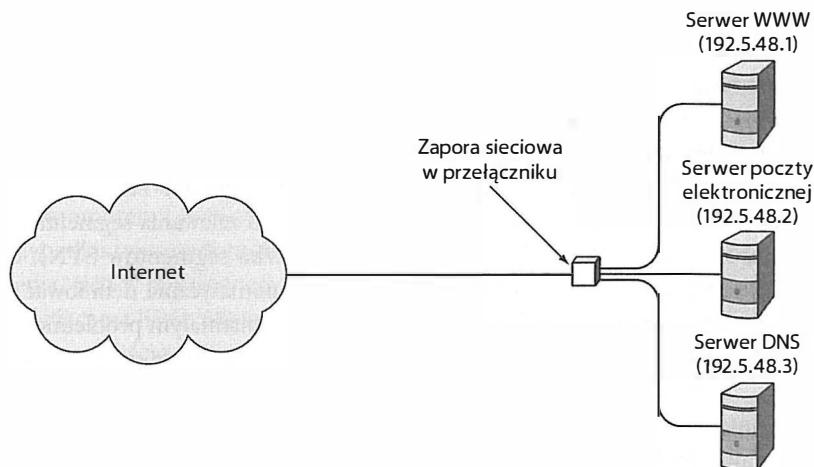
### 30.14. Zapory sieciowe z filtrowaniem pakietów

Choć zapory sieciowe można budować w formie niezależnych urządzeń, większość rozwiązań tego typu jest implementowana w routeraх i przełącznikach. Niemniej w każdym z wymienionych przypadków podstawą działania zapory jest **filtr pakietów**. Filtr pakietów składa się z wielu konfigurowalnych mechanizmów, które sprawdzają pola w nagłówkach wszystkich pakietów i podejmują decyzję o przekazaniu danego pakietu lub jego odrzuceniu. Administrator definiuje filtr pakietów, określając, które pakiety mogą zostać przekazane w danym kierunku (znacznie bezpieczniejszym rozwiązaniem jest określanie, które pakiety są przenoszone, niż definiowanie pakietów, które są odrzucane).

W przypadku sieci TCP/IP definicja filtra pakietów obejmuje zazwyczaj wartość typu ramki 0800 (odpowiadającą protokołowi IP), **źródłowy adres IP** lub **docelowy adres IP** bądź obydwa, typ datagramu oraz numer portu. Na przykład aby umożliwić użytkownikom internetu odwołania do firmowego serwera WWW, filtr pakietów musiałby pozwalać na dostarczanie ramek, które zawierają datagram IP (z segmentem TCP) wysłany przez komputer o dowolnym adresie źródłowym i dowolnym porcie źródłowym, kierowany na adres IP serwera WWW na port 80.

Dzięki temu, że administrator może definiować zarówno docelowe, jak i źródłowe adresy IP oraz docelowe i źródłowe numery portów, filtr pakietów doskonale nadaje się do określania, jakie usługi będą dostępne na wskazanych komputerach. Technikę tę można by zastosować do zdefiniowania takiej polityki ruchu, która pozwoliłaby na funkcjonowanie usługi WWW na jednym komputerze, serwera poczty elektronicznej na drugim komputerze oraz serwera DNS na trzecim komputerze. Oczywiście, należałoby zdefiniować analogiczne reguły przenoszenia pakietów dla ruchu wychodzącego. Opisana konfiguracja została przedstawiona na rysunku 30.6.

Możliwość wybierania pakietów wchodzących do sieci na podstawie informacji o usługach docelowych pozwala administratorowi decydować o tym, które usługi będą „widoczne” z sieci zewnętrznej. Dzięki temu przypadkowe (lub celowe) uruchomienie serwera pocztowego w innym komputerze będzie oznaczało udostępnienie go w internecie.



Kierunek	Zawartość ramki	Źródłowy adres IP	Docelowy adres IP	Zawartość datagramu	Port źródłowy	Port docelowy
Wejściowy	0800	*	192.5.48.1	TCP	*	80
Wejściowy	0800	*	192.5.48.2	TCP	*	25
Wejściowy	0800	*	192.5.48.3	TCP	*	53
Wejściowy	0800	*	192.5.48.3	UDP	*	53
Wyjściowy	0800	192.5.48.1	*	TCP	80	*
Wyjściowy	0800	192.5.48.2	*	TCP	25	*
Wyjściowy	0800	192.5.48.3	*	TCP	53	*
Wyjściowy	0800	192.5.48.3	*	UDP	53	*

Rysunek 30.6. Konfiguracja zapory sieciowej chroniącej sieć z trzema serwerami  
(znak gwiazdki reprezentuje dowolną wartość)

Podsumowując:

Zapory sieciowe wykorzystują filtrowanie pakietów do blokowania niepożądanych form komunikacji. Każda definicja filtru odnosi się do pól nagłówkowych i obejmuje źródłowy i docelowy adres IP, numery portów oraz typ protokołu transportowego.

## 30.15. Systemy wykrywania włamań

System wykrywania włamań (IDS — ang. *Intrusion Detection System*) monitoruje wszystkie pakiety dostarczane do sieci firmowej i informuje administratora o przypadkach naruszenia zasad bezpieczeństwa. Rozwiązań IDS wyznaczają kolejny poziom zabezpieczeń — nawet

gdy zapora sieciowa zapobiegnie atakowi, system IDS powiadomi administratora o tym zdarzeniu.

Większość systemów IDS można skonfigurować w taki sposób, aby reagowała na określone rodzaje ataków. Doskonały przykład ich zastosowania to wykrywanie prób **skanowania portów**. Atak ten polega na wysyłaniu serii datagramów UDP na kolejne porty protokołu UDP lub inicjowaniu połączeń TCP na kolejnych portach protokołu TCP. Odpowiednia konfiguracja systemu IDS zapewnia również wykrywanie prób zalewania segmentami SYN (sprawdzanie, czy z jednego źródła nie napływa duża liczba segmentów SYN). Często jednostki IDS są połączone z **zaporami sieciowymi**, aby automatycznie definiować reguły filtrowania. Zamiast jedynie powiadamiać administratora o zaistniałym problemie, system IDS dodaje do zapory sieciowej nową regułę, która powoduje zablokowanie pakietów z określonego źródła. W ten sposób można na przykład zablokować zalew sieci pakietami SYN. Celem automatyzacji działań jest zwiększenie szybkości reakcji. Interwencja człowieka zajmuje co najmniej kilka sekund od odebrania powiadomienia. Tymczasem w gigabitowym Etherenecie jedna sekunda wystarczy, aby przesłać 50 000 pakietów. Szybka reakcja jest więc kluczowym czynnikiem w zabezpieczeniu sieci przed przeciążeniem.

Główna różnica w działaniu systemów IDS i zapór sieciowych wynika ze sposobu weryfikacji pakietów. Mechanizmy IDS rejestrują **informacje o stanie** połączeń. W przeciwieństwie do zapór sieciowych, które porównują każdy pakiet niezależnie z zestawem reguł, weryfikują historię pakietów. Działanie zapory sieciowej ogranicza się zatem jedynie do sprawdzenia, czy dany rodzaj pakietów SYN jest dozwolony. Jednostka IDS może dodatkowo określić, ile pakietów SYN wygenerowało jedno źródło. Oczywiście, z uwagi na większe niż w przypadku zapory sieciowej zapotrzebowanie na moc procesora oraz pamięć operacyjną systemy IDS nie przetwarzają tak dużej liczby pakietów na sekundę.

### 30.16. Skanowanie treści i szczegółowa inspekcja pakietów

Zapory sieciowe, mimo że eliminują wiele zagrożeń, mają jedno poważne ograniczenie — weryfikują jedynie pola zawarte w nagłówkach pakietów. Nie mogą weryfikować przesyłanych treści. O tym, jak duże znaczenie ma weryfikacja zawartości pakietów, świadczy problem wirusów komputerowych. Jedna z najczęściej stosowanych metod zarażania komputerów wirusami polega na przesyłaniu ich jako załączników do poczty elektronicznej. Osoba atakująca wysyła wiadomość e-mail z dołączonym programem. Jeśli adresat otworzy załącznik, program uruchomi się w jego komputerze, instalując nieznane oprogramowanie, które może być **złośliwe**, tak jak wirus.

W jaki sposób administratorzy zapobiegają instalowaniu wirusów? Korzystają z systemów **analizy treści**. Istnieją dwie kategorie tego typu mechanizmów:

- skanowanie plików,
- szczegółowa inspekcja pakietów (DPI — ang. *Deep Packet Inspection*).

**Skanowanie plików.** Najmniej skomplikowana metoda analizy zawartości sprowadza się do skanowania całych plików. Skanowanie plików jest doskonale znaną techniką, z której korzysta oprogramowanie zabezpieczające większości komputerów PC. Działanie

mechanizmu polega na odczytaniu zawartości pliku i sprawdzeniu, czy nie występują w nim kombinacje bajtów charakterystyczne dla znanych zagrożeń. Skanery antywirusowe poszukują w ten sposób ciągów bajtowych nazywanych **sygnaturami**. Firma sprzedająca oprogramowanie antywirusowe zbiera kopie wirusów, zapisuje każdą z nich w pliku, wyszukuje charakterystyczne sekwencje bajtowe i na ich podstawie tworzy bazę sygnatur. Gdy użytkownik uruchamia skaner antywirusowy, program przeszukuje pliki zapisane na dysku komputera i porównuje ich zawartość z elementami listy. Metoda ta sprawdza się w przypadku typowych problemów. Oczywiście, skanowanie plików może dawać błędne trafienia, jeśli w poprawnym pliku znajduje się kombinacja bajtowa zgodna z sygnaturą występującą na liście. Może również dojść do przeoczenia wirusa, jeśli nie został on zdefiniowany na liście podejrzanych ciągów.

**Szczegółowa inspekcja pakietów (DPI).** Druga technika analizy treści odnosi się do pakietów, a nie do plików. Zamiast analizować nagłówki pakietów przesyłanych do sieci wewnętrznej, mechanizm DPI sprawdza zawartość pola danych pakietu. Nie oznacza to, że pomija nagłówki. W wielu przypadkach weryfikacja zawartości pakietu nie jest możliwa, jeśli nie zostały wcześniej sprawdzone odpowiednie pola nagłówka.

Jako przykład zastosowania mechanizmu DPI warto przeanalizować przypadek ataku, w którym nieznacznie zmieniony sposób zapisu nazwy domenowej ma na celu oszukanie użytkownika i skłonienie go do skorzystania z niebezpiecznego serwisu. Organizacje chroniące pracowników przed tego rodzaju problemami opracowują **czarne listy** adresów URL stanowiących zagrożenie dla bezpieczeństwa systemów. Administratorzy sieci uruchamiają wówczas **pośredniczące serwery WWW** (tj. dodatkowe systemy WWW, które sprawdzają adresy URL przed wysłaniem żądania dostarczenia strony) i nakładają użytkowników przeglądarki do korzystania z nich. Rozwiązaniem alternatywnym jest zastosowanie filtra DPI, który będzie skanował wszystkie wychodzące pakiety, poszukując w nich żądań HTTP kierowanych do zabronionych witryn.

Największą wadą mechanizmu DPI jest zwiększyły narzucona obliczeniowa. Pole danych ramki ethernetowej jest bowiem ponad dwadzieścia razy większe niż nagłówek pakietu, co oznacza dwadzieścia razy większe zużycie zasobów procesora niż w przypadku weryfikacji nagłówków. Poza tym obszar danych nie jest podzielony na pola o stałym rozmiarze. Z tego powodu filtr DPI musi interpretować treść w czasie jej sprawdzania.

*Z uwagi na konieczność analizowania obszarów danych pakietów (które są znacznie większe niż ich nagłówki i nie są podzielone na pola o stałym rozmiarze) systemy szczegółowej inspekcji pakietów są przeznaczone do stosowania w sieciach o mniejszej szybkości transmisji danych.*

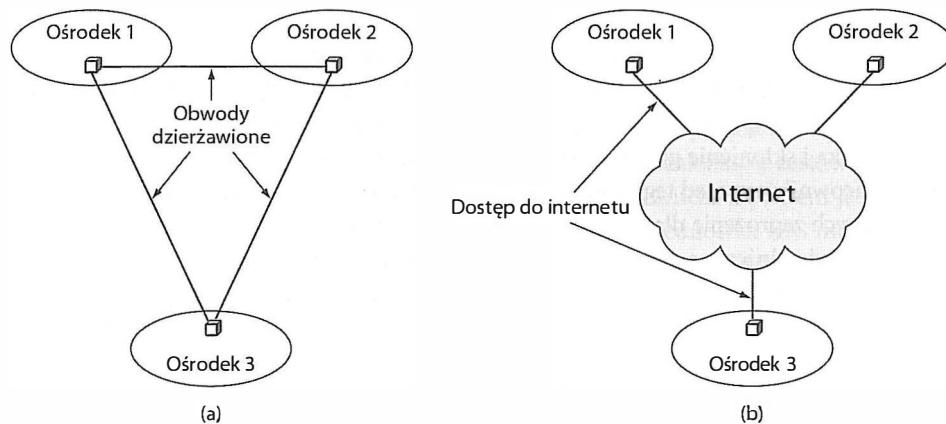
## 30.17. Wirtualne sieci prywatne (VPN)

**Wirtualna sieć prywatna** (VPN — ang. *Virtual Private Network*) jest jednym z najważniejszych rozwiązań, które zapewniają dostęp do sieci wewnętrznej organizacji (do intranetu) ze zdalnych lokalizacji z użyciem szyfrowania. Technologia ta została opracowana

jako tani sposób na łączenie ośrodków korporacji odległych od siebie geograficznie. Chcąc zrozumieć powody jej powstania, trzeba zastanowić się nad alternatywnymi metodami łączenia ośrodków:

- **Prywatne połączenia sieciowe.** Organizacja dzierżawi obwody transmisji danych, które łączą jej ośrodki. Na końcach każdego połączenia dzierżawionego pracują routery. Dane są przekazywane bezpośrednio między routерem jednej sieci organizacji a routерem drugiej sieci.
- **Połączenia za pośrednictwem publicznego internetu.** Każdy ośrodek podpisuje umowę z lokalną firmą ISP na świadczenie usług internetowych. Dane między sieciami korporacyjnymi są przesyłane przez internet.

Na rysunku 30.7 przedstawiono obydwa rozwiązania w odniesieniu do przedsiębiorstwa o trzech ośrodkach.



Rysunek 30.7. Połączenie ośrodków za pomocą obwodów dzierżawionych (a) i internetu (b)

Największą zaletą stosowania obwodów dzierżawionych jest zachowanie całkowitej prywatności sieci (a tym samym poufności wymiany danych). Takie rozwiązanie gwarantuje, że żadne inne przedsiębiorstwo nie ma dostępu do łącza. Nie może więc odczytać danych, które są przekazywane między sieciami ośrodków. Z kolei największą zaletą wykorzystania internetu jest niski koszt użytkowania — zamiast płacić za dostępność linii łączających ośrodki, firma opłaca rachunek za dostęp do internetu (w każdym z ośrodków). Niestety, dostawca usług internetowych nie może zagwarantować poufności danych. Podczas przesyłania informacji ze stacji źródłowej do docelowej pakiety są przekazywane przez sieci pośrednie, które często są współdzielone przez wiele organizacji. Trzeba się więc liczyć z tym, że osoby spoza firmy mogą otrzymywać kopie transportowanych datagramów i mogą zapoznać się z ich zawartością.

Mechanizmy VPN wykorzystują najlepsze rozwiązania z dwóch dziedzin komunikacji sieciowej — używają internetu do transmisji danych i zabezpieczają dane w taki sposób, aby nie mogły zostać wykorzystane przez osoby spoza organizacji. Zastępują kosztowne

łącza dzierżawione szyfrowanymi połączniami internetowymi (wszystkie pakiety przesyłane między ośrodkami przedsiębiorstwa są szyfrowane przed wysłaniem do sieci publicznej).

Aby zwiększyć odporność sieci VPN na ataki, administratorzy systemów firmowych wykorzystują specjalnie przygotowane routery VPN oraz zapory sieciowe, które uniemożliwiają routerom VPN akceptowanie nieautoryzowanych pakietów. Jako przykład rozważmy przypadek zaprezentowany na rysunku 30.7b, zakładając, że każdy router jest przystosowany do pracy z połączniami VPN (przymijmy, że w każdej sieci funkcjonuje osobny router, który obsługuje klasyczny ruch kierowany do i z internetu). Zapora sieciowa zainstalowana w ośrodku 1 odpowiadałaby wówczas za sprawdzanie, czy nadchodzące pakiety mają źródłowe adresy IP odpowiadające adresom routerów VPN w ośrodkach 2 lub 3. Analogicznie, routery w pozostałych sieciach ograniczałyby ruch jedynie do pakietów z innych sieci firmowych. Takie rozwiązanie uodparnia system na ataki z podmianą adresów oraz ataki DoS.

## 30.18. Wykorzystanie technologii VPN w pracy zdalnej

Choć technologia VPN została opracowana jako sposób na łączenie odległych sieci, zyskała również ogólną popularność w **pracy na odległość** (czyli w pracy ze zdalnych lokalizacji). Obecnie funkcje VPN są realizowane przez dwa rodzaje punktów końcowych:

- niezależne urządzenia,
- oprogramowanie VPN.

**Niezależne urządzenia.** Firma wyposaży pracownika w specjalne urządzenie, nazywane niekiedy **routerem VPN**. Urządzenie to łączy się z internetem, automatycznie ustanawia bezpieczne połączenie z serwerem VPN pracującym w siedzibie przedsiębiorstwa i zaczyna przekazywać ruch z firmowej sieci LAN, pozwalając użytkownikowi na komunikowanie się z innymi komputerami i telefonami IP. W warstwie logicznej moduł VPN rozszerza sieć przedsiębiorstwa o sieć użytkownika, zapewniając komputerom przyłączonym do urządzenia VPN możliwość pracy na takich samych zasadach, jakie obowiązują w sieci korporacyjnej. Zatem w czasie uruchamiania komputer użytkownika pobiera adres IP z serwera firmowego. Również tablica routingu jest konfigurowana w taki sposób, jakby pracował on w sieci przedsiębiorstwa. Każdy pakiet wysłany przez komputer jest szyfrowany przez moduł VPN i przesyłany za pośrednictwem internetu do sieci firmowej. Analogicznie, każdy pakiet dostarczany do użytkownika jest transportowany przez internet, rozszыfrowywany w urządzeniu VPN i przesyłany do jego komputera.

**Oprogramowanie VPN.** Choć oddzielne urządzenia doskonale sprawdzają się u użytkowników, którzy pracują w domu lub odległym biurze, bywają bardzo kłopotliwe dla osób podróżujących. W takich przypadkach firmy dostarczają **oprogramowanie VPN**, które jest uruchamiane w komputerze pracownika. Zadanie użytkownika sieci polega wówczas na przyłączeniu komputera do internetu i uruchomieniu aplikacji VPN. Działający program stanowi element pośredniczący w wymianie danych między komputerem i internetem. Przechwytuje wszystkie wychodzące i przychodzące pakiety. Szyfruje dane kierowane do korporacyjnego serwera VPN i rozszыfrowuje informacje dostarczane z tego serwera.

### 30.19. Szyfrowanie pakietów a tunelowanie

Z wcześniejszych rozważań na temat technologii VPN rodzi się interesujące pytanie: w jaki sposób dane powinny być szyfrowane przed przesłaniem przez internet? Dostępne są trzy rozwiązania:

- szyfrowanie pola danych,
- tunelowanie IP-w-IP,
- tunelowanie IP-w-TCP.

{

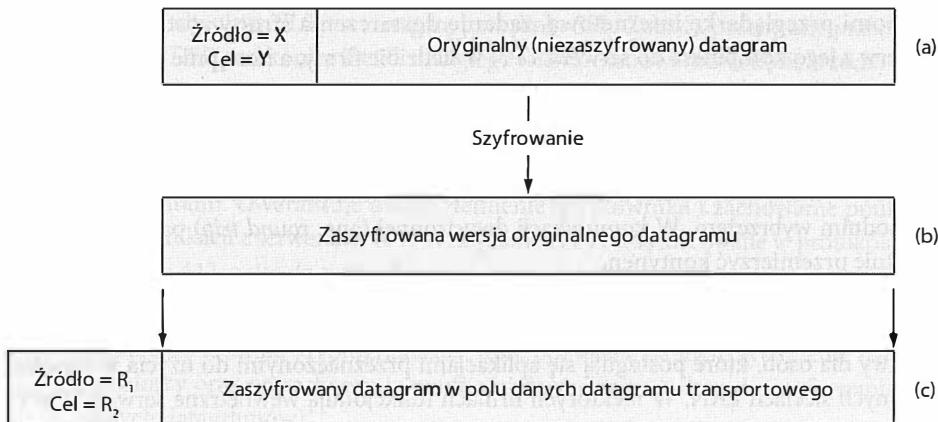
**Szyfrowanie pola danych.** W celu zachowania poufności danych metoda **szyfrowania pola danych** zakłada zaszyfrowanie informacji przekazywanych w datagramie, ale pozostawienie niezmienionych wartości nagłówkowych. Oznacza to, że osoby spoza sieci mogą poznać źródłowe i docelowe adresy pakietów, a także numery portów transmitowanych w nich segmentów. Założymy, że po jednej stronie łącza VPN znajduje się biuro prezesa firmy, a po drugiej pracuje główny księgowy. Przymijmy również, że księgowy wysyła do prezesa krótki list za każdym razem, gdy dane o stanie finansów firmy są pozytywne, oraz obszerne raporty, gdy informacje są niepokojące. Osoba spoza organizacji mogłaby w takim przypadku zaobserwować, że tuż po przesłaniu krótkich wiadomości ceny akcji firmy rosną.

**Tunelowanie IP-w-IP.** Część systemów VPN bazuje na rozwiązaniu **tunelowania IP-w-IP**, czyli na technice ukrywania nagłówka przed wysłaniem datagramu do sieci zdalnej za pośrednictwem internetu. Oprogramowanie VPN szyfruje cały wychodzący datagram (włącznie z nagłówkiem), a następnie umieszcza szyfrogram w polu danych kolejnego datagramu. W zrozumieniu zasady działania mechanizmu pomocny jest rysunek 30.7 ze strony X. Założymy, że komputer X pracujący w ośrodku 1 przygotowuje do wysłania datagram przeznaczony dla komputera Y działającego w ośrodku 2. Datagram jest przesyłany przez sieć ośrodka 1 do routera R<sub>1</sub> (tj. routera łączącego sieć ośrodka z internetem). Oprogramowanie routera R<sub>1</sub> szyfruje datagram i zapisuje go w nowym datagramie, który następnie jest przesyłany do routera R<sub>2</sub> w ośrodku 2. Po odebraniu pakietu router R<sub>2</sub> rozszyfrowuje jego zawartość i uzyskuje pierwotny datagram. Dostarcza go więc do komputera Y. Proces enkapsulacji został przedstawiony na rysunku 30.8.

Rysunek 30.8a przedstawia pierwotny datagram. Na rysunku 30.8b widać ten sam datagram po zaszyfrowaniu. Natomiast rysunek 30.8c prezentuje zewnętrzny datagram, który jest przesyłany z routera R<sub>1</sub> do routera R<sub>2</sub>. Warto zwrócić uwagę na to, że wewnętrzne adresy zostały ukryte, ponieważ wszystkie datagramy wymieniane między ośrodkami 1 i 2 są opatrzone adresami źródłowymi i docelowymi odpowiadającymi adresom routerów R<sub>1</sub> i R<sub>2</sub>.

Podsumowując:

*W przypadku zastosowania mechanizmu tunelowania IP-w-IP wszystkie pola oryginalnego datagramu są zaszyfrowane, włącznie z nagłówkiem.*



Rysunek 30.8. Przykład enkapsulacji datagramu IP w datagramie IP

**Tunelowanie IP-w-TCP.** Trzecia metoda zachowania poufności danych zakłada sformowanie tunelu TCP. Zgodnie z założeniem tego rozwiązania dwie strony ustanawiają połączenie TCP i przesyłają za jego pomocą zaszyfrowane wersje datagramów. Po stronie nadawczej procedura sprowadza się do zaszyfrowania datagramu, dołączenia niewielkiego nagłówka rozdzielającego kolejne datagramy i przesłania wynikowego zbioru danych w ramach połączenia TCP. Nagłówek składa się zazwyczaj z dwubajtowej liczby całkowitej, która określa rozmiar datagramu składowego. Po stronie odbiorczej oprogramowanie VPN odczytuje nagłówek i wskazaną liczbę kolejnych bajtów zawierających zaszyfrowaną wersję datagramu. Następnie rozszyfrowuje otrzymany blok danych i uzyskuje oryginalny datagram.

Główną przewagą techniki tunelowania IP-w-TCP nad tunelowaniem IP-w-IP jest gwarancja poprawności dostarczania danych. Protokół TCP zapewnia dostarczenie wszystkich datagramów przesyłanych między ośrodkami (niezawodnie i we właściwej kolejności). Największą jej wagą jest natomiast uzależnienie przekazu od wcześniejszych segmentów. Jak wiadomo, wszystkie datagramy muszą być dostarczane w odpowiedniej kolejności. Jeśli więc jeden segment TCP zostanie utracony lub opóźniony, mechanizm TCP nie będzie mógł dostarczyć danych z kolejnych segmentów, nawet jeśli zostały przesłane poprawnie. Jeśli potraktujemy system VPN jak mechanizm kolejkowania pakietów, to zauważymy, że przetwarzanie kolejki jest zablokowane, dopóki nie zostanie dostarczony pierwszy datagram.

O wyborze techniki tunelowania decyduje jeszcze jeden czynnik — wydajność. Oto trzy cechy, które trzeba uwzględnić:

- opóźnienie,
- przepustowość,
- narzut transmisyjny i fragmentacja.

**Opóźnienie.** Opóźnienie ma bardzo duże znaczenie, o czym może świadczyć przykład pracownika firmy zlokalizowanej na zachodnim wybrzeżu Stanów Zjednoczonych, który pracuje na wschodnim wybrzeżu (5 000 kilometrów dalej). Zadanie oprogramowania VPN sprowadza się jedynie do wysyłania datagramów do sieci firmowej. Dostarczane pakiety są wówczas przekazywane do odpowiednich miejsc docelowych. Jeśli pracownik firmy

uruchomi przeglądarkę internetową, żądanie dostarczenia strony zostanie przesłane najpierw z jego komputera do serwera VPN w siedzibie firmy, a następnie do wskazanego serwera WWW. Odpowiedź również musi najpierw zostać dostarczona do serwera VPN, skąd zostanie przekazana do komputera odbiorcy. Opóźnienia rejestrowane w dostępie do zasobów (nawet nieodległych od miejsca, w którym przebywa pracownik) są bardzo duże, co wynika, oczywiście, z konieczności przesyłania pakietów między wschodnim i zachodnim wybrzeżem. W komunikacji dwustronnej (ang. *round trip*) pakiet musi czterokrotnie przemierzyć kontynent.

**Przepustowość.** Drugie ograniczenie w stosowaniu konwencjonalnych mechanizmów VPN wynika z przepustowości dostępnej w sieci internetowej. Problem jest szczególnie dotkliwy dla osób, które posługują się aplikacjami przeznaczonymi do użycia w wysoko-wydajnych sieciach LAN. W niektórych firmach funkcjonują wewnętrzne serwisy WWW udostępniające materiały o bogatej szacie graficznej. Przesyłanie tego typu dokumentów w sieci lokalnej nie zajmuje zbyt wiele czasu. Jednak pobranie takiej strony przez użytkownika korzystającego z połączenia VPN może się wiązać z dłuższym oczekiwaniem i narażającą frustracją pracownika.

**Narzut transmisyjny i fragmentacja.** Trzeci czynnik wpływający na wydajność pracy to narzut transmisyjny wynikający z konieczności tunelowania datagramów. Aby uświadomić sobie istotę problemu, warto przeanalizować przykład sieci Ethernet, w której aplikacje formują datagramy o rozmiarze 1 500 bajtów (odpowiadające dokładnie parametrowi MTU sieci). Gdy router VPN zapisuje zaszyfrowany datagram w zewnętrznym datagramie IP, musi do niego dodać co najmniej 20 bajtów (nagłówka drugiego datagramu). Powstały w ten sposób datagram wynikowy ma rozmiar przekraczający wartość MTU i musi zostać podzielony przed przesaniem przez sieć. Odtworzenie datagramu po drugiej stronie łącza jest możliwe dopiero po odebraniu dwóch fragmentów, a to zwiększa prawdopodobieństwo utraty datagramu oraz opóźnia jego dostarczenie.

### 30.20. Rozwiązania z zakresu bezpieczeństwa sieci

Oto kilka rozwiązań, które zwiększą bezpieczeństwo komunikacji internetowej:

- **PGP.** System kryptograficzny umożliwiający aplikacjom szyfrowanie danych przed wysłaniem ich do sieci. Mechanizm PGP został opracowany w MIT i cieszy się szczególnym uznaniem naukowców.
- **SSH.** Jest to protokół warstwy aplikacji przeznaczony do zdalnego logowania się w systemie, który gwarantuje poufność wymienianych informacji (szyfruje dane przed wprowadzeniem ich do internetu).
- **SSL.** Technologia opracowana przez firmę Netscape Communications, która zapewnia uwierzytelnianie i poufność danych. Oprogramowanie SSL stanowi element pośredni między aplikacją i interfejsem gniazd. Jego działanie polega na szyfrowaniu danych przesyłanych przez internet. Mechanizm SSL jest wykorzystywany w połączeniach z serwerami WWW i umożliwia bezpieczną realizację takich zadań, jak transakcje finansowe (na przykład przesyłanie numeru karty kredytowej).

- **TLS.** Mechanizm TLS jest następcą protokołu SSL. Został opracowany przez organizację IETF pod koniec lat 90. ubiegłego wieku. Stanowi rozwinięcie trzeciej wersji mechanizmu SSL. Protokoly TLS i SSL mogą być stosowane w połączeniach HTTPS.
- **HTTPS.** W zasadzie nie jest to osobne rozwiązanie. Protokół HTTPS łączy w sobie funkcje rozwiązań HTTP oraz SSL lub TLS, a także mechanizmu zarządzania certyfikatami. Gwarantuje uwierzytelnienie użytkownika i zachowanie poufności w komunikacji z serwerami WWW. Żądania HTTPS są kierowane w protokole TCP do portu 443, a nie do portu 80.
- **IPSec.** Jest to standard zabezpieczenia datagramów IP. W rozwiązaniach IPSec wykorzystuje się techniki kryptograficzne, które pozwalają na uwierzytelnienie nadawcy i odbiorcy oraz na zachowanie poufności dostarczanych danych (szifrowanie pola danych datagramów).
- **RADIUS.** Jest to protokół przeznaczony do centralnego zarządzania uwierzytelnianiem, autoryzacją i rejestracją zdarzeń. Systemy RADIUS są szczególnie chętnie wykorzystywane przez dostawców usług internetowych, którzy obsługują połączenia modemowe. Są one również stosowane w połączeniach VPN do weryfikowania praw dostępu użytkowników sieci.
- **WEP.** Pierwotnie mechanizm ten wchodził w skład standardu bezprzewodowych sieci LAN<sup>87</sup> i miał zapewniać poufność transmisji. Naukowcy z Uniwersytetu Kalifornijskiego w Berkeley udowodnili jednak, że ma on wiele wad. Został więc zastąpiony protokołem **WPA**.

## 30.21. Podsumowanie

Lokalne sieci komputerowe i internet są wykorzystywane do działalności przestępczych. Większość zagrożeń ma związek z phishingiem, oszustwami finansowymi, blokowaniem usług, przejmowaniem kontroli nad stacjami oraz usuwaniem danych. Najczęściej stosowane techniki ataków to: podsłuchiwanie, powtarzanie pakietów, przepelnianie buforów, fałszowanie adresów i nazw, zalewanie sieci pakietami SYN, łamanie kluczy, skanowanie portów i przechwytywanie pakietów.

Każda organizacja opracowuje politykę bezpieczeństwa, w której definiuje zasady zachowania spójności danych (ochrona przed modyfikacjami), dostępności danych (zapewnienie ciągłości działania usługi) oraz zachowania poufności (ochrona przed przeglądaniem informacji i wykrywaniem jednostek sieciowych). Ponadto polityka powinna obejmować zalecenia odnośnie rejestracji zdarzeń i autoryzacji dostępu do danych (tj. zasad przekazywania odpowiedzialności za dane między pracownikami firmy).

Implementacja zapisów polityki bezpieczeństwa wymaga zastosowania różnych technik, w tym: szifrowania, obliczania wartości skrótu, generowania podpisów cyfrowych oraz certyfikatów i implementowania wirtualnych sieci prywatnych. Szifrowanie jest jednym z najważniejszych mechanizmów gwarantujących bezpieczeństwo danych.

<sup>87</sup> Współdzielała ze wszystkimi rozwiązaniami z grupy 802.11.

Systemy klucza prywatnego umożliwiają szyfrowanie i rozszyfrowywanie wiadomości z wykorzystaniem jednego tajnego klucza. Systemy klucza publicznego wymagają zastosowania dwóch kluczy — prywatnego (który nie jest przekazywany innym jednostkom) oraz publicznego (udostępnianego bez ograniczeń). Szyfrowanie jest elementem generowania podpisu cyfrowego, który uwierzytelnia nadawcę wiadomości. Z kolei autentyczność kluczy potwierdzają certyfikaty wystawiane przez organa zarządzające kluczami.

Ochrona sieci przed atakami zewnętrznymi należy do zadań zapór sieciowych, które filtryują ruch wchodzący do sieci i wychodzący z niej. Konfiguracja zapory sprowadza się do zdefiniowania zbioru reguł identyfikujących pakiety na podstawie wybranych pól nagłówka. Działanie zapór sieciowych jest uzupełniane przez systemy wykrywania włamań, które przechowują informacje o stanie połączeń i dzięki nim wykrywają zagrożenia, takie jak powtarzające się żądania SYN.

Wirtualne sieci prywatne (VPN) są rozwiązaniami, które gwarantują poufność wymiany danych przy niskim koszcie wdrożenia. Umożliwiają również pracę zdalną. Zachowanie poufności informacji jest możliwe dzięki szyfrowaniu pola danych, przekazywaniu pakietów IP w pakietach IP lub tunelowaniu pakietów IP w połączeniach TCP. Zaletą tunelowania jest to, że obejmuje szyfrowaniem zarówno pole danych, jak i nagłówek datagramu. Niestety, nie wszystkie aplikacje działają poprawnie w sieciach VPN z powodu większych opóźnień, niższej przepustowości oraz większego narzutu transmisyjnego w porównaniu z połączeniami bezpośredniimi.

Za bezpieczeństwo sieci odpowiada wiele rozwiązań, w tym: PGP, SSH, SSL, TLS, HTTPS, IPSec, RADIUS oraz WEP.

## ZADANIA

- 30.1. Wymień najważniejsze zagrożenia internetowe i krótko jest scharakteryzuj.
- 30.2. Wymień techniki ataków.
- 30.3. Załóżmy, że osoba atakująca znalazła sposób na zapisanie dowolnego odwzorowania w serwerze DNS. W jaki sposób może wykorzystać ten fakt do pozyskania informacji o rachunku bankowym użytkownika sieci?
- 30.4. W atakach DoS często wykorzystuje się segmenty TCP SYN. Czy można przeprowadzić atak DoS z użyciem segmentów TCP z danymi? Uzasadnij odpowiedź.
- 30.5. Ile maksymalnie prób trzeba wykonać, aby odgadnąć hasło składające się z ośmiu znaków, małych i dużych liter oraz cyfr?
- 30.6. Dlaczego opracowanie polityki bezpieczeństwa jest trudnym zadaniem?
- 30.7. Załóżmy, że zgodnie z polityką bezpieczeństwa firmy dostęp do informacji o płacach ma jedynie dział kadr. Jakiego rodzaju mechanizm należałoby zastosować, aby wdrożyć to zalecenie polityki? Wyjaśnij zagadnienie.
- 30.8. Wymień i opisz osiem podstawowych technik zwiększenia bezpieczeństwa sieci.
- 30.9. Czym są listy kontroli dostępu (ACL) i do czego służą?
- 30.10. Czym zajmuje się kryptografia?
- 30.11. Zapoznaj się ze standardem szyfrowania danych DES (ang. *Data Encryption Standard*). Kluczem o jakiej długości należy szyfrować dane o bardzo dużym znaczeniu?

- 30.12. Założmy, że jeden użytkownik dysponuje kluczem publicznym i kluczem prywatnym. Czy może przesyłać poufną wiadomość do drugiego użytkownika (tzn. wiadomość, którą odczyta tylko drugi użytkownik)? Uzasadnij odpowiedź.
- 30.13. Założmy, że każdy z dwóch użytkowników dysponuje kluczem publicznym i kluczem prywatnym. Co muszą zrobić, żeby mogli się codziennie komunikować, bez narażania się na atak z powtarzaniem pakietów?
- 30.14. W jaki sposób dwóch użytkowników może wykorzystać mechanizm szyfrowania z użyciem klucza publicznego do podpisania kontraktu, który zostanie następnie przesłany do trzeciej osoby?
- 30.15. Czym jest certyfikat cyfrowy?
- 30.16. Czym jest zapora sieciowa i gdzie się ją instaluje?
- 30.17. Wiele komercyjnych zapór sieciowych umożliwia administratorowi zdefiniowanie pakietów, które są **odrzucane**, oraz pakietów **akceptowanych**. Jaka jest wada konfiguracji, w której wymieniane są pakiety odrzucane?
- 30.18. Zmień konfigurację zapory sieciowej przedstawionej na rysunku 30.6 tak, aby umożliwiała użytkownikom zewnętrznym wykonywanie polecenia ping w odniesieniu do każdego z serwerów wewnętrznych.
- 30.19. Zmień konfigurację zapory sieciowej przedstawionej na rysunku 30.6, aby odzwierciedlała stan sieci po przeniesieniu usługi pocztowej do komputera z serwerem WWW.
- 30.20. Znajdź informacje na temat komercyjnych systemów IDS i sporządź listę ataków, które te systemy mogą wykrywać.
- 30.21. Założmy, że analizowany system DPI poszukuje ciągu K bajtów w każdym pakiecie. Ile porównań trzeba wykonać w najgorszym przypadku, jeśli wiadomo, że pole danych pakietu ma rozmiar 1486 bajtów?
- 30.22. Dlaczego w sieciach o dużej przepustowości nie wykorzystuje się mechanizmów szczegółowej inspekcji pakietów?
- 30.23. Jakie są dwa zadania systemu VPN?
- 30.24. Wymień trzy sposoby przesyłania danych przez internet w ramach połączeń VPN.
- 30.25. W jaki sposób nagłówek datagramu jest chroniony przed podsłuchem w sieci VPN wykorzystującej technikę tunelowania IP-w-IP?
- 30.26. W niektórych systemach VPN przed zaszyfrowaniem datagramu nadawca dodaje losową liczbę bitów o zerowej wartości. Odbiorca usuwa te bity po rozszyfrowaniu pakietu. Efektem takiego działania jest uniezależnienie rozmiaru szyfrogramu od pakietu oryginalnego. Po co więc stosuje się tę technikę?
- 30.27. Wymień osiem rozwiązań zwiększających bezpieczeństwo pracy internetowej i opisz każde z nich.
- 30.28. Poszukaj informacji na temat luk protokołu WEP. W jaki sposób problemy protokołu WEP rozwiązano w protokole WPA?

# Zawartość rozdziału

- 31.1. Wprowadzenie 557
- 31.2. Zarządzanie intranetem 557
- 31.3. Model FCAPS 558
- 31.4. Przykładowe elementy sieci 560
- 31.5. Narzędzia do zarządzania siecią 561
- 31.6. Aplikacje do zarządzania siecią 562
- 31.7. Prosty protokół zarządzania siecią 563
- 31.8. Zasada „pobierz-zapisz” w protokole SNMP 564
- 31.9. Baza MIB i nazwy obiektów 565
- 31.10. Różnorodność zmiennych MIB 565
- 31.11. Zmienne tablicowe w bazie MIB 566
- 31.12. Podsumowanie 567

# 31

## Zarządzanie siecią (SNMP)

### 31.1. Wprowadzenie

W poprzednich rozdziałach zostało opisanych wiele standardowych aplikacji internetowych. Celem tego rozdziału jest poszerzenie wspomnianej listy o narzędzia związane z zarządzaniem sieciami. Przedstawiono tutaj teoretyczny model zarządzania, który jest stosowany w praktyce i który doskonale oddaje zakres tego zagadnienia. W pierwszej części rozdziału znajduje się wyjaśnienie, dlaczego zarządzanie siecią jest tak istotne, a zarazem tak trudne. W dalszej części przedstawione zostały techniki zarządzania oraz dostępne narzędzia (w tym oprogramowanie komputerowe wykorzystywane przez administratorów do nadzorowania pracy przełączników, routerów i innych urządzeń wchodzących w skład internetu). Zaprezentowano tutaj ogólne założenia tego rodzaju systemów oraz realizowane przez nie funkcje. W końcowej części rozdziału znajduje się omówienie konkretnego protokołu zarządzania siecią oraz wyjaśnienie zasad działania oprogramowania korzystającego z tego protokołu.

### 31.2. Zarządzanie intranetem

**Administrator sieci** jest osobą odpowiedzialną za planowanie, instalowanie, uruchamianie, konfigurowanie i monitorowanie urządzeń i oprogramowania, które składają się na sieć komputerową organizacji (czyli intranet). Do zadań administratora należy również projektowanie sieci spełniających założenia wydajnościowe, nadzorowanie jej pracy, wykrywanie i naprawianie błędów uniemożliwiających lub utrudniających komunikację oraz zapobieganie ponownemu wystąpieniu tych samych problemów. Monitorowaniem objęty jest sprzęt i oprogramowanie, ponieważ zarówno urządzenia, jak i aplikacje mogą być przyczyną problemów w działaniu sieci.

Zarządzanie siecią bywa trudne. Są ku temu trzy powody. Po pierwsze, większość organizacji wykorzystuje sieci zbudowane w różnych technologiiach (urządzenia intranetu pochodzą od różnych dostawców). Po drugie, technologie ciągle się zmieniają, co oznacza konieczność okresowego instalowania nowych urządzeń i uruchamiania nowych usług. Po trzecie, większość intranetów ma duże rozmiary. Oznacza to, że część sieci jest oddalona geograficznie od pozostałych.

Dodatkowa trudność wynika z tego, że wiele mechanizmów sieciowych jest zaprojektowanych w taki sposób, aby automatycznie rozwiązywały problemy. Protokoły routingu zapewniają ominięcie uszkodzonych odcinków sieci, a chwilowa utrata pakietów pozostaje niezauważona dzięki mechanizmowi retransmisji zaimplementowanemu w protokole TCP. Niestety, automatyczne naprawianie błędów ma pewne negatywne konsekwencje. Retransmisje pakietów powodują zajęcie pasma, które mogłyby zostać wykorzystane do przesyłania nowych danych. Także niezauważone awarie sprzętowe mogą po pewnym czasie doprowadzić do poważnych problemów (gdy uszkodzeniu ulegną również trasy zapasowe).

Podsumowując:

*Mimo że urządzenia i oprogramowanie uwzględniają mechanizmy automatycznego doboru sprawnych tras lub retransmisji utraconych pakietów, administrator sieci musi monitorować pracę sieci i usuwać występujące w niej problemy.*

### 31.3. Model FCAPS

Firmy z branży sieciowej posługują się modelem FCAPS do definiowania zakresu działań w ramach systemu zarządzania siecią. Nazwa została zaczerpnięta z zalecenia M.3400 opublikowanego przez Międzynarodową Unię Telekomunikacyjną<sup>88</sup>. Kolejne litery akronimu pochodzą od angielskich słów oznaczających czynności związane z zarządzaniem. Ich znaczenie zostało opisane w tabeli 31.1.

Tabela 31.1. Model FCAPS opisujący zarządzanie siecią

Skrót	Znaczenie
F	<i>Fault</i> — wykrywanie i naprawianie usterek.
C	<i>Configuration</i> — konfiguracja i utrzymywanie sieci.
A	<i>Accounting</i> — rejestracja zdarzeń i rozliczanie.
P	<i>Performance</i> — kontrola wydajności i optymalizacja.
S	<i>Security</i> — monitorowanie zabezpieczeń i ochrona zasobów.

<sup>88</sup> Zalecenie M.3400 wchodzi w skład standardów określających zasady konfigurowania i utrzymywania sieci zarządzania systemami telekomunikacyjnymi (TMN — ang. *Telecommunications Management Network*).

**Wykrywanie i naprawianie usterek.** Wykrywanie jest najważniejszym elementem zarządzania siecią. Administrator musi monitorować działanie urządzeń sieciowych, aby wykryć problemy w ich funkcjonowaniu oraz podejmować odpowiednie działania w celu ich wyeliminowania. Najczęstszymi przyczynami awarii są: błędy w działaniu oprogramowania (na przykład zawieszenie systemu operacyjnego), uszkodzenia łączy (na przykład przypadkowe przerwanie włókna optycznego) oraz uszkodzenia sprzętowe (na przykład awaria zasilacza w routerze).

Użytkownicy sieci zgłoszają problemy, opisując ich bezpośrednie skutki. Są to stwierdzenia w rodzaju „straciłem dostęp do wspólnego dysku”. Administrator musi w takich przypadkach ustalić, czy przyczyna tkwi w oprogramowaniu, systemie zabezpieczeń (po zmianie hasła), w działaniu serwera, czy w samym połączeniu. Musi więc przeprowadzić **analizę w poszukiwaniu podstawowej przyczyny** (ang. *root-cause analysis*). Często przyczynę problemu można ustalić, łącząc ze sobą kilka faktów. Jeśli wielu użytkowników jednej sieci nagle zacznie narzekać na brak dostępu do różnych usług, można podejrzewać, że problem występuje w połączeniu, z którego korzystają wszystkie te usługi.

**Konfiguracja i utrzymanie.** Wydawałoby się, że konfiguracja jest mało istotnym elementem polityki zarządzania siecią, ponieważ wykonuje się ją tylko raz — po zdefiniowaniu wartości poszczególnych ustawień konfiguracja jest zapisywana na dysku, dzięki czemu urządzenie może ją odczytywać po każdym ponownym uruchomieniu. W praktyce jednak proces konfiguracji okazuje się bardzo złożony (z trzech powodów). Po pierwsze, w sieci pracuje wiele urządzeń i usług, a konfiguracja musi być spójna w całej sieci. Po drugie, w przypadku dodawania nowych komponentów lub zmiany założeń pracy sieci administrator musi przeanalizować ustawienia wszystkich urządzeń sieciowych, aby zyskać pewność, że wprowadzone zmiany nie zائدzą poprawnej pracy systemu. Po trzecie, większość narzędzi pozwala administratorowi na konfigurowanie pojedynczych urządzeń i protokołów. Nie ma łatwego sposobu na wprowadzenie zmian w wielu heterogenicznych komponentach.

**Rejestracja zdarzeń i rozliczanie.** W większości korporacyjnych intranetów rozliczanie jest bardzo prostą operacją. Opłaty za utrzymanie sieci są zapisane na jednym rachunku, podobnie jak opłaty za energię elektryczną czy telefon. Jednak w sieciach dostawców usługi internetowych rozliczanie może zajmować większość czasu pracy administratora. Na przykład jeśli dostawca oferuje usługi o wielu przedziałach taryfowych, w których ważne jest natężenie ruchu, system musi naliczyć opłaty oddzielnie dla każdego użytkownika. Często kontrakt z klientem zawiera zapisy uzależniające opłaty od takich parametrów, jak ilość danych wysłanych w ciągu dnia. Dostawca musi więc zapisywać szczegółowe informacje na temat generowanego ruchu, aby móc później wystawić rachunek.

**Kontrola wydajności i optymalizacja.** Administrator wykonuje dwa rodzaje zadań związanych z kontrolą wydajności — przeprowadza **testy diagnostyczne**, które umożliwiają wykrycie problemów i niedoskonałości, oraz **analizuje trendy**, co pozwala mu przewidzieć konieczność zwiększenia pojemności systemu. Testy diagnostyczne mają na celu sprawdzenie, czy jest możliwe zwiększenie stopnia wykorzystania istniejącej sieci. Na przykład jeśli administrator znajdzie łącze o małym obciążeniu, może podjąć decyzję o skierowaniu do niego większego ruchu. Analiza trendów pozwala z kolei na ustalenie, co trzeba zrobić, aby sieć była gotowa na większe obciążenie w przyszłości. Wielu admi-

nistratorów obserwuje stopień wykorzystania łącza między siecią firmową a internetem i planuje zwiększenie jego pojemności, gdy obciążenie przekroczy 50%.

**Monitorowanie zabezpieczeń i ochrona zasobów.** Zagwarantowanie odpowiedniego poziomu bezpieczeństwa jest jednym z najtrudniejszych elementów zarządzania siecią. Dotyczy bowiem wielu warstw stosu protokołów oraz wielu urządzeń. W szacowaniu jakości zabezpieczeń obowiązuje zasada najsłabszego ogniwa. Jeśli można złamać zabezpieczenia choćby jednego urządzenia, cała sieć jest zagrożona. Poza tym fakt opracowywania coraz nowszych form ataków sprawia, że sieć bezpieczna w danej chwili wkrótce może się okazać podatna na włamania (chyba że administrator wprowadzi odpowiednie zabezpieczenia).

### 31.4. Przykładowe elementy sieci

W specyfikacjach systemów zarządzania sieciami stosuje się pojęcie **element sieci** (ang. *network element*), które odnosi się do dowolnego urządzenia sieciowego, systemu lub mechanizmu, którego pracę można zarządzać. Choć wiele sieci składa się jedynie z fizycznych urządzeń, definicja ta obejmuje również usługi (takie jak DNS). Kilka przykładowych elementów sieci zostało wymienionych w tabeli 31.2.

Tabela 31.2. Przykładowe elementy sieci, które podlegają zarządzaniu

Zarządzalne elementy sieci	
Przełącznik warstwy 2.	Router IP
Przełącznik VLAN	Zapora sieciowa
Punkt dostępu bezprzewodowego	Moduł CSU/DSU
Centralowy modem DSL	Koncentrator DSLAM
Serwer DHCP	Serwer DNS
Serwer WWW	Moduł rozkładania obciążenia

W literaturze branżowej często wykorzystuje się termin **zarządzania elementem** w odniesieniu do działań związanych z konfiguracją i podtrzymaniem pracy pojedynczego elementu sieci. Niestety, większość dostępnych narzędzi zawiera funkcję zarządzania tylko pojedynczymi elementami. Oznacza to, że gdy konieczne jest skonfigurowanie określonej usługi między dwoma punktami końcowymi, administrator musi definiować ustawienia każdego elementu na trasie między tymi punktami. Na przykład aby utworzyć tunel MPLS przebiegający przez kilka routerów, administrator musi skonfigurować każdy z routerów oddzielnie. Podobnie, w przypadku wdrożenia w sieci jednolitej polityki routingu konieczne jest konfigurowanie każdego elementu osobno.

Oczywiście, w takich przypadkach zawsze istnieje ryzyko pomyłki, co oznacza, że zarządzanie elementami jest narażone na błędy konfiguracyjne. Co ważniejsze, znalezienie błędu wymaga od administratora sprawdzania kolejno pojedynczych systemów.

*System zarządzania elementami jest pracochnonny i narażony na błędy, ponieważ umożliwia konfigurowanie, monitorowanie i sterowanie tylko jednego elementu sieci naraz.*

## 31.5. Narzędzia do zarządzania siecią

Narzędzia do zarządzania siecią są zaliczane do dwunastu kategorii określających ich przeznaczenie. Oto lista tych kategorii:

- testy warstwy fizycznej,
- dostępność urządzeń i poprawność połączeń,
- analiza pakietów,
- wykrywanie sieci,
- odpytywanie urządzeń,
- monitorowanie zdarzeń,
- monitorowanie wydajności,
- analiza ruchu,
- routing i inżynieria ruchu,
- konfiguracja,
- wdrażanie zabezpieczeń,
- projektowanie sieci.

Testowanie warstwy fizycznej sprowadza się do wykrywania nośnej (zadanie to wykonuje moduł zainstalowany w każdej karcie sieciowej) i pomiaru siły sygnału w łączności radiowej. Polecenie ping jest doskonałym narzędziem sprawdzającym dostępność urządzeń sieciowych, które jest często wykorzystywane przez administratorów. Do przechwytywania i wyświetlania pakietów, a także do sporządzania statystyk służą narzędzia nazywane **monitorami pakietów** lub **analizatorami protokołów**. Jednym z takich narzędzi jest darmowy program **Wireshark**.

Narzędzia przeznaczone do wykrywania sieci sporządzają schematy sieci, wykorzystując do tego celu technikę sondowania obecności urządzeń. Administratorzy często posługują się przygotowanymi w ten sposób schematami do odnajdywania określonych urządzeń, a następnie używają narzędzi do odpytywania tych urządzeń. Programy odpowiadające za monitorowanie zdarzeń generują różnego rodzaju alerty. Administrator zazwyczaj konfiguruje urządzenia sieciowe w taki sposób, aby wysyłały powiadomienia o przekroczeniu określonych wartości progowych (na przykład gdy stopień wykorzystania łącza osiągnie wartość 80%). Narzędzia monitorujące zajmują się wówczas wyświetlaniem tego typu ostrzeżeń. Programy monitorowania wydajności przygotowują wykresy wydajności sieci w czasie, dzięki czemu osoby zarządzające pracą sieci mogą zaobserwować pewne trendy w jej działaniu.

W określaniu trendów pomocne są również analizatory ruchu, takie jak NetFlow. Dostarczane przez nie informacje nie ograniczają się jedynie do wartości całkowitego natężenia ruchu, ale są podzielone na kategorie odzwierciedlające zmiany w komunikacji o różnym charakterze (przekazują na przykład dane o zwiększeniu ruchu VoIP).

Narzędzia odpowiedzialne za konfigurację, routing i inżynierię ruchu są ze sobą powiązane. Każde z nich ułatwia zarządzanie komponentami sieci. Programy związane z routingu odpowiadają za parametryzowanie protokołów routingu dynamicznego i monitorowanie ich pracy. Nadzorują również stan tablic routingu. Aplikacje z zakresu inżynierii ruchu są przeznaczone do konfigurowania i monitorowania tuneli MPLS oraz parametrów QoS. Narzędzia konfiguracyjne ogólnego przeznaczenia umożliwiają administratorowi instalowanie i modyfikowanie konfiguracji w różnych komponentach sieciowych. Niektóre z nich ułatwiają na przykład wykonywanie tych samych zadań w odniesieniu do wielu (zazwyczaj identycznych) elementów sieci. Na przykład jeśli jedna z reguł zapory sieciowej zostanie zmieniona, a w sieci funkcjonuje wiele takich zapór, zautomatyzowane narzędzie konfiguracyjne (często skrypt języka Perl) może wprowadzić tę samą regułę w innych jednostkach.

Na rynku oprogramowania dostępnych jest wiele narzędzi do zarządzania pracą jednostek odpowiedzialnych za bezpieczeństwo sieci. Niektóre z tych narzędzi umożliwiają zdefiniowanie polityki bezpieczeństwa, a następnie próbują skonfigurować urządzenia tak, aby tę politykę wdrożyć, lub testują komponenty sieciowe, sprawdzając, czy dana polityka jest stosowana. Inne aplikacje służą do sprawdzania poziomu zabezpieczeń sieci — prowadzą ataki na urządzenia i usługi, a później generują raport o skuteczności tych działań.

Projektowanie sieci jest bardzo złożoną operacją, a wspomagające ją narzędzia należą do najbardziej wyrafinowanych. Na przykład implementacje algorytmów programowania liniowego ułatwiają administratorom optymalizację architektury sieci i zaplanowanie mechanizmów zarządzania ruchem. Dostępne są również programy, które wyszukują słabe punkty w sieci (na przykład miejsca, w których wystąpienie dwóch lub większej liczby błędów może spowodować odłączenie użytkowników od sieci).

*Administratorzy mają do dyspozycji wiele różnorodnych narzędzi, które ułatwiają im konfigurowanie, testowanie i analizowanie sieci.*

### 31.6. Aplikacje do zarządzania siecią

Większość narzędzi opisanych w poprzednim punkcie obejmuje swoim zasięgiem całą sieć — administrator, pracując w swoim biurze, może dzięki odpowiednim rozwiązaniom sieciowym komunikować się z dowolnie wybranym elementem sieci. Co ciekawe, zarządzanie siecią nie zostało zdefiniowane jako część składowa protokołów niższych warstw. Protokoły wykorzystywane do monitorowania i konfigurowania urządzeń działają w warstwie aplikacji. Interakcja z określonym urządzeniem polega na tym, że administrator uruchamia aplikację pełniąącej funkcję klienta, która komunikuje się z urządzeniem

działającym tak jak serwer. Klient i serwer używają standardowych protokołów transportowych (UDP i TCP). Zarządzanie nie wymaga też budowania osobnej sieci. W większości przypadków ruch związany z zarządzaniem jest przekazywany w samej sieci użytkowej.

Aby odróżnić aplikacje uruchamiane przez użytkowników komputerów od programów zarezerwowanych dla administratorów sieci, w systemach zarządzania unika się używania określeń **klient** i **serwer**. Program kliencki uruchomiony w komputerze administratora jest nazywany **menedżerem**. Natomiast serwer działający w urządzeniu sieciowym nazywa się **agentem**<sup>89</sup>.

Wykorzystanie klasycznych protokołów transportowych do przenoszenia ruchu związanego z zarządzaniem może się wydawać nieefektywne, ponieważ usterki oprogramowania protokołów lub urządzeń mogą zakłócić przekazywanie pakietów do urządzenia (lub od urządzenia), a tym samym uniemożliwić zmianę jego sposobu działania. Niektórzy operatorzy instalują oddzielne urządzenia przeznaczone do obsługi ruchu związanego z zarządzaniem wyjątkowo ważnymi komponentami sieciowymi (na przykład modemy telefoniczne, które przyłączone do routerów rdzeniowych zapewniają administratorowi kontakt z urządzeniem nawet w sytuacji, gdy sieć nie przenosi danych). W praktyce jednak takie rozwiązania nie są często wdrażane. Wykorzystanie protokołów warstwy aplikacji sprawdza się z dwóch powodów. Po pierwsze, w przypadku uszkodzenia sprzętu, skutkującego przerwaniem komunikacji, administrator może połączyć się z urządzeniami, które nadal działają, i za pomocą kilku prób i błędów zlokalizować miejsce problemu. Po drugie, zastosowanie standardowych protokołów transportowych powoduje, że pakiety generowane przez administratora są przesyłane w takich samych warunkach, jak ruch użytkowy. Jeśli więc wzrosną opóźnienia, administrator natychmiast ten fakt zauważa.

## 31.7. Prosty protokół zarządzania siecią

**Prosty protokół zarządzania siecią** (SNMP — ang. *Simple Network Management Protocol*) jest podstawowym systemem zarządzania siecią. W czasie pisania książki obowiązywała 3. wersja standardu, zapisywana jako SNMPv3. Protokół SNMP wyznacza zasady wymiany danych między menedżerem i agentem. W specyfikacji SNMP zapisano między innymi format żądań przesyłanych z menedżera do agenta oraz format odpowiedzi dostarczanych w przeciwnym kierunku. Zdefiniowano w niej także znaczenie każdego żądania i każdej odpowiedzi. Zgodnie z zaleceniem komunikaty SNMP są zapisywane w standardzie o nazwie **abstrakcyjna notacja składniowa 1** (ASN.1 — ang. *Abstract Syntax Notation*).

Choć szczegółowe omówienie notacji ASN.1 wykracza poza ramy tematyczne książki, analiza jednego przykładu powinna pomóc w zrozumieniu zasady kodowania komunikatów. Założmy, że między agentem a menedżerem trzeba przesyłać liczbę całkowitą. Aby uniknąć marnowania pamięci i pasma podczas przesyłania danych, umożliwiając jednocześnie przekazywanie dużych wartości liczbowych, zastosowano taką technikę kodowania, w której każdy obiekt jest opisywany za pomocą dwóch parametrów — wartości

<sup>89</sup> Choć w dalszej części rozdziału stosowane są terminy **menedżer** i **agent**, należy pamiętać, że odnoszą się one do programów pełniących odpowiednio funkcje klienta i serwera.

oraz rozmiaru pola. Na przykład do przesyłania liczby całkowitej z przedziału od 0 do 255 wystarczy jeden oktet. Natomiast wartości z przedziału od 256 do 65535 są reprezentowane za pomocą dwóch oktetów. Kolejne wartości wymagają użycia trzech lub większej liczby oktetów. Zatem przesyłanie liczby całkowitej zgodnie z notacją ASN.1 oznacza obowiązek wygenerowania dwóch wartości — rozmiaru  $R$ , po którym następuje  $R$  bajtów reprezentujących liczbę. Specyfikacja ASN.1 umożliwia kodowanie dowolnie dużych wartości liczbowych dzięki przeznaczeniu na pole długości więcej niż jednego oktetu. Jednak tak duże liczby całkowite nie są wykorzystywane w protokole SNMP. Zasada kodowania liczb została przedstawiona w tabeli 31.3.

Tabela 31.3. Przykłady zapisu liczb całkowitych zgodnie z notacją ASN.1

Wartość dziesiętna	Szesnastkowy odpowiednik wartości dziesiętnej	Bajt długości	Bajty wartości (w formacie szesnastkowym)
27	1B	01	1B
792	318	02	03 18
24 567	5FF7	02	5F F7
190 345	2E789	03	02 E7 89

### 31.8. Zasada „pobierz-zapisz” w protokole SNMP

Standard SNMP nie opisuje dużego zbioru polecień. W zasadzie całe jego działanie opiera się na **zasadzie „pobierz-zapisz”**, która definiuje dwie podstawowe operacje — **pobierania** określonych wartości z urządzenia lub **zapisywania** wartości w urządzeniu. Każdy pobierany lub zapisywany obiekt ma niepowtarzalną nazwę. Polecenie odpowiedzialne za pobranie lub zapisanie obiektu musi tę nazwę określić.

Wykorzystanie operacji pobierania do monitorowania pracy urządzenia lub pozyskiwania informacji statusowych wydaje się oczywiste. Wymaga jedynie uprzedniego zdefiniowania obiektów i nadania im odpowiednich nazw. Aby uzyskać informacje o stanie urządzenia, administrator pobiera wartość skojarzoną z określonym obiektem. Na przykład można zdefiniować obiekt, który będzie zliczał ramki odrzucone przez urządzenie z powodu błędnej wartości sumy kontrolnej. Urządzenie musi jednak zostać zaprojektowane w taki sposób, aby zwiększało wartość licznika po każdorazowym odebraniu ramki z niewłaściwą sumą kontrolną. Uzyskanie informacji o liczbie niepoprawnych ramek sprowadza się wówczas do użycia protokołu SNMP, który pobierze wartość licznika.

Użycie mechanizmu „pobierz-zapisz” do sterowania pracą urządzenia nie jest już takie oczywiste. Operacje sterowania są bowiem efektem ubocznym zapisu wartości w obiekcie. Protokół SNMP nie ma oddzielnych polecień wyzerowania licznika błędów lub ponownego uruchomienia urządzenia. Gdyby administrator chciał wyzerować licznik, to mógłby, oczywiście, zapisać zerową wartość w obiekcie skojarzonym z licznikiem. Jednak wykonanie operacji takiej jak ponowne uruchomienie urządzenia jest możliwe tylko wtedy, gdy oprogramowanie agenta SNMP interpretuje **zapis** odpowiedniej wartości jako rozkaz

realizacji określonej sekwencji czynności. Oprogramowanie SNMP mogłoby więc zawierać definicję obiektu odpowiadającego za ponowne uruchomienie i działać w taki sposób, że zapisanie w obiekcie zera powoduje restart urządzenia. Oczywiście, obiekty SNMP są komponentami wirtualnymi. Nie są implementowane bezpośrednio w samym mechanizmie wykonawczym urządzenia. Możliwość monitorowania stanu urządzenia i sterowania jego pracą wynika z działalności agenta, który odbiera żądania i realizuje zadania skojarzone z operacjami **pobierania i zapisu** określonych obiektów.

*Protokół SNMP wykorzystuje zasadę „pobierz-zapisz” do komunikacji między menedżerem i agentem. Menedżer pobiera wartości w celu sprawdzenia stanu urządzenia. Natomiast sterowanie urządzeniem jest efektem ubocznym zapisywania w nim wartości obiektów.*

## 31.9. Baza MIB i nazwy obiektów

Każdy obiekt, do którego protokół SNMP ma dostęp, musi zostać wcześniej zdefiniowany i opatrzony niepowtarzalną nazwą. Programy menedżera i agenta muszą posługiwać się tymi samymi nazwami i w jednakowy sposób interpretować operacje pobierania i zapisu wartości. Wszystkie obiekty dostępne z poziomu protokołu SNMP są więc definiowane w **bazie danych informacji zarządzania** (MIB — ang. *Management Information Base*).

W praktyce definicja bazy MIB nie jest bezpośrednio powiązana z protokołem SNMP. Standard SNMP opisuje jedynie format komunikatów i sposób ich kodowania. Specyfikacja zmiennych MIB oraz znaczenie operacji odczytu i zapisu wartości są zawarte w oddzielnym standardzie. Zmienne te są również zdefiniowane w standardach dotyczących poszczególnych typów urządzeń.

Nazewnictwo obiektów zdefiniowanych w bazie MIB odpowiada założeniom notacji ASN.1, zgodnie z którą każdemu obiektowi jest przypisywany długis prefiks gwarantujący niepowtarzalność nazwy. Na przykład licznik datagramów IP dostarczanych do urządzenia ma nazwę:

iso.org.dod.internet.mgmt.mib.ip.ipInReceives

Jednak gdy nazwy obiektów są wykorzystywane w komunikatach SNMP, każdej ich części przypisywana jest pewna liczba całkowita. Zatem nazwa obiektu *ipInReceives* w komunikacie SNMP ma postać:

1.3..6.1.2.1.4.3

## 31.10. Różnorodność zmiennych MIB

Dzięki temu, że specyfikacja SNMP nie określa zbioru zmiennych MIB, mechanizm ten charakteryzuje się dużą elastycznością. W razie potrzeby baza MIB może zostać uzupełniona o kolejne zmienne bez konieczności wprowadzania zmian w samym protokole SNMP.

Co więcej, oddzielenie protokołu komunikacyjnego od definicji obiektów pozwala każdemu na powoływanie własnych zmiennych. Na przykład autorzy nowego protokołu mogą zdefiniować zmienne MIB, które będą wykorzystywane do monitorowania oprogramowania protokołu oraz do sterowania nim. Analogicznie, twórcy nowego urządzenia mogą zdefiniować zmienne MIB, które posłużą do nadzorowania pracy tego urządzenia.

Zgodnie z przewidywaniami autorów standardu baza MIB składa się obecnie z wielu zbiorów zmiennych. Odpowiadają one protokołom takim jak UDP, TCP, IP, ARP, a także komponentom sprzętowym (na przykład elementom sieci Ethernet, routerom, przełącznikom, modemom i drukarkom)<sup>90</sup>.

### 31.11. Zmienne tablicowe w bazie MIB

Poza prostymi zmiennymi (takimi jak liczby całkowite reprezentujące liczniki) w bazie danych MIB definiowane są także zmienne tablicowe. Okazują się one bardzo użyteczne, ponieważ odpowiadają analogicznym strukturam danych w systemie komputera. Doskonałym przykładem jest tutaj tablica routingu. W większości praktycznych implementacji jest ona przedstawiana jako zbiór wpisów, z których każdy zawiera adres sieci docelowej oraz informacje o kolejnym węźle na trasie do danej sieci.

W przeciwieństwie do klasycznych języków programowania notacja ASN.1 nie uwzględnia odwołań z użyciem indeksu. Operowanie indeksem ma charakter niejawny. Nadawca musi wiedzieć, że obiekt jest tablicą, i musi dodać wartość indeksu do nazwy obiektu. Na przykład zmienna:

```
standardowy prefix MIB.ip.ipRoutingTable
```

odpowiada tablicy routingu IP, w której każdy wpis ma kilka pól. Zgodnie z przeznaczeniem tablicy routingu indeksem w tablicy jest adres IP sieci docelowej. Chcąc pobrać wartość określonego pola w wierszu, administrator może określić nazwę obiektu w następujący sposób:

```
standardowy prefix  
→MIB.ip.ipRoutingTable.ipRouteEntry.pole.adresSieci
```

Element *pole* odpowiada tutaj jednemu z pól występujących we wpisie, a element *adresSieci* reprezentuje czterobajtowy adres IP, który w tym przypadku został wykorzystany jako indeks w tablicy. Wiedząc, że nazwa *ipRouteNextHop* odpowiada adresowi następnego węzła na trasie, możemy pobrać informacje na temat tego węzła za pomocą następującego żądania (nazwy zostały zamienione na identyfikatory liczbowe):

```
1.3.6.1.2.1.4.21.1.7.adresSieci
```

Ciąg 1.3.6.1.2.1 odpowiada w tym przypadku standardowemu prefiksowi MIB, 4 jest kodem zmiennej *ip*, 21 reprezentuje zmienną *ipRouterEntry*, 7 odpowiada nazwie *ipRouteNextHop*, a *adresSieci* odpowiada liczbowej wartości adresu IP sieci docelowej.

---

<sup>90</sup> Poza zestawem standardowych zmiennych MIB, które są obsługiwane przez urządzenia różnych producentów, każdy dostawca może zdefiniować własne wartości, obowiązujące w jego oprogramowaniu lub urządzeniach.

*Mimo że notacja ASN.1 nie uwzględnia odwołań z użyciem indeksu, zmienne MIB mogą się odnosić do tablic. Odwołanie do tablicy z użyciem zmiennej ASN.1 wymaga dodania indeksu pozycji do nazwy zmiennej. Gdy oprogramowanie agenta napotka nazwę odpowiadającą tablicy, wykorzystuje indeks do wskazania określonej wartości w tablicy.*

## 31.12. Podsumowanie

Administrator sieci jest osobą, która monitoruje urządzenia i oprogramowanie wchodzące w skład intranetu oraz steruje ich pracą. Zgodnie z modelem FCAPS pięć podstawowych zadań związanych z zarządzaniem siecią to wykrywanie usterek, konfiguracja, rozliczanie, monitorowanie wydajności i zabezpieczenie systemu. W zarządzaniu siecią pomaga administratorowi wiele narzędzi. Większość z nich odnosi się jednak tylko do pojedynczych urządzeń. Zadania obejmujące wiele komponentów wymagają od administratora większego zaangażowania.

Oprogramowanie wspomagające zarządzanie siecią bazuje na modelu klient-serwer. Składa się więc z dwóch komponentów. Moduł uruchomiony po stronie komputera zarządzającego nazywa się **menedżerem** (i pełni funkcję klienta). Natomiast komponent zaimplementowany w urządzeniu sieciowym jest nazywany **agentem** (i pełni rolę serwera).

Zarządzanie urządzeniami internetowymi należy do zadań **prostego protokołu zarządzania siecią** (SNMP). Standard SNMP definiuje format i znaczenie komunikatów, które menedżer i agent wymieniają między sobą. W specyfikacji SNMP określono jedynie operacje pobierania i zapisywania wartości zmiennych, sterowanie urządzeniem jest efektem ubocznego zapisu wartości.

W standardzie SNMP nie określono zmiennych, które są obsługiwane przez protokół. Wykaz zmiennych oraz ich znaczenie są zawarte w oddzielnych specyfikacjach. Dzięki temu producenci sprzętu i oprogramowania mogą definiować własne zbiorzy zmiennych MIB. Nazwy zmiennych MIB muszą być zgodne z notacją ASN.1. Każdej zmiennej odpowiada długi ciąg nazw odzwierciedlających hierarchię parametrów, który na czas transmisji jest przekształcany do postaci liczbowej. W standardzie ASN.1 nie uwzględniono możliwości operowania tablicami. Jednak odwołania do tablic są emulowane przez dodanie indeksu na końcu zmiennej.

## ZADANIA

- 31.1. Podaj przykłady mechanizmów, które ukrywają błędy występujące w protokołach.
- 31.2. Których elementów modelu FCAPS dotyczy skarga użytkownika, jeśli informuje on o braku dostępu do określonej usługi?
- 31.3. Które elementy modelu FCAPS odnoszą się do przypadku błędnego funkcjonowania zapory sieciowej? Uzasadnij odpowiedź.

- 31.4. Podaj dwa przykłady elementów zarządzalnych, które nie zostały wymienione w tabeli 31.2.
- 31.5. Czym jest monitor pakietów?
- 31.6. O jakich problemach informuje analizator ruchu?
- 31.7. Jakich terminów używa się (w oprogramowaniu do zarządzania siecią) zamiast określeń **klient i serwer**?
- 31.8. W notacji ASN.1 zdefiniowany jest dokładny format liczby całkowitej. Dlaczego nie wystarczy stwierdzenie, że liczba całkowita składa się z 32 bitów?
- 31.9. Wiadomo, że nie zaleca się korzystania z sieci do rozwiązywania problemów tej sieci. Dlaczego więc systemy SNMP wymieniają dane w sieci, którą zarządzają?
- 31.10. Napisz program, który odczyta liczbę o dowolnie dużej wartości zapisaną w formacie dziesiętnym i wyświetli ją na ekranie w formacie przedstawionym w tabeli 31.3.
- 31.11. Jakie dwie operacje opisuje standard SNMP?
- 31.12. Pobierz darmowe oprogramowanie SNMP i spróbuj wykorzystać je do komunikacji z dowolnym urządzeniem sieciowym (na przykład z drukarką).
- 31.13. Czy standard SNMP zawiera wykaz nazw wszystkich zmiennych występujących w bazie MIB? Wyjaśnij zagadnienie.
- 31.14. Jaka jest przewaga dodawania indeksu do nazwy nad wykorzystaniem klasycznych tablic indeksowanych liczbami całkowitymi?
- 31.15. Poszukaj informacji na temat zapisu nazw i wartości w standardzie ASN.1, a następnie napisz program komputerowy, których będzie kodował i dekodował nazwy (takie jak *ipInReceives*).



# Zawartość rozdziału

- 32.1. Wprowadzenie 571
- 32.2. Zapotrzebowanie na skalowalne usługi internetowe 571
- 32.3. Buforowanie treści (Akamai) 572
- 32.4. Rozkładanie obciążenia serwerów WWW 572
- 32.5. Wirtualizacja serwerów 573
- 32.6. Komunikacja P2P 573
- 32.7. Rozproszone centra danych i replikacja 574
- 32.8. Jednolita reprezentacja danych (XML) 574
- 32.9. Sieci społecznościowe 575
- 32.10. Mobilność i sieci bezprzewodowe 575
- 32.11. Cyfrowy przekaz wideo 575
- 32.12. Multiemisja 576
- 32.13. Dostęp szerokopasmowy i przełączanie 576
- 32.14. Przełączanie optyczne 577
- 32.15. Sieć w biznesie 577
- 32.16. Czujniki w domu i otoczeniu 577
- 32.17. Sieci ad hoc 578
- 32.18. Procesory wielordzeniowe i sieciowe 578
- 32.19. IPv6 578
- 32.20. Podsumowanie 579

## *Trendy w technologiach sieciowych i sposobach wykorzystywania sieci*

### **32.1. Wprowadzenie**

Jedną z najbardziej ekscytujących rzeczy związanych z internetem jest ciągłe pojawianie się nowych aplikacji i technologii sieciowych. Znaczna większość aplikacji wymieniających pakiety w internecie została opracowana w poprzedniej dekadzie. Ich działanie nie było możliwe w początkowych fazach rozwoju sieci, ponieważ nie istniały wówczas odpowiednie technologie. Nie było również stosownej infrastruktury sieciowej.

W tym rozdziale omówionych zostało kilka trendów w rozwoju technologii sieciowych, aplikacji i usług. Uwzględniono tutaj zarówno najnowsze prace wdrożeniowe, jak i badania trwające od dłuższego czasu.

### **32.2. Zapotrzebowanie na skalowalne usługi internetowe**

W jednostkowym ujęciu model komunikacji klient-serwer zakłada, że jedna aplikacja (serwer) jest uruchamiana jako pierwsza i oczekuje na kontakt ze strony drugiej aplikacji (klienta). W szerszej perspektywie określenie **klient-serwer** charakteryzuje architekturę, w której wiele jednostek klienckich odwołuje się do jednego centralnego serwera. Firma utrzymująca witrynę WWW spodziewa się połączeń od nieznanych jej użytkowników. Wadą skoncentrowanych usług jest ich wydajność. Wraz ze wzrostem liczby klientów serwer (lub fragment sieci odpowiadający za przekazywanie danych z serwera) staje się „wąskim gardłem”, szczególnie jeśli aplikacje klienckie pobierają duże ilości danych.

Przeciążenie serwerów wydaje się największym ograniczeniem większości usług internetowych. W związku z tym zarówno jednostki badawcze, jak i producenci komponentów sieciowych pracują nad rozwiązaniami organizacyjnymi i technologiami, które umożliwiają skalowanie usługi i wdrażanie nowych pomysłów. Kilka z tych prac zostało opisanych w kolejnych podrozdziałach.

*Opracowano wiele technologii umożliwiających skalowanie usług internetowych. Choć poszczególne rozwiązania bardzo się od siebie różnią, każde z nich znajduje zastosowanie w pewnych konfiguracjach.*

### 32.3. Buforowanie treści (Akamai)

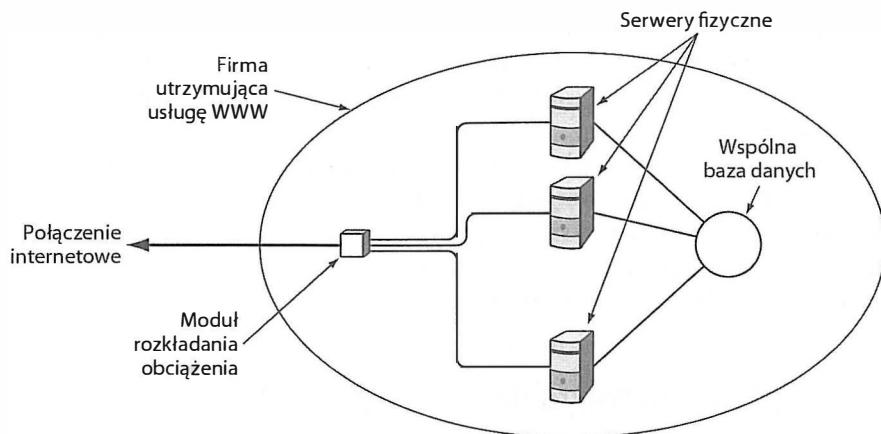
Jednym z podstawowych sposobów zwiększenia dostępności serwisów jest buforowanie treści WWW. Dostawcy usług internetowych często przechowują kopie statycznych stron internetowych (stron, których zawartość nie zmienia się zbyt często). Dzięki temu, jeśli  $N$  użytkowników sieci będzie chciało pobrać tę samą stronę, tylko jedno żądanie zostanie wysłane do serwera. Pozostałe  $N-1$  żądania zostaną obsłużone na podstawie danych zapisanych w pamięci podręcznej.

Firmy, takie jak Akamai, rozszerzyły ideę buforowania danych o usługę rozproszonego buforowania treści. Firma Akamai utrzymuje wiele serwerów rozmieszczone w różnych obszarach internetu i umożliwia innym organizacjom (po podpisaniu umowy) zapisywanie w nich wstępnie przygotowanych dokumentów. Dbanie o aktualność informacji należy do zadań klienta, który może okresowo aktualizować treści zapisane w serwerach buforujących. Rozwiązanie to gwarantuje, że użytkownicy serwerów internetowych otrzymają większość danych z pobliskiego serwera Akamai, a nie z centralnego serwera organizacji. Tym samym obciążenie serwerów klienckich zostaje znacznie zmniejszone.

### 32.4. Rozkładanie obciążenia serwerów WWW

W związku z intensywnym wykorzystywaniem sieci i wzrostem zainteresowania sprzedażą wyrobów za pośrednictwem serwisów WWW coraz większą wagę przykłada się do problemu optymalizacji pracy serwerów WWW. Jednym z ciekawszych rozwiązań stosowanych w projektowaniu dużych serwisów internetowych jest wykorzystywanie mechanizmów **rozkładania obciążenia**. Umożliwiają one uruchomienie usługi na kilku komputerach i dystrybuowanie nadchodzących żądań pomiędzy kilkoma fizycznymi jednostkami. Zasadę działania tego mechanizmu ilustruje rysunek 32.1.

Moduł rozkładania obciążenia analizuje każde nadchodzące żądanie HTTP i kieruje je do jednego z serwerów. Jednocześnie zapamiętuje źródło pakietów, dzięki czemu kolejne żądania z tej danej jednostki zawsze trafiają do tego samego serwera. Aby zagwarantować



Rysunek 32.1. Rozkładanie obciążenia w serwisach obsługujących dużą liczbę żądań

zwracanie jednakowych odpowiedzi na żądania, wszystkie serwery muszą korzystać ze wspólnej bazy danych. Jeśli więc jeden z użytkowników systemu złoży jakieś zamówienie, wszystkie serwery WWW będą miały do niego dostęp.

## 32.5. Wirtualizacja serwerów

Inny sposób zwiększania skalowalności wiąże się z **wirtualizacją serwerów**. Zainteresowanie firm tym rozwiązaniem wynika z obserwacji, że w większości sieci utrzymywanych jest wiele serwerów (na przykład serwery pocztowe, serwery WWW, serwery baz danych). W tradycyjnej architekturze sieci każdy serwer jest uruchamiany w osobnym komputerze. Jeśli serwery działające w systemie komputera A są ciągle zajęte, a serwery komputera B pozostają nieaktywne, pojawia się problem z wydajnością całego systemu.

Rozwiązaniem problemu jest wirtualizacja. Umożliwia ona administratorowi przekształcanie oprogramowania serwerowego z jednego komputera do drugiego w dowolnym czasie. Oczywiście, wymaga to uwzględnienia wielu parametrów technicznych, w tym zmiany routingu. Jednak sama idea nie wydaje się skomplikowana. Wystarczy uruchomić serwer w systemie **maszyny wirtualnej** (VM — ang. *Virtual Machine*), która obsługuje funkcję migracji systemu. Jeśli jeden z komputerów fizycznych stanie się niedostatecznie wydajny, administrator może przenieść określony proces (lub większą liczbę procesów) do innego komputera.

## 32.6. Komunikacja P2P

W latach 90. ubiegłego wieku kilka grup badawczych testowało nową technikę zwiększenia efektywności pobierania plików. Zgodnie z jej założeniami zamiast kopiować cały plik z serwera, klient może pobierać poszczególne jego fragmenty. Fragmenty te są zapisywane na serwerze również z wykorzystaniem sieci. Jeśli klientowi jest potrzebny określony fragment pliku, może go znaleźć na jednym z pobliskich serwerów. Aby zwiększyć

liczbę jednostek udostępniających dane, każdy użytkownik pobierający fragment pliku musi wyrazić zgodę na wykorzystanie jego komputera jako serwera dostępnego dla innych użytkowników internetu. Powstający w ten sposób system jest nazywany **systemem peer-to-peer** (P2P).

Największą popularność zyskały systemy P2P służące do wymiany plików muzycznych. Zarówno Napster, jak i Kazaa były usługami P2P, z których bardzo chętnie korzystały nastolatki. Oczywiście, większość użytkowników nie interesuje się zasadami działania określonego mechanizmu i nie wie, że korzystając z systemu P2P, zgadzają się na udostępnianie plików innym użytkownikom.

### 32.7. Rozproszone centra danych i replikacja

Mimo dostępności buforów treści, mechanizmów rozkładania obciążenia, systemów virtualizacji serwerów oraz komunikacji P2P niektóre serwisy generują ruch o tak dużym natężeniu, że potrzebne jest inne rozwiązanie — replikacja całego serwisu, zapewniana przez **rozproszone centra danych**.

Przykładem takiego rozwiązania jest system wyszukiwania Google. Usługi Google realizują miliardy połączeń dziennie. Aby obsłużyć tak wiele zapytań, firma Google utworzyła kilka centrów danych w różnych regionach świata. Dzięki temu po wpisaniu w przeglądarce adresu [www.google.com](http://www.google.com) użytkownik jest kierowany do systemu działającego najbliżej miejsca, w którym dany użytkownik się znajduje. Można więc rozpatrywać działanie tego mechanizmu jako szczególną formę rozkładania obciążenia. Oczywiście, usługa musi działać jednakowo niezależnie od tego, do którego centrum danych użytkownik się odwoła.

### 32.8. Jednolita reprezentacja danych (XML)

Jednym z ciekawszych trendów rozwojowych sieci jest upowszechnianie się **rozszerszanego języka znaczników** (XML — ang. *eXtensible Markup Language*). Początkowo format XML służył do ujednolicenia struktury dokumentów sieciowych tak, aby były one zrozumiałe dla różnych aplikacji. Zamiast definiowania zbioru znaczników umożliwiono programistom stosowanie własnych elementów, które odzwierciedlają znaczenie opisywanych pól. Na przykład dokument składający się ze znaczników <nazwisko>, <ulica>, <miejsce>, <kraj> i <kod\_pocztowy> z pewnością przechowuje dane kontaktowe innych osób. Jedną z głównych zalet dokumentów XML jest to, że są to dokumenty samoopisowe — zawierają **arkusz stylu**, który określa poprawną strukturę dokumentu.

Format XML stał się de facto standardem reprezentowania danych i jest wykorzystywany w różny sposób, często nieprzewidziany przez twórców specyfikacji. Znajduje zastosowanie na przykład w komunikacji między serwerem WWW i bazą danych. Jest także interpretowany przez niektóre systemy rozkładania obciążenia. Poza tym definiuje proces pobierania danych przez urządzenia mobilne oraz opisuje zestawy danych w systemach zarządzania sieciami.

## 32.9. Sieci społecznościowe

Na początku nowego wieku zmienił się sposób wykorzystania internetu. Model konsumentki został zastąpiony siecią powiązań między użytkownikami internetu. Początkowo większość informacji publikowanych w internecie pochodziła od **dostawców treści**, czyli firm multimedialnych. Zwykły użytkownik nie miał w nich własnego wkładu. Od roku 2000 zaczęły powstawać takie serwisy jak Facebook, Myspace i YouTube, które każdemu pozwalają na publikowanie własnych materiałów audiowizualnych. Oznacza to zwiększenie ilości danych przesyłanych do serwerów.

Zmiana sposobu pracy w sieci jest zauważalna przede wszystkim wśród młodszych użytkowników. Wiele nastolatków zaczęło tworzyć własne blogi lub zarejestrować się w jednym ze wspomnianych serwisów. Nie mniej istotny jest fakt, że spora liczba zawieranych małżeństw jest wynikiem korzystania z serwisów randkowych. Rośnie też liczba czatów i wszelkiego rodzaju systemów komunikacji międzymiędzyludzkiej.

## 32.10. Mobilność i sieci bezprzewodowe

Mobilność jest jednym z najważniejszych trendów w komunikacji sieciowej. Użytkownicy oczekują stałego dostępu do internetu. Większość hoteli oferuje połączenia internetowe swoim gościom, a linie lotnicze wprowadziły tego typu usługi nawet w samolotach. Jakość połączeń na pokładach samolotów jest tak wysoka, że można je wykorzystywać nawet do rozmów VoIP.

Konieczność zapewnienia mobilności użytkowników zwiększyła zainteresowanie technologiami bezprzewodowymi, co doprowadziło do opracowania wielu nowych standardów, w tym standardu 802.11n, który gwarantuje znacznie większą przepustowość niż jego poprzednik (802.11b). Jednak największa rewolucja jest zauważalna w telefonii komórkowej. Już niedługo telefony komórkowe nie będą korzystały z innych protokołów niż IP. Gdy operatorzy telefonii zaczną masowo wdrażać rozwiązania WiMAX, cały system będzie działał na bazie protokołu IP. To z kolei doprowadzi do ujednolicenia usług w sieciach komórkowych i internetowych.

Jak na ironię, gdy branża telefonii komórkowej wybrała protokół IP w długookresowej strategii, przemysł sieciowy nie zaadaptował rozwiązań o nazwie **mobile IP**. Większość użytkowników urządzeń mobilnych musi więc polegać na technologii Wi-Fi i oprogramowaniu VPN w połączeniach z sieciami firmowymi.

## 32.11. Cyfrowy przekaz wideo

Operatorzy sieci kablowych zastępują urządzenia analogowe rozwiązaniami cyfrowymi i wkrótce będą dostarczać sygnał telewizyjny za pośrednictwem sieci pakietowych. Wielu operatorów już dzisiaj korzysta z protokołu IP, a określenie **IPTV** staje się coraz bardziej popularne.

Zastosowanie protokołu IP do transmisji sygnałów wideo daje wiele ciekawych możliwości. Po pierwsze, łączy telewizję z internetem. Pozwala tym samym na oglądanie programów telewizyjnych za pośrednictwem komputera oraz wykorzystanie telewizora w zastępstwie komputera. Ponadto mechanizm IP ułatwia wdrażanie usług **na żądanie**, które pozwalają abonentowi na zamawianie określonych programów, sterowanie odtwarzaniem emitowanych programów (przewijanie lub zatrzymywanie przekazu) oraz nagrywanie audycji.

### 32.12. Multiemisja

Choć multiemisja w internecie nie zyskała dużej popularności, wzrost zainteresowania technologią IPTV spowodował ponowny zwrot ku multiemisji. Przyczyną jest możliwości zoptymalizowania technik dostarczania danych. Oferta operatora telewizji kablowej składa się zazwyczaj z kilkuset programów. Jednak abonent ma zazwyczaj tylko kilka odbiorników telewizyjnych, za pomocą których odbiera przekaz w danej chwili. Poza tym większość odbiorców jest zainteresowana jedynie kilkoma kanałami.

Multiemisja IP umożliwia abonentom zgłaszanie zainteresowania określonym programem przez przyłączanie się do grupy multiemisji skojarzonej z tym programem. Odbiorcy z danej okolicy są przyłączani do jednego logicznego segmentu LAN. Zatem gdy jeden z nich przyłączy się do grupy, operator telewizji kablowej rozpoczyna emisję programu w danym segmencie sieci. Stan ten utrzymuje się tak długo, dopóki którykolwiek z abonentów ogląda dany program.

*Dzięki multiemisji IP tylko jeden strumień wideo musi być dostarczany do logicznego segmentu sieci LAN. Nadawanie programu ustaje, gdy wszyscy abonenci przestaną odbierać program.*

### 32.13. Dostęp szerokopasmowy i przełączanie

Technologie dostępowe stosowane na obrzeżach internetu (takie jak DSL i modemy kablowe) pracują z przepustwościami na poziomie kilku megabitów na sekundę, czyli o dwa rzędy większymi niż w przypadku modemowych połączeń telefonicznych. W niektórych regionach Stanów Zjednoczonych dostawcy usług internetowych oferują łączą FTTH, które umożliwiają przesyłanie danych z szybkością kilku gigabitów na sekundę, czyli o trzy rzędy większą niż w przypadku linii DSL i modemów kablowych.

Przełączniki ethernetowe wykorzystywane w sieciach korporacyjnych umożliwiają przesyłanie danych do komputerów z przepływnością 1 Gb/s. Łączą o większej pojemności pracują z szybkością 10 Gb/s, a prawdopodobnie w najbliższej przyszłości będzie ona wynosiła 40 Gb/s. Takie przepływności w zupełności wystarczają do przekazywania sygnałów wideo o wysokiej rozdzielczości.

## 32.14. Przełączanie optyczne

Najważniejsze pytanie, jakie zadają sobie osoby zarządzające pracą rdzenia internetowego, to: w jaki sposób połączymy technologie elektroniczne z optycznymi? Urządzenia optyczne umożliwiają operatorom telekomunikacyjnym budowanie optycznych torów transmisyjnych, które przesyłają dane między punktami końcowymi z przepustowością 10 Gb/s. Mimo że w obecnie wykorzystywanych technologiach zestawienie toru zajmuje sporo czasu (kilka sekund), badane obecnie rozwiązania pozwolą na skrócenie tego czasu do milisekund.

Jakie korzyści wynikają z możliwości szybkiego zestawienia toru optycznego? Czy dostawcy usług internetowych wykorzystają linie światłowodowe do łączenia routerów, a technologie pakietowe będą stosowali w części dostępowej? Czy dostawca powinien tworzyć tor optyczny za każdym razem, gdy użytkownik ustanowi połączenie TCP? Pytania te stanowią podstawę nowego obszaru badań. Większość firm ISP jest bowiem przekonana o zasadności przełączania optycznego.

## 32.15. Sieć w biznesie

Większość dużych przedsiębiorstw wykorzystuje sieci komputerowe we wszystkich obszarach swojej działalności. Wspomniane sieci zmieniają również sam sposób prowadzenia biznesu. Po pierwsze, dostępność technologii RFID ma istotny wpływ na produkcję, magazynowanie i spedycję. Po drugie, dostępność wysokowydajnych przełączników warstwy 2. i technologii pakietowych przeznaczonych do transmisji głosu i obrazu pozwala na wyeliminowanie podróży służbowych i zastąpienie ich wideokonferencjami. Po trzecie, wiele firm odstępuje od klasycznej hierarchii służbowej na rzecz pracy grupowej i wspólnego podejmowania decyzji. Taka forma działalności jest wspierana przez wiele narzędzi ułatwiających interakcje w grupie.

## 32.16. Czujniki w domu i otoczeniu

Niski koszt sieci przewodowej i bezprzewodowej oraz dostępność czujników o małym poborze energii sprawiły, że realne staje się budowanie dużych sieci sensorycznych i łączenie ich z internetem. Elementy takich sieci (czujnik) są wykorzystywane do pomiaru parametrów środowiskowych (na przykład jakości powietrza i wody lub czynników atmosferycznych), śledzenia migracji zwierząt, monitorowania wzrostu upraw, monitorowania ruchu osób w budynkach i sterowania ruchem na autostradach.

Szczególnie interesującym obszarem zastosowań czujników jest wykorzystanie ich w domu do pomiaru temperatury, wilgotności oraz ostrzegania przed dymem lub tlenkiem węgla. Domowa sieć sensoryczna może być połączona z internetem, co pozwala właścielowi posesji na monitorowanie jej stanu w czasie podróży<sup>91</sup>. W niedługim czasie dostępne będą tanie czujniki, które będzie można instalować w żarówkach lub innych urządzeniach.

---

<sup>91</sup> Autor utworzył taki system w swoim domu.

### 32.17. Sieci ad hoc

Od początku istnienia sieci pakietowych wojsko amerykańskie prowadzi badania nad **sieciami ad hoc**, czyli sieciami, które same organizują swoje działanie. Zgodnie z założeniem projektu stacje bezprzewodowe powinny same wyszukiwać inne pobliskie jednostki, wybierać odpowiednią topologię i konfigurować routing gwarantujący komunikację pomiędzy wszystkimi elementami sieci. Wojskowe zastosowania takiego rozwiązania wydają się oczywiste. Każdy żołnierz mógłby nosić urządzenie bezprzewodowe, które wraz z innymi podobnymi urządzeniami budowałoby sieć komunikacyjną.

Sieci formowane na żądanie stają się również coraz bardziej potrzebne cywilom, szczególnie w obszarach wiejskich i w krajach rozwijających się. Technologia sieci ad hoc pozwala amerykańskim rolnikom na przyłączanie się do internetu. Każdy farmer instaluje urządzenie bezprzewodowe (zazwyczaj na wysokich budynkach), które pośredniczy w przekazywaniu pakietów, gdy zachodzi taka potrzeba. W krajach rozwijających się ta sama technika może posłużyć do przyłączania całych wiosek do internetu.

### 32.18. Procesory wielordzeniowe i sieciowe

Duże przepustowości są problemem dla producentów sprzętu sieciowego, którzy muszą budować systemy zdolne do szybkiego przetwarzania pakietów. Wysokiej klasy routery muszą obsługiwać pakiety nadchodzące z szybkością 10 Gb/s. Służą do tego specjalnie przygotowane układy elektroniczne (ASIC). Jednak ich projektowanie i testowanie zajmuje wiele miesięcy, a koszt wytworzenia nie należy do najniższych. Tradycyjne procesory nadają się do stosowania w mniej wydajnych urządzeniach sieciowych, takich jak routery bezprzewodowe (używane w sieciach domowych), ale nie mają dostatecznej mocy obliczeniowej, aby przekazywać ruch o większej przepływności.

Dostawcy układów scalonych oferują dwa rozwiązania. Pierwszym z nich są wielordzeniowe procesory. Rozłożenie odbieranego strumienia pakietów na  $N$  rdzeni sprawia, że jeden rdzeń przetwarza tylko  $1/N$  wszystkich pakietów. Drugie rozwiązanie to używanie **procesorów sieciowych**. Są to wydajne wielordzeniowe procesory obsługujące specjalne instrukcje związane z przetwarzaniem pakietów.

### 32.19. IPv6

Lista najnowszych trendów sieciowych nie byłaby kompletna bez wzmianki o protokole IPv6. Prace nad nim rozpoczęły się w 1993 roku i były prowadzone przez wiele lat. Początkowo zwolennicy mechanizmu IPv6 przekonywali, że jest on niezbędny, ponieważ protokół IPv4 nie nadaje się do przenoszenia danych audiowizualnych, nie gwarantuje bezpieczeństwa informacji i nie zapewnia dostatecznie dużej przestrzeni adresowej. Co rok, od czasu powstania specyfikacji IPv6, jakaś grupa naukowców lub przedstawicieli przemysłu wieszczy schyłek technologii IPv4 i nadejście IPv6. W międzyczasie jednak protokół IPv4 jest dostosowywany do bieżących wymagań. Obsługuje aplikacje multimedialne,

zapewnia taki sam poziom bezpieczeństwa, jak IPv6, a mechanizm NAT i adresowanie CIDR zmniejsza zapotrzebowanie na przestrzeń adresową. IPv4 jest nadal podstawowym protokołem internetu. Niektórzy operatorzy sieci komórkowych (szczególnie w Azji) postrzegają protokół IPv6 jako mechanizm umożliwiający wdrożenie adresowania IP w telefonach komórkowych. Nic jednak nie stoi na przeszkodzie, żeby wykorzystać do tego celu adresowanie w warstwie 2.

W czasie gdy powstawała ta książka, nie było żadnego technicznego powodu wdrażania protokołu IPv6. W praktyce zwiększyły narzuć obliczeniowy związany ze stosowaniem protokołu IPv6 mógłby nawet zmniejszyć szybkość przesyłania pakietów. Wprowadzenie mechanizmów IPv6 jest więc zależne od ekonomii. Umożliwia zrezygnowanie z funkcji NAT i posługiwanie się rzeczywistymi adresami stacji, ale wiąże się jednocześnie z koniecznością wymiany całego sprzętu i oprogramowania sieciowego. Trudno powiedzieć, kiedy klienci zdecydują, że zmiana jest uzasadniona ekonomicznie.

## 32.20. Podsumowanie

Internet wciąż ewoluje. Opracowywane są nowe aplikacje i technologie. Obecne trendy prowadzą do zwiększania szybkości transmisji, mobilności urządzeń i skalowalności usług. Aplikacje internetowe wydają się zmierzać ku sieciom społecznościowym, w których każdy użytkownik może być autorem publikowanych treści. Przedsiębiorstwa kładą nacisk na pracę grupową i wideokonferencje zastępujące podróże służbowe.

## ZADANIA

- 32.1. Wyjaśnij, w jaki sposób buforowanie treści wpływa na zwiększenie dostępności usług.
- 32.2. W której części sieci instalowany jest komponent rozkładania obciążenia?
- 32.3. Usługa udostępniana za pomocą  $N$  serwerów fizycznych niekoniecznie musi obsługiwać  $N$  razy więcej żądań niż pojedynczy serwer. Przyczyną jest korzystanie ze współdzielonych zasobów. Wymień dwa rodzaje takich zasobów.
- 32.4. Poza skalowaniem wirtualizacja serwerów pozwala również na zaoszczędzenie energii w czasie obniżonego zapotrzebowania na usługi (na przykład w weekendy). Wyjaśnij dlaczego.
- 32.5. Do czego najczęściej wykorzystywana jest komunikacja P2P?
- 32.6. Czy rozproszone centra danych znajdują zastosowanie w przedsiębiorstwach, w których obsługa każdego żądania sieciowego wymaga odwołania do bazy danych? Uzasadnij odpowiedź?
- 32.7. Podaj trzy przykłady sieciowych aplikacji społecznościowych.
- 32.8. Na czym polega ujednolicanie usług internetu i sieci telefonii komórkowej?
- 32.9. Jakie korzyści dla użytkowników wynikają z cyfrowej transmisji wideo?
- 32.10. O ile szybciej można przesyłać dane za pomocą łącza światłowodowego niż w przypadku stosowania modemów DSL lub modemów kablowych?
- 32.11. Wymień przykłady nowych trendów sieciowych w biznesie.
- 32.12. Do czego są wykorzystywane sieci sensoryczne?

- 32.13. Jakie technologie są stosowane do zapewnienia dostępu do internetu w regionach wiejskich?
- 32.14. Podaj nazwy dwóch technologii, które pozwalają na zwiększenie wydajności routerów i przełączników.
- 32.15. Dlaczego operatorzy telefonii komórkowej są szczególnie zainteresowani protokołem IPv6?

# *Dodatek A*

## *Uproszczony interfejs programistyczny*

### **Wprowadzenie**

W rozdziale 3. opisany został interfejs API, który programiści wykorzystują do tworzenia programów klienckich i serwerowych. W tym dodatku przedstawione zostało rozwiązanie alternatywne — uproszczony interfejs API, który pozwala na budowanie aplikacji sieciowych bez konieczności poznawania wszystkich szczegółów implementacyjnych interfejsu gniazd. W Dodatku znajdują się wszystkie informacje potrzebne do uruchomienia kodu. Nie trzeba znać zasad działania internetu ani protokołu TCP, aby móc z niego korzystać.

Twarzyszące omówieniu przykłady są zgodne z założeniem, że:

*Programista może tworzyć aplikacje internetowe bez znajomości zasad działania niskopoziomowych technologii sieciowych oraz protokołów komunikacyjnych.*

Celem Dodatku jest zaprezentowanie niewielkiej biblioteki funkcji wspomagających komunikację sieciową oraz przedstawienie zasad użycia tych funkcji w budowaniu aplikacji. Zamieszczony tutaj kod jest dostępny na stronie internetowej wydawnictwa, a czytelnicy książki są zachęcani do modyfikowania go i tworzenia na jego podstawie własnych aplikacji.

## Model komunikacji sieciowej

Transfer wszystkich danych w internecie jest wynikiem działania aplikacji. Programy korzystające z internetu zazwyczaj pracują w parach. Na przykład gdy użytkownik próbuje pobrać stronę internetową, zainstalowana w jego komputerze przeglądarka kontaktuje się z serwerem aplikacji WWW działającym w komputerze zdalnym. Przeglądarka generuje żądanie, na które serwer WWW odpowiada. Tylko te dwa programy „rozumieją” format komunikatu oraz jego znaczenie.

## Model klient-serwer

Komunikacja między dwoma aplikacjami za pośrednictwem internetu jest relatywnie prostym mechanizmem. Jeden program jest uruchamiany wcześniej i czeka, aż drugi program nawiąże z nim kontakt. Druga aplikacja musi znać lokalizację pierwszej. Taka forma zależności między stacjami jest nazywana relacją **klient-serwer**. Program oczekujący na kontakt jest nazywany **serwerem**. Natomiast aplikacja inicjująca wymianę danych pełni rolę **klienta**. Aby rozpocząć komunikację, klient musi wiedzieć, w jaki sposób może skontaktować się z serwerem. Lokalizacja serwera w internecie jest określana za pomocą pary identyfikatorów:

(komputer, aplikacja)

W tym zapisie wartość *komputer* identyfikuje jednostkę, w której działa serwer, a parametr *aplikacja* wskazuje program w danym komputerze. Choć sama aplikacja przetwarza obydwie wartości jako liczby binarne, użytkownicy nigdy nie posługują się takim zapisem. Używają ciągów alfanumerycznych, które program automatycznie zamienia na wartości binarne.

## Zasady komunikacji

Większość aplikacji internetowych działa (inicjuje komunikację, wymienia dane i kończy komunikację) zgodnie z kilkoma podstawowymi zasadami. Oto lista czynności realizowanych w trakcie pracy:

- Program serwerowy jest uruchamiany jako pierwszy i oczekuje na kontakt ze strony klienta.
- Klient określa lokalizację serwera i żąda ustanowienia połączenia.
- Po nawiązaniu połączenia klient i serwer wymieniają komunikaty.
- Wraz z zakończeniem wymiany informacji klient i serwer wysyłają znacznik **konca pliku** (ang. *end-of-file*), kończąc połączenie.

## Przykładowy interfejs programistyczny

W poprzednich punktach omówione zostały ogólne interakcje między dwoma aplikacjami. Trzeba je jednak przeanalizować bardziej szczegółowo. Zgodnie z obowiązującą w informatyce definicją **interfejs programowania aplikacji** (API — ang. *Application Programming Interface*) jest zbiorem operacji, z których może korzystać programista aplikacji. W specyfikacji API wymienione są wszystkie funkcje, parametry funkcji oraz informacje o przeznaczeniu poszczególnych funkcji.

Prezentacja zasad programowania sieciowego bazuje na przygotowanym wcześniej interfejsie API, którego siedem funkcji wymieniono w tabeli A.1. W dalszej części dodatku znajduje się opis interfejsu oraz przykłady aplikacji, które z niego korzystają.

**Tabela A.1.** Przykład interfejsu API złożonego z siedmiu funkcji, wystarczających do przygotowania większości aplikacji sieciowych<sup>1</sup>

Funkcja	Przeznaczenie
await_contact	Funkcja ta jest wykorzystywana przez serwer do oczekiwania na kontakt ze strony klienta.
make_contact	Funkcja ta jest wykorzystywana przez klienta do kontaktu z serwerem.
appname_to_appnum	Funkcja ta jest wykorzystywana do zamiany nazwy aplikacji na odpowiadającą jej wewnętrzną wartość binarną.
cname_to_comp	Funkcja ta jest wykorzystywana do zamiany nazwy komputera na odpowiadającą mu wewnętrzną wartość binarną.
send	Funkcja ta jest wykorzystywana przez klienta lub serwer do wysłania danych.
recv	Funkcja ta jest wykorzystywana przez klienta lub serwer do odbierania danych.
send_eof	Funkcja ta jest wykorzystywana przez klienta i serwer po zakończeniu przesyłania danych.

Uwaga: w przykładowym kodzie została użyta także funkcja `recvln`. Nie występuje ona na powyższej liście, ponieważ trudno ją uznać za oddzielną funkcję. Składa się z pojedynczej pętli, w której do czasu odebrania znacznika końca wiersza wywoływana jest funkcja `recv`.

---

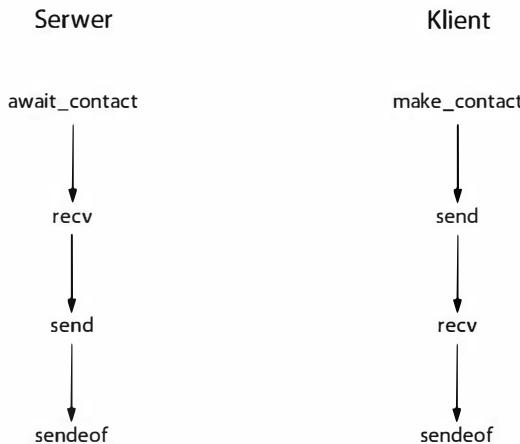
<sup>1</sup> Funkcje `send` i `recv` są udostępniane przez system operacyjny. Pozostałe funkcje interfejsu API pochodzą z przygotowanej biblioteki.

## Intuicyjna praca z interfejsem API

Serwer rozpoczyna swoje działanie od wywołania funkcji `await_contact`, która wstrzymuje wykonanie kodu do czasu nawiązania połączenia ze strony klienta. Praca programu klienckiego rozpoczyna się natomiast od wywołania funkcji `make_contact`, która odpowiada za ustanowienie połączenia. Po nawiązaniu połączenia między klientem i serwerem obydwie strony mogą wymieniać komunikaty za pomocą funkcji `send` i `recv`. Kod programów musi zostać przygotowany w taki sposób, aby było wiadomo, czy dana aplikacja ma wysyłać, czy odbierać dane. Jeśli obydwie będą oczekiwane na dostarczenie informacji bez wysyłania jakichkolwiek, zablokują się wzajemnie.

Po zakończeniu operacji wysyłania danych program wywołuje funkcję `send_eof`, która generuje znacznik końca pliku. Dostarczenie znacznika końca pliku jest w aplikacji odbiorczej sygnalizowane zwrotem wartości 0 przez funkcję `recv`. Zatem jeśli klient wywoła funkcję `send_eof`, serwer zarejestruje ten fakt, odczytując wartość 0 jako wynik wywołania funkcji `recv`. Po obustronnym wywołaniu funkcji `send_eof` komunikacja zostaje przerwana.

W zrozumieniu zasad działania interfejsu API pomocna może się okazać analiza prostego przykładu jego użycia. Rozważmy więc działanie aplikacji, która kontaktuje się z serwerem, wysyła pojedyncze żądanie i odbiera jedną odpowiedź. Odpowiednie sekwencje wywołań funkcji API po stronie klienta i serwera zostały przedstawione na rysunku A.1.



Rysunek A.1. Wywołania funkcji API podczas komunikacji, w której klient wysyła jedno żądanie i otrzymuje jedną odpowiedź serwera

## Opis interfejsu API

W kodzie opisywanym w dalszej części Dodatku, oprócz standardowych typów danych języka C, wykorzystane zostały trzy typy zadeklarowane w bibliotece API. Użycie ich uniezależnia interfejs API od konkretnego systemu operacyjnego i oprogramowania sieciowego. Wykaz wspomnianych typów wraz z ich opisami przedstawiono w tabeli A.2.

**Tabela A.2.** Trzy typy danych przykładowej biblioteki API

Nazwa typu	Przeznaczenie
appnum	Wartość binarna służąca do identyfikacji aplikacji.
computer	Wartość binarna służąca do identyfikacji komputera.
connection	Wartość wykorzystywana do identyfikacji połączenia między klientem i serwerem.

Znając trzy nazwy typów, możemy zająć się dokładniejszą analizą samej biblioteki API. W dalszej części punktu opisane zostały funkcje biblioteki. Omówieniu każdej z nich towarzyszy charakterystyczna dla języka C deklaracja, w której wymienione są typy parametrów oraz typ zwracanej wartości.

### Funkcja `await_contact`

Serwer wywołuje funkcję `await_contact`, aby rozpoczęć oczekiwanie na kontakt ze strony klienta.

```
connection await_contact(appnum a)
```

Funkcja pobiera jeden parametr typu `appnum` i zwraca wartość typu `connection`. Przekazywany w wywołaniu parametr odpowiada numerowi aplikacji po stronie serwera. Klient musi określić ten numer podczas ustanawiania połączenia. Wartość wynikowa funkcji (wartość typu `connection`) jest później wykorzystywana do przesyłania danych.

### Funkcja `make_contact`

Klient wywołuje funkcję `make_contact` w celu ustanowienia połączenia z serwerem.

```
connection make_contact(computer c, appnum a)
```

W wywołaniu trzeba określić dwa parametry. Pierwszy z nich wskazuje komputer, w którym pracuje serwer. Drugi odpowiada numerowi aplikacji, który jest wykorzystywany przez serwer działający w zdalnym komputerze. Wynik (wartość typu `connection`) umożliwia późniejsze wysyłanie danych.

### Funkcja `appname_to_appnum`

Oprogramowanie klienckie i serwerowe wykorzystuje funkcję `appname_to_appnum` do przekształcania opisowych nazw usług na odpowiadające im wartości binarne, wykorzystywane wewnętrznie. Nazwy usług są zgodne ze standardem internetowym (na przykład wartość `www` oznacza serwis WWW).

```
appnum appname_to_appnum(char *a)
```

Funkcja pobiera jeden parametr (ciąg tekstowy reprezentowany przez wskaźnik na znak [char \*]) i zwraca wartość binarną typu appnum.

### Funkcja cname\_to\_comp

Oprogramowanie klienckie wywołuje funkcję cname\_to\_comp w celu przekształcenia nazwy komputera na odpowiadającą jej wartość binarną, wykorzystywaną wewnętrznie.

```
computer cname_to_comp(char *c)
```

Funkcja pobiera jeden parametr typu char \* (ciąg tekstowy) i zwraca wartość binarną typu computer.

### Funkcja send

Zarówno klient, jak i serwer wykorzystują funkcję send do wysyłania danych przez sieć.

```
int send(connection con, char *buffer, int length, int flags)
```

Wywołanie funkcji wymaga określenia czterech parametrów. Pierwszy przekazuje wartość połączenia, zwróconą wcześniej przez funkcje await\_contact i make\_contact. Drugi jest adresem bufora z danymi przeznaczonymi do wysłania. Trzeci parametr określa rozmiar przesyłanego bloku danych (w bajtach). Natomiast czwarty ma wartość zero (w przypadku zwykłej operacji wysłania danych). Zwracany wynik odpowiada liczbie wysłanych bajtów. Wartość ujemna świadczy o wystąpieniu błędu. Po wysłaniu wszystkich danych program powinien wywołać funkcję send\_eof.

### Funkcje recv i recvln

Programy klienckie i serwerowe korzystają z funkcji recv do pobierania danych dostarczonych przez sieć.

```
int recv(connection con, char *buffer, int length, int flags)
```

Wywołanie funkcji wymaga określenia czterech parametrów. Pierwszy przekazuje wartość połączenia zwróconą wcześniej przez funkcje await\_contact i make\_contact. Drugi jest adresem bufora przeznaczonego na odebrane dane. Trzeci parametr zwraca rozmiar odebranego bloku danych (w bajtach). Natomiast czwarty ma wartość zero (w przypadku standardowego transferu danych). Funkcja recv zwraca liczbę bajtów, które zostały zapisane w buforze. Wartość zero oznacza odebranie znacznika końca pliku. Z kolei liczba

ujemna sygnalizuje wystąpienie błędu. W prezentowanym dalej przykładzie wykorzystana została również funkcja `recvln`, która wywołuje wielokrotnie funkcję `recv`, aż do odczytania całego wiersza tekstu. Prototyp funkcji `recvln` ma postać:

```
int recvln(connection con, char *buffer, int length)
```

## Funkcja send\_eof

Po zakończeniu wymiany informacji zarówno klient, jak i serwer muszą wywołać funkcję `send_eof` do poinformowania drugiej strony o zakończeniu komunikacji. Odebranie znacznika końca pliku jest sygnalizowane w programie zdalnym zwróceniem wartości zero jako wyniku wywołania funkcji `recv`.

```
int send_eof(connection con)
```

Parametr funkcji odpowiada połączeniu, które zostało ustanowione wcześniej za pomocą funkcji `await_contact` lub `make_contact`. W przypadku wystąpienia błędu wynikiem wywołania jest wartość ujemna. Poprawne wykonanie operacji kończy się zwróceniem liczby nieujemnej.

## Wykorzystanie typów biblioteki API

W tabeli A.3 zestawiono typy parametrów wykorzystywane w każdej funkcji interfejsu API. Ponadto w każdym wierszu został określony typ zwracanej wartości, a w ostatniej kolumnie zapisane są typy danych przekazywanych na trzeciej i czwartej pozycji wywołania. Mimo że funkcje `send` i `recv` pobierają cztery parametry, funkcja `recvln` wymaga określenia jedynie trzech wartości.

Tabela A.3. Zestawienie typów parametrów i zwracanych wartości funkcji przykładowej biblioteki API

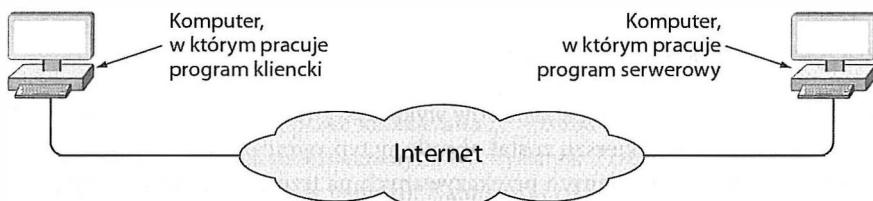
Nazwa funkcji	Typ zwracanej wartości	Typ 1. parametru	Typ 2. parametru	Typ 3. i 4. parametru
<code>await_contact</code>	<code>connection</code>	<code>appnum</code>	-	-
<code>make_contact</code>	<code>connection</code>	<code>computer</code>	<code>appnum</code>	-
<code>appname_to_appnum</code>	<code>appnum</code>	<code>char *</code>	-	-
<code>cname_to_comp</code>	<code>computer</code>	<code>char *</code>	-	-
<code>send</code>	<code>int</code>	<code>connection</code>	<code>char *</code>	<code>int</code>
<code>recv</code>	<code>int</code>	<code>connection</code>	<code>char *</code>	<code>int</code>
<code>recvln</code>	<code>int</code>	<code>connection</code>	<code>char *</code>	<code>int</code>
<code>send_eof</code>	<code>int</code>	<code>connection</code>	-	-

W kolejnych punktach zostały przedstawione przykłady programów, które demonstrują sposób wykorzystania interfejsu API do utworzenia komunikujących się ze sobą aplikacji klienckich i serwerowych. Aby skrócić kod i ułatwić jego interpretację, pominięto w nim mechanizmy sprawdzania poprawności informacji podawanych w wierszu polecenia. Warto więc jako ćwiczenie uzupełnić kod o operację sprawdzenia wartości i poinformowania użytkownika o ewentualnym błędzie.

## Kod aplikacji echo

Działanie pierwszej z prezentowanych aplikacji jest bardzo proste — klient wysyła dane do serwera, a serwer odsyła je z powrotem do klienta. Program kliencki cyklicznie monitoruje użytkownika o wprowadzenie dowolnego tekstu, wysyła podany ciąg znaków do serwera i wyświetla dane odesłane przez serwer. Choć aplikacja nie jest szczególnie użyteczna dla większości użytkowników, podobne narzędzia są często wykorzystywane przez administratorów do testowania połączeń sieciowych.

Podobnie jak wszystkie pozostałe programy opisane w Dodatku, również aplikacja echo bazuje na standardowych protokołach internetowych. Oznacza to, że programy kliencki i serwerowy można uruchomić w dowolnym komputerze przyłączonym do internetu, zgodnie z rysunkiem A.2.



Rysunek A.2. Klient i serwer aplikacji echo mogą pracować w dowolnych komputerach

Aby uruchomić program serwerowy, użytkownik musi wybrać numer aplikacji z przedziału od 1 do 32767, ale taki, który nie został wcześniej wykorzystany przez inną aplikację. Numer ten należy podać jako parametr w wierszu polecenia. Założymy, że program ten zostanie uruchomiony w komputerze *komputer1.lab.helion.pl* z numerem 20 000. Polecenie powinno wówczas mieć następującą treść:

```
echoserver 20000
```

Jeżeli numer 20 000 będzie zajęty przez inną aplikację, program serwera wyświetli komunikat o błędzie i zakończy swoje działanie. Aby go uruchomić, konieczne będzie wprowadzenie innego numeru.

Po uruchomieniu serwera można uruchomić aplikację kliencką. Należy w tym celu określić nazwę komputera, w którym pracuje serwer, oraz związany z nim numer aplikacji. Na przykład do nawiązania połączenia z serwerem uruchomionym w poprzednim przykładzie należałoby wykonać następującą instrukcję (w dowolnym komputerze przyłączonym do internetu):

```
echoclient komputer1.lab.helion.pl 20000
```

## Kod serwera aplikacji echo

Kod serwera jest zapisany w pliku *echoserver.c*. Mimo że zawiera on komentarze i puste wiersze zwiększające przejrzystość listingu, mieści się na jednej stronie książki. Zasadniczy program składa się z siedmiu wierszy kodu. Pozostała część to instrukcje sprawdzające, czy został poprawnie uruchomiony:

```
/* echoserver.c */
#include <stdlib.h>
#include <stdio.h>
#include <cnaiaapi.h>

#define BUFFSIZE 256

/*
 * Program: echoserver
 * Działanie: oczekiwanie na połoczenie i odeslanie danych dostarczonych przez klienta
 * Użycie: echoserver <numer_apl>
 *
*/
int
main(int argc, char *argv[])
{
    connection conn;
    int len;
    char buff[BUFFSIZE];

    if (argc != 2) {
        (void) fprintf(stderr, "użycie: %s <numer_apl>\n",
                      argv[0]);
        exit(1);
    }

    /* oczekiwanie na połoczenie ze strony klienta */
    conn = await_contact((appnum) atoi(argv[1]));
    if (conn < 0)
        exit(1);

    /* iteracyjne odsyłanie danych aż do końca pliku */
    while ((len = recv(conn, buff, BUFFSIZE, 0)) > 0)
        (void) send(conn, buff, len, 0);
    send_eof(conn);
    return 0;
}
```

Jak nietrudno zauważyc, program pobiera jeden parametr wiersza poleceń, który określa numer aplikacji. W języku C parametry wiersza poleceń są przekazywane do kodu programu za pośrednictwem tablicy ciągów tekstowych (*argv*) wraz z informacją o liczbie parametrów (*argc*). Wpisany przez użytkownika numer jest więc pobierany za pomocą

odwołania `argv[1]`, a za przekształcenie go z ciągu tekstowego w wartość liczbową odpowiada standardowa funkcja języka C `atoi`. Wynik wywołania funkcji `atoi` jest przekazywany jako parametr funkcji `await_contact`. Po wykonaniu operacji `await_contact` serwer cyklicznie wywołuje funkcje `recv`, pobierając dostarczane dane, i `send` w celu odesłania odebranych wartości. Wykonywanie pętli kończy się w chwili odebrania znacznika końca pliku i zwrócenia przez funkcję `recv` wartości zero. Serwer wysyła wówczas znacznik końca pliku jednostki zdalnej i przerywa swoje działanie.

## Kod klienta aplikacji echo

Kod aplikacji klienckiej jest zapisany w pliku `echoclient.c`. Choć listing nie jest tak krótki, jak w przypadku serwera, zasadnicza część programu zajmuje tylko kilka wierszy kodu.

```
/*
 * echoclient.c
 *include <stdlib.h>
 *include <stdio.h>
 #include <cnaiaapi.h>

#define BUFFSIZE      256
#define INPUT_PROMPT  "Wysylane> "
#define RECEIVED_PROMPT "Odebrane> "

int readln(char *, int);

/*
 *-----*
 * Program: echoclient
 * Dzialanie: połoczenie z serwerem, wyslanie danych użytkownika i wyświetlenie
 * odpowiedzi
 * Użycie: echoclient <nazwa_komputera> [numer_apl]
 * Uwaga: Numer portu jest opcjonalny. W przypadku braku parametru zostanie użyta
 * wartość domyślna (7).
 *-----*
 */
int
main(int argc, char *argv[])
{
    computer      comp;
    appnum        app;
    connection    conn;

    char buff[BUFFSIZE];
    int expect, received, len;
    if (argc < 2 || argc > 3) {
        (void) fprintf(stderr, "użycie: %s <nazwa_komputera>
                           ↳[numer_apl]\n", argv[0]);
        exit(1);
    }

    /* zamiana parametrów na format binarny */
```

```
comp = cname_to_comp(argv[1]);
if (comp == -1)
    exit(1);
if (argc == 3)
    app = (appnum) atoi(argv[2]);
else
    if ((app = appname_to_appnum("echo")) == -1)
        exit(1);

/* nawiązanie połączenia z serwerem */

conn = make_contact(comp, app);
if (conn < 0)
    exit(1);

(void) printf(INPUT_PROMPT);
(void) fflush(stdout);

/* pętla: odczyt danych wprowadzonych przez użytkownika, wysłanie ich do serwera, */
/* odbiór odpowiedzi z serwera i wyświetlenie jej na ekranie */

while((len = readln(buff, BUFFSIZE)) > 0) {

    /* wysłanie wprowadzonych danych do serwera */

    (void) send(conn, buff, len, 0);
    (void) printf(RECEIVED_PROMPT);
    (void) fflush(stdout);

    /* odczyt i wyświetlenie wartości odebranych od serwera */

    expect = len;

    for (received = 0; received < expect;) {
        len = recv(conn, buff, (expect - received) <
                    BUFFSIZE ?
                    (expect - received) : BUFFSIZE, 0);
        if (len < 0) {
            send_eof(conn);
            return 1;
        }
        (void) write(STDOUT_FILENO, buff, len);
        received += len;
    }
    (void) printf("\n");
    (void) printf(INPUT_PROMPT);
    (void) fflush(stdout);
}

/* iteracje kończą się po napotkaniu w strumieniu wejściowym znacznika EOF */

(void) send_eof(conn);
(void) printf("\n");
return 0;
}
```

Program kliencki można uruchomić z jednym parametrem lub z dwoma parametrami. Pierwszy parametr odpowiada nazwie komputera, w którym pracuje aplikacja serwerowa. Drugi parametr, o ile został podany, wyznacza numer aplikacji (określony podczas uruchamiania serwera). W przypadku braku numeru program wywołuje funkcję `appname_to_appnum` z wartością `echo`.

Po przekształceniu do formatu binarnego parametry są przekazywane do funkcji `make_contact`, która ustanawia połączenie z serwerem. Następnie na ekranie jest wyświetlane monit o wprowadzenie danych, a sam kod zaczyna pracować w pętli. Odczytuje wiersz tekstu wpisany przez użytkownika, wysyła pozyskane dane do serwera i wyświetla odpowiedź serwera wraz z nowym monitem o wpisanie informacji wejściowych. Gdy użytkownik przestanie wprowadzać teksty (funkcja `readln` zwróci zerową liczbę znaków), program wywołuje funkcję `send_eof` i kończy swoje działanie.

Kod aplikacji klienckiej jest nieco bardziej skomplikowany z kilku powodów. Po pierwsze, program kliencki korzysta z funkcji `readln` do odczytywania kolejnych wierszy tekstu wprowadzanych przez użytkownika. Po drugie, sprawdza wynik wywołania każdej funkcji i kończy swoje działanie w przypadku wykrycia błędu. Po trzecie, wykonuje funkcję `fflush`, aby się upewnić, że cała treść przeznaczona do wyświetlenia została zaprezentowana i nic nie gromadzi się w buforze. Po czwarte (i najważniejsze), działanie klienta nie ogranicza się tylko do pojedynczego wywołania funkcji `recv` w celu odebrania danych z serwera. Ta operacja jest powtarzana cyklicznie (w pętli), aż do odebrania takiej liczby bajtów, jaka została wysłana.

Zastosowanie wielu wywołań funkcji `recv` wynika z bardzo ważnej cechy omawianego interfejsu API:

*Odbiorca nie może założyć, że dane będą nadchodziły w blokach o takich samych rozmiarach, w jakich były wysyłane. Wywołanie funkcji `recv` może się zakończyć zwróceniem mniejszej ilości danych, niż przekazano do funkcji `send` podczas wysyłania informacji.*

Powyższa uwaga wyjaśnia specyficzny sposób działania funkcji `recv`. Funkcja zwraca bowiem dane w takich porcjach, w jakich zostały one do niej dostarczone, czyli po ewentualnym podzieleniu na mniejsze pakiety. Co ciekawe, dopuszczalna jest również sytuacja odwrotna. Kilkukrotne wywołanie funkcji `send` może spowodować dostarczenie wszystkich pakietów, zanim odbiorca wywoła funkcję `recv`. To z kolei oznacza, że wszystkie dane zostaną zwrócone w wyniku jednego wywołania funkcji `recv`.

## Kod serwera czatu

Drugi prezentowany program jest uproszczoną aplikacją **czatu**. Czaty internetowe umożliwiają grupom osób komunikowanie się za pomocą krótkich informacji tekstowych, które po wprowadzeniu w jednym komputerze są wyświetlane na ekranie innych uczestników konwersacji. Przedstawiona aplikacja jest uproszczoną wersją mechanizmu, ponieważ

zapewnia wymianę informacji jedynie między dwoma użytkownikami (tekst wprowadzony przez jedną osobę jest wyświetlany na ekranie drugiego użytkownika i odwrotnie). Podobnie jak omawiana wcześniej aplikacja echo, również ten program działa poprawnie w dowolnym komputerze przyłączonym do internetu. Działanie mechanizmu rozpoczyna użytkownik, który uruchomi program serwera. Musi w tym celu określić numer aplikacji. Założymy, że serwer zostanie uruchomiony w systemie *komputer2.lab.helion.pl* za pomocą polecenia:

```
chatserver 25000
```

Wówczas użytkownik drugiego komputera powinien uruchomić program kliencki w następujący sposób:

```
chatclient komputer2.lab.helion.pl 25000
```

Podczas tworzenia programu przyjęto założenie, że użytkownicy będą wprowadzali informacje na zmianę. Dzięki temu kod jest krótszy i bardziej przejrzysty. Klient i serwer wyświetlają monit o wprowadzenie danych za każdym razem, gdy druga strona oczekuje na dostarczenie nowych informacji. Wymianę komunikatów rozpoczyna użytkownik programu klienckiego. Gdy wprowadzi dowolny wiersz tekstu, aplikacja wysyła go do serwera, po czym role się odwracają. Naprzemienne wysyłanie wiadomości kończy się w chwili, gdy jeden z programów dostarczy znacznik końca pliku.

Sam kod nie jest skomplikowany. Serwer rozpoczyna działanie od oczekiwania na kontakt ze strony klienta. Następnie rozpoczyna wykonywanie pętli, w której pobiera i wyświetla wiersz tekstu przesyłany przez klienta, odczytuje informacje wprowadzane z klawiatury i wysyła je do drugiego komputera. Do chwili odebrania znacznika końca pliku praca programu polega na naprzemennym wyświetlaniu danych dostarczonych od klienta i wysyłaniu informacji wprowadzanych przez użytkownika.

Działanie kodu klienckiego rozpoczyna się od ustanowienia połączenia z serwerem. Następnie program wykonuje kod pętli. W każdym cyklu wyświetla monit o wprowadzenie wiersza tekstu, odczytuje informacje wpisywane za pomocą klawiatury, wysyła je do serwera, a następnie odbiera i wyświetla wiadomość wygenerowaną po stronie serwera. Praca klienta sprowadza się więc do naprzemennego wysyłania wiersza tekstu wprowadzonego przez użytkownika i wyświetlania tekstu odebranego od serwera.

Kod serwera jest zapisany w pliku *chatserver.c*.

```
/* chatserver.c */

#include <stdlib.h>
#include <stdio.h>
#include <cnaiaapi.h>

#define BUFFSIZE 256
#define INPUT_PROMPT      "Wysylane > "
#define RECEIVED_PROMPT   "Odebrane> "

int recvln(connection, char *, int);
int readln(char *, int);

/*
*
```

```

* Program: chatserver
* Działanie: oczekiwanie na połczenie ze strony klienta i prowadzenie czatu
* Użycie: chatserver <numer_apl>
*
*-----
*/
int
main(int argc, char *argv[])
{
    connection conn;
    int         len;
    char        buff[BUFFSIZE];

    if (argc != 2) {
        (void) fprintf(stderr, "użycie: %s <numer_apl>\n",
                      argv[0]);
        exit(1);
    }

    (void) printf("Serwer czatu oczekuje na połczenie.\n");

    /* oczekiwanie na połczenie ze strony klienta */

    conn = await_contact((appnum) atoi(argv[1]));
    if (conn < 0)
        exit(1);
    (void) printf("Połczenie ustanowione.\n");

    /* pętla: odczyt danych przesyłanych przez klienta i pobranie danych od użytkownika */

    while((len = recvln(conn, buff, BUFFSIZE)) > 0) {
        (void) printf(RECEIVED_PROMPT);
        (void) fflush(stdout);
        (void) write(STDOUT_FILENO, buff, len);

        /* wysłanie wiersza tekstu do programu chatclient */

        (void) printf(INPUT_PROMPT);
        (void) fflush(stdout);
        if ((len = readln(buff, BUFFSIZE)) < 1)
            break;
        buff[len - 1] = '\n';
        (void) send(conn, buff, len, 0);
    }

    /* iteracje kończą się po napotkaniu w strumieniu wejściowym znacznika EOF */
    /* lub po przerwaniu połczenia */

    (void) send_eof(conn);
    (void) printf("\nPołczenie zakończone.\n\n");
    return 0;
}

```

Funkcje `recvln` i `readln` zwiększą przejrzystość kodu. Obydwie zawierają pętle, które kończą się w chwili wprowadzenia całego wiersza tekstu lub napotkania znacznika końca pliku. Funkcja `recvln` wywołuje wewnętrznie funkcję `recv` (aby odebrać dane dostarczone w ramach połączenia sieciowego), a funkcja `readln` bazuje na funkcji `read` (odczytującej znaki wprowadzane za pomocą klawiatury).

Ogólna struktura serwera czatu jest podobna do budowy serwera aplikacji echo, przedstawionej wcześniej. Tak samo jak w poprzednim przypadku podczas uruchamiania serwera trzeba podać jeden parametr, który odpowiada numerowi aplikacji. Gdy klient ustanowi połączenie, serwer wyświetla powiadomienie o tym zdarzeniu i rozpoczyna wykonywanie pętli. W każdej iteracji odbiera wiersz tekstu z połączenia sieciowego, wyświetla go na ekranie, odczytuje wiersz tekstu wprowadzany z klawiatury i wysyła w ramach połączenia sieciowego. W chwili odebrania znacznika końca pliku serwer wysyła taki sam znacznik do aplikacji zdalnej i przerywa swoje działanie.

## Kod klienta czatu

Kod programu klienckiego został zapisany w pliku `chatclient.c`. Jak można się było spodziewać, listing jest nieznacznie dłuższy niż w przypadku serwera.

```
/* chatclient.c */

#include <stdlib.h>
#include <stdio.h>
#include <cnaiapi.h>

#define BUFFSIZE 256
#define INPUT_PROMPT    "Wysyłane > "
#define RECEIVED_PROMPT "Odebrane> "

int recvln(connection, char *, int);
int readln(char *, int);

/*
 * Program: chatclient
 * Działanie: nawiązanie połączenia z serwerem i prowadzenie czatu
 * Użycie: chatclient <nazwa_komputera> <numer_apl>
 */
int
main(int argc, char *argv[])
{
    computer comp;
    connection conn;
    char buff[BUFFSIZE];
    int len;

    if (argc != 3) {
```

```

(void) fprintf(stderr, "użycie: %s <compname>
    ↵<appnum>\n",
argv[0]);
exit(1);
}

/* przekształcenie nazwy komputera na format binarny (comp) */

comp = cname_to_comp(argv[1]);
if (comp == -1)
    exit(1);

/* ustanowienie połączenia z programem chatserver */

conn = make_contact(comp, (appnum) atoi(argv[2]));
if (conn < 0)
    exit(1);
(void) printf("Połączenie ustanowione.\n");
(void) printf(INPUT_PROMPT);
(void) fflush(stdout);

/* pętla: pobranie danych od użytkownika i odbiór danych z serwera */

while((len = readln(buff, BUFFSIZE)) > 0) {
    buff[len - 1] = '\n';
    (void) send(conn, buff, len, 0);

    /* odebranie i wyświetlenie wiersza dostarczonego z serwera */

    if ((len = recvln(conn, buff, BUFFSIZE)) < 1)
        break;
    (void) printf(RECEIVED_PROMPT);
    (void) fflush(stdout);
    (void) write(STDOUT_FILENO, buff, len);
    (void) printf(INPUT_PROMPT);
    (void) fflush(stdout);
}

/* iteracje kończą się po napotkaniu w strumieniu wejściowym znacznika EOF *
/* lub po przerwaniu połączenia */

(void) printf("\nPołączenie zakończone.\n");
(void) send_eof(conn);
exit(0);
}

```

Działanie programu klienckiego rozpoczyna się od nawiązania połączenia z serwerem. Po wykonaniu tego zadania aplikacja rozpoczyna wykonywanie pętli, w której odczytuje informacje wprowadzane z klawiatury, wysyła je do serwera, odbiera dane z serwera i wyświetla na ekranie. Iteracje są ponawiane do chwili odebrania znacznika końca pliku (z serwera lub z klawiatury). Wówczas program kliencki sam wysyła znacznik końca pliku i przerywa swoją pracę.

## Aplikacja WWW

Ostatnia z prezentowanych aplikacji demonstruje wymianę danych w odwołaniach do serwerów WWW. Aby uruchomić program serwerowy, użytkownik musi zapisać w wierszu polecenia numer aplikacji. Założmy, że serwer powinien działać w komputerze o adresie *komputer3.lab.helion.pl* z numerem 27 000. Polecenie uruchomienia programu jest wówczas następujące:

```
webserver 27000
```

Po stronie klienckiej trzeba określić nazwę komputera, nazwę dokumentu oraz numer aplikacji:

```
webclient komputer3.lab.helion.pl /index.html 27000
```

Mimo że listing kodu źródłowego nie jest długi, serwer spełnia założenia standardowych protokołów. Można się więc do niego odwoływać za pośrednictwem tradycyjnych (kомерcyjnych) przeglądarek internetowych. Na przykład aby użyć standardowej przeglądarki internetowej zamiast przykładowego kodu klienckiego, wystarczy w polu adresu URL wpisać następujący wiersz:

```
http://komputer3.lab.helion.pl:27000/index.html
```

W celu maksymalnego skrócenia kodu przyjęto kilka założeń upraszczających jego działanie. Po pierwsze, serwer dostarcza tylko trzy strony, a każda z nich składa się jedynie z tekstu. Po drugie, treść każdej ze stron została trwale zapisana w samym kodzie. Zmiana zawartości strony wymaga więc ponownej komplikacji kodu serwera (ograniczenia te można usunąć w ramach samodzielnego ćwiczeń).

Największe ograniczenie prezentowanej aplikacji wynika jednak ze sposobu działania klienta. W przeciwieństwie do tradycyjnych przeglądarek, przedstawiony program nie zawiera interpretera kodu stron internetowych. Jego działanie sprowadza się jedynie do wyświetlenia dokumentu źródłowego. Mimo tego nadaje się do pobierania stron z komer cyjnych serwerów WWW. Można go wykorzystać do wyświetlania kodu źródłowego dowolnie wybranej strony w internecie.

## Kod klienta WWW

Kod programu klienckiego jest zapisany w pliku *webcontent.c*.

```
/* webclient.c */

#include <stdlib.h>
#include <stdio.h>
#include <cnaiapi.h>

#define BUFFSIZE 256

/*
 * 
 * Program: webclient
```

```

* Działanie: pobranie strony z serwera i przekazywanie treści z nagłówkami do
* standardowego strumienia wyjściowego
* Użycie: webclient <nazwa_komputera> <strona> [numer_apl]
* Uwaga: Numer aplikacji (numer_apl) jest opcjonalny. W przypadku braku parametru
* zostanie użyta wartość domyślna (80).
*
*-----
*/

```

```

int
main(int argc, char *argv[])
{
    computer    comp;
    appnum     app;
    connection conn;

    char buff[BUFFSIZE];
    int len;
    if (argc < 3 || argc > 4) {
        (void) fprintf(stderr, "%s%s%s", "użycie: ",
                      argv[0],
                      " <nazwa_komputera> <strona> [numer_apl]\n");
        exit(1);
    }
    /* zamiana parametrów na wartości binarne */

    comp = cname_to_comp(argv[1]);
    if (comp == -1)
        exit(1);
    if (argc == 4)
        app = (appnum) atoi(argv[3]);
    else
        if ((app = appname_to_appnum("www")) == -1)
            exit(1);

    /* nawiązanie połączenia z serwerem */

    conn = make_contact(comp, app);
    if (conn < 0)
        exit(1);

    /* wysłanie żądania HTTP/1.0 do serwera */

    len = sprintf(buff, "GET %s HTTP/1.0\r\n\r\n", argv[2]);
    (void) send(conn, buff, len, 0);

    /* przekazanie wszystkich odebranych danych do strumienia stdout */

    while((len = recv(conn, buff, BUFFSIZE, 0)) > 0)
        (void) write(STDOUT_FILENO, buff, len);
    return 0;
}

```

Kod programu klienckiego jest bardzo prosty. Po ustanowieniu połączenia z serwerem aplikacja wysyła żądanie o treści:

```
GET /strona HTTP/1.0 CRLF CRLF
```

w którym ciąg *strona* odpowiada nazwie pobieranej strony (na przykład *index.html*), a element *CRLF* symbolizuje znaki powrotu kurSORA i zmiany wiersza. Po wysłaniu żądania program oczekuje, aż serwer dostarczy kod strony, a następnie wyświetla dokument na ekranie.

## Kod serwera WWW

Kod (bardzo uproszczonego) serwera WWW został zapisany w pliku *webserver.c*. Obejmuje on treść trzech stron oraz instrukcje potrzebne do wygenerowania odpowiedzi na żądanie.

```
/* webserver.c */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cnaiaapi.h>

#if defined(LINUX) || defined(SOLARIS)
#include <sys/time.h>
#endif

#define BUFFSIZE      256
#define SERVER_NAME   "Demonstracyjny serwer WWW"
#define ERROR_400     "<html><head></head><body><h1>Błąd
                    400</h1><p>Serwer nie może zinterpretować
                    żądań.</p></body></html>\n"
#define ERROR_404     "<html><head></head><body><h1>Błąd
                    404</h1><p>Nie znaleziono dokumentu</p></body></html>\n"
#define HOME_PAGE     "<html><head></head><body><h1>Witaj na
                    serwerze demonstracyjnym</h1><p>Polecamy strony:
                    <ul><li><a href=\"http://helion.pl\">Strona główna
                    Wydawnictwa Helion</a></li><a
                    href=\"http://helion.pl/kategorie/sieci-komputerowe/
                    budowa-sieci\">>Lista książek na temat sieci
                    komputerowych</a></ul></body></html>\n"
#define TIME_PAGE      "<html><head></head><body><h1>Aktualny
                    czas: %s</h1></body></html>\n"

int recvln(connection, char *, int);
void send_head(connection, int, int);

/*
*
* Program: webserver
* Działanie: udostępnianie wstępnie zdefiniowanych stron internetowych
* Użycie: webserver <numer_apl>
```

```

*
*-----*
*/
int
main(int argc, char *argv[])
{
    connection conn;
    int             n;
    char           buff[BUFFSIZE], cmd[16], path[64],
    ↳vers[16];
    char           *timestr;
#if defined(LINUX) || defined(SOLARIS)
    struct timeval tv;
#elif defined(WIN32)
    time_t          tv;
#endif

    if (argc != 2) {
        (void) fprintf(stderr, "usage: %s <appnum>\n",
        ↳argv[0]);
        exit(1);
    }

    while(1) {

        /* oczekiwanie na połczenie ze strony klienta */

        conn = await_contact((appnum) atoi(argv[1]));
        if (conn < 0)
            exit(1);

        /* odczytanie i interpretacja wiersza żądania */

        n = recvln(conn, buff, BUFFSIZE);
        sscanf(buff, "%s %s %s", cmd, path, vers);

        /* pominięcie nagłówków - odczyt do wiersza złożonego jedynie ze znaków \r\n */

        while((n = recvln(conn, buff, BUFFSIZE)) > 0) {
            if (n == 2 && buff[0] == '\r' && buff[1]
            ↳== '\n')
                break;
        }

        /* sprawdzenie, czy nie wystąpił niespodziewany koniec pliku */

        if (n < 1) {
            (void) send_eof(conn);
            continue;
        }

        /* sprawdzenie, czy żądanie jest zrozumiałe */

        if (strcmp(cmd, "GET") || (strcmp(vers,
        ↳"HTTP/1.0")) &&

```

```

                strcmp(vers,
                ↳"HTTP/1.1")) {
    send_head(conn, 400, strlen(ERROR_400));
    (void) send(conn, ERROR_400,
    ↳strlen(ERROR_400), 0);
    (void) send_eof(conn);
    continue;
}

/* wyslanie żądanej strony lub komunikat o błędzie, jeśli strona nie istnieje */

if (strcmp(path, "/") == 0) {
    send_head(conn, 200, strlen(HOME_PAGE));
    (void) send(conn, HOME_PAGE,
    ↳strlen(HOME_PAGE), 0);
} else if (strcmp(path, "/time") == 0) {
#ifndef LINUX || !defined(SOLARIS)
    gettimeofday(&tv, NULL);
    timestr = ctime(&tv.tv_sec);
#else defined(WIN32)
    time(&tv);
    timestr = ctime(&tv);
#endif
    (void) sprintf(buff, TIME_PAGE, timestr);
    send_head(conn, 200, strlen(buff));
    (void) send(conn, buff, strlen(buff), 0);
} else { /* nie znaleziono strony */
    send_head(conn, 404, strlen(ERROR_404));
    (void) send(conn, ERROR_404,
    ↳strlen(ERROR_404), 0);
}
(void) send_eof(conn);
}

/*
* send_head - wyslanie nagłówka HTTP 1.0 z kodem statusu i informacją o długości treści
*/
void
send_head(connection conn, int stat, int len)
{
    char *statstr, buff[BUFFSIZE];

/* zamiana kodu statusowego na ciąg tekstowy */

switch(stat) {
    case 200:
        statstr = "OK";
        break;
    case 400:
        statstr = "Bad Request";
        break;
    case 404:
        statstr = "Not Found";
        break;
}

```

```

        default:
            statstr = "Unknown";
            break;
    }

/*
* Wysłanie odpowiedzi HTTP/1.0 z nagłówkami Server, Content-Length,
* i Content-Type.
*/
(void) sprintf(buff, "HTTP/1.0 %d %s\r\n", stat,
➥statstr);
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "Server: %s\r\n", SERVER_NAME);
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "Content-Length: %d\r\n", len);
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "Content-Type: text/html\r\n");
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "\r\n");
(void) send(conn, buff, strlen(buff), 0);
}

```

Choć kod serwera WWW może się wydawać bardziej skomplikowany niż poprzednie programy, większa część listingu odnosi się do samych stron WWW, a nie do mechanizmów komunikacji sieciowej. Poza odczytaniem i zinterpretowaniem żądania serwer musi bowiem wygenerować „nagłówek” oraz treść odpowiedzi. Nagłówek składa się z kilku wierszy tekstu zakończonych znakami powrotu kurSORA i nowego wiersza. Oto zawartość nagłówka:

```

HTTP/1.0 status ciąg_statusu CRLF
Server: Demonstracyjny serwer WWW CRLF
Content-Length: rozmiar_danych CRLF
Content-Type: text/html CRLF
CRLF

```

Wartość *rozmiar\_danych* odpowiada rozmiarowi strony (wysyłanej po nagłówku) wyrażonemu w bajtach.

Za wygenerowanie nagłówka odpowiada procedura *send\_head*. Przekazywany do niej parametr *stat* zawiera liczbowy kod statusu, a parametr *len* przechowuje informację o długości treści. Odpowiedni tekst komunikatu jest zapisywany w zmiennej *statstr* w wyniku wykonania instrukcji *switch* (na podstawie liczbowego kodu statusowego). Do wygenerowania całego nagłówka wykorzystano standardową funkcję języka C *sprintf*. Z kolei dostarczeniem poszczególnych wierszy nagłówka w ramach połączenia z aplikacją kliencką zajmuje się funkcja *send*.

Dodatkowa złożoność kodu wynika również z uwzględnienia mechanizmów obsługi błędów (komunikaty o błędach muszą być wysyłane w formacie zrozumiałym dla przeglądarki). Jeśli serwer odbierze niepoprawnie sformowane żądanie, odeśle komunikat z kodem

400. Jeśli natomiast użytkownik zażąda dostarczenia nieistniejącej strony, do przeglądarki zostanie odesłany komunikat o kodzie 404.

W przeciwnieństwie do przedstawionych wcześniej rozwiązań, program serwera nie kończy działania po obsłużeniu jednego żądania. Jego praca jest kontynuowana dzięki pętli nieskończonej, która w każdej iteracji wywołuje funkcję `await_contact`, oczekując na połączenie ze strony klienta. Gdy połączenie zostanie ustanowione, program wywołuje funkcję `recvln` odpowiedzialną za dostarczenie treści żądania. Następnie wykonuje funkcję `send`, która wysyła odpowiedź. Po tej operacji rozpoczyna się nowy cykl pętli i program ponownie czeka na dostarczenie żądania. Dzięki temu serwer działa nieskończennie długo (tak jak serwery komercyjne).

## Obsługa wielu połączeń z użyciem funkcji `select`

Zaprezentowany interfejs API poprawnie obsługuje interakcje typu jeden-do-jednego zachodzące pomiędzy klientem i serwerem. Nie pozwala jednak na realizację połączeń typu jeden-do-wielu. Zastanówmy się, z czego to wynika. Aby doprowadzić do ustanowienia wielu połączeń, aplikacja musi wielokrotnie wywołać funkcję `make_contact`, przekazując różne wartości parametrów `computer` i `appnum`. Jednak po ich ustanowieniu nie może określić, w którym z połączeń nadąają pierwsze dane do przetworzenia. Program nie może więc wywołać funkcji `recv`, ponieważ blokuje ona wykonywanie kodu do czasu odebrania danych ze wskazanego połączenia.

W wielu systemach operacyjnych do rozwiązania tego problemu służy funkcja `select`. Jej działanie odnosi się do zbioru połączeń i polega na wstrzymaniu wykonywania programu do czasu, gdy w ramach jednego z połączeń zostaną dostarczone jakiekolwiek dane. Funkcja zwraca wówczas wartość informującą o tym, w ramach którego połączenia odebrano dane (dzięki temu wywołanie funkcji `recv` nie spowoduje wstrzymania programu).

Jako przykład przeanalizujmy działanie aplikacji, która musi odbierać żądania i wysyłać odpowiedzi w ramach dwóch połączeń. Ogólna struktura kodu programu byłaby wówczas następująca:

```
Wywołaj make_contact, aby ustanowić połączenie 1;
Wywołaj make_contact, aby ustanowić połączenie 2;
Powtarzaj nieskończonie wiele razy {
    Wywołaj select, aby sprawdzić, czy połączenia są gotowe;
    if (połączenie 1 jest gotowe) {
        Wywołaj recv, aby pobrać żądanie z połączenia 1;
        Przygotuj odpowiedź na żądanie;
        Wywołaj send, aby wysłać odpowiedź w połączeniu 1;
    } if (połączenie 2 jest gotowe) {
        Wywołaj recv, aby pobrać żądanie z połączenia 2;
        Przygotuj odpowiedź na żądanie;
        Wywołaj send, aby wysłać odpowiedź w połączeniu 2;
    }
}
```

## Podsumowanie

Programista może utworzyć aplikację sieciową, która działa w internecie, nie znając zasad funkcjonowania sieci ani mechanizmów odpowiedzialnych za przesyłanie danych między komputerami. Musi jednak otrzymać odpowiednio ogólne funkcje, które utworzą interfejs programistyczny (API). Zaprezentowana w tym Dodatku biblioteka API składa się zaledwie z siedmiu prostych funkcji, które wystarczają do opracowania aplikacji zdolnych do komunikowania się z oprogramowaniem komercyjnym.

{}

## ZADANIA

- A1.1. Przedstawione przykłady nie uwzględniają mechanizmów weryfikacji parametrów przekazywanych w wierszu polecenia. Zmień kod tak, aby sprawdzenie poprawności danych było wykonywane.
- A1.2. Usługa *echo* jest standardową usługą internetową o numerze aplikacji równym 7. Pobierz, skompiluj i uruchom program kliencki, aby sprawdzić, czy inne komputery w sieci mają uruchomioną tę usługę.
- A1.3. Zmień program serwera *echo* w taki sposób, aby nie kończył działania po obsłużeniu jednego żądania klienckiego. Podpowiedź: sprawdź, jak ten problem rozwiązano w kodzie serwera WWW.
- A1.4. Pobierz, skompiluj i przetestuj program czatu, uruchamiając go w dwóch komputerach.
- A1.5. Program czatu umożliwia dwóm użytkownikom naprzemienne wprowadzanie pojedynczych wierszy tekstu. Zmień jego działanie tak, aby każdy z użytkowników mógł wpisywać wiele wierszy tekstu w dowolnej chwili. Podpowiedź: wykorzystaj wątki.
- A1.6. Zmień program czatu tak, aby wraz z wiadomością wysyłał nazwę użytkownika. Zmień również program serwera, aby informował o nadawcy wiadomości.
- A1.7. Zmień program opisany we wcześniejszym zadaniu tak, aby wysyłał informację o nazwie użytkownika jedynie podczas pierwszego kontaktu z aplikacją zdalną. Program powinien zapamiętać tę nazwę i wyświetlać ją na początku każdego wiersza tekstu.
- A1.8. Opracuj program czatu, który umożliwi wymianę komunikatów między dowolną liczbą użytkowników, a także przyłączanie się do rozmowy oraz opuszczanie sesji w dowolnym momencie.
- A1.9. Użyj programu telnet do nawiązania połączenia z serwerem WWW. Wyślij żądanie GET i przeanalizuj uzyskaną odpowiedź.
- A1.10. Wypróbuj działanie klienta WWW w komunikacji z internetowym serwerem WWW. Wybierz dowolną nazwę serwera. Jako nazwę strony wpisz ciąg *index.html* lub *index.htm*, a jako numer aplikacji liczbę 80.
- A1.11. Dodaj kolejną „stronę” do kodu serwera WWW.
- A1.12. Zmień kod serwera WWW tak, aby pobierał treść każdej strony z pliku, a nie ze wstępnie zdefiniowanych stałych.
- A1.13. Rozbuduj kod opisany w poprzednim zadaniu tak, aby rozpoznawał nazwy o rozszerzeniu *gif* i wysyłał odpowiadające im pliki z nagłówkiem Content-type o wartości *image/gif*, a nie *text/html*.

- A1.14. Trudne zadanie: Napisz program kliencki i program serwerowy usługi transferu plików.
- A1.15. Trudne zadanie: Zaimplementuj mechanizm CGI zgodnie ze specyfikacją opublikowaną na stronie:  
<http://hoohoo.ncsa.uiuc.edu/cgi/>
- A1.16. Trudne zadanie: Rozbuduj program serwera WWW tak, aby mógł obsługiwać wiele połączeń równolegle. Podpowiedź: użyj funkcji `fork` i `pthread_create`.
- A1.17. Trudne zadanie: Napisz program kliencki, który połączy się z serwerem SMTP i wyśle wiadomość e-mail.



# **Skorowidz**

100BaseT, 282

10Base2, 279

10Base5, 278

10BaseT, 282

16-PSK, 198

2-PSK, 198

3-way handshake, 462

4-PSK, 198

## **A**

ABR, Available Bit Rate, 503

ACK, Acknowledgement, 453

ACL, Access Control List, 538

administrator sieci, 557

adres, 252

  docelowy, 389, 393

  emisji pojedynczej, 254

  IP, 61, 366, 379

  MAC, 366

  multiemisji, 254

  następnego skoku, 389

  ograniczonego rozgłaszenia, 377

  pętli zwrotnej, 377

  rozgłoszeniowy, 253, 254, 378

  sieci, 367

  URL, 83

  własny komputera, 377

  źródłowy, 393

adresacja

  CIDR, 376

  IPv6, 434

  w sieciach WAN, 329

adresowanie

  bezklasowe, 370

  hierarchiczne, 329

  klastrowe, 434

adresy

  IP o specjalnym przeznaczeniu, 375, 378

  MAC, 254

  nieroutowalne, 416

  prywatne, 416

  rozgłoszenia kierowanego, 376

  stacji, 375

ADSL, 225

  DMT, 226

  filtry, 227

  instalacja, 227

  łącza adaptacyjne, 225

  podkanał, 226

  podział pasma, 225

  przepustowość łączys, 226

agent, 567

algorytm

  cieknącego wiadra, 507

  CRC, 173

  CSMA/CA, 295

  CSMA/CD, 270, 280

  Dijkstry, 336, 337

  drzewa rozpinającego, 319

  karuzelowy, 212, 506

  karuzelowy deficytowy, 507

  karuzelowy ważony, 507

  klucza prywatnego, 539

  klucza publicznego, 539

  obliczania sumy kontrolnej, 172

  parzystości wierszy i kolumn, 171

  powolnego startu, 465

  RAC, 171

  statystyczny TDM, 215

  wektora odległości, 338

  wiadra z żetonami, 507

  wyboru tras, 332

ALOHA, 267

- B**
- alokacja
    - kanałów, 262
    - kanałów częstotliwościowych, 208
    - kanałów dynamiczna, 263
    - kanałów statyczna, 262
    - rodzaje protokołów, 263
    - subkanałów, 209
  - analizator ruchu NetFlow, 562
  - antena paraboliczna, 306
  - API, Application Programming Interface, 64, 583
  - aplikacja
    - czatu, 592
    - echo, 588
    - FTP, 494
    - kliencka, 58
    - serwerowa, 58
    - WWW, 597
  - aplikacje do zarządzania siecią, 562
  - aplikacje internetowe, 50
  - aplikacje sieciowe, 31
  - aproksymacja sygnału, 130
  - architektura
    - bezprzewodowej sieci LAN, 292
    - internetu, 358
    - przełącznika, 321
    - sieci WAN, 326
  - ARP, Address Resolution Protocol, 403
    - buforowanie, 406
    - enkapsulacja, 405
    - format komunikatu, 403
    - przetwarzanie, 406
  - ARPA, Advanced Research Projects Agency, 46
  - ARPANET, 46, 349
  - ARQ, Automatic Repeat reQuest, 165
    - komunikat potwierdzenia, 175
    - retransmisja wiadomości, 175
  - arytmetyka uzupełnień do jedności, 172
  - ASK, Amplitude Shift Keying, 195
  - asynchroniczna transmisja RS-232, 185
  - atak DoS, 533
  - atak man-in-the-middle, 534
  - ataki sieciowe, 531
  - ATM, 351
  - audio, 518
  - autentyczność wiadomości, 541
  - automatyczne powtarzanie żądania, 165
  - bajt, 185
  - baza MIB, 565
  - best-effort, 390
  - bezpieczeństwo
    - autoryzacja, 536
    - dostępność danych, 535
    - DPI, 546
    - filtr pakietów, 544
    - IDS, 545
    - kontrola, 536
    - poufność danych, 536
    - prywatność, 536
    - rejestrowanie zdarzeń, 536
    - skanowanie plików, 546
    - spójność danych, 535
    - uwierzytelnienie, 536
    - zapory sieciowe, 543
  - bezpieczeństwo sieci, 552
    - HTTPS, 553
    - IPSec, 553
    - PGP, 552
    - RADIUS, 553
    - SSH, 552
    - SSL, 552
    - TLS, 553
    - WEP, 553
  - BGP, Border Gateway Protocol, 476
    - cechy protokołu, 477
  - biblioteka API, 585, 587
  - bit, 129
    - bit LSB, 182
    - bit MSB, 182
    - bit parzystości, 167
    - bit startu, 183
    - bit stopu, 184
    - bitowa reprezentacja maski, 388
    - blok, 185
    - Bluetooth, 288, 299
    - błąd synchronizacji, 132
    - błędy powtórzeniowe, replay errors, 453
    - błędy transmisyjne, 163
      - automatyczne powtarzanie żądania, 165
      - błąd pojedynczego bitu, 165
      - interferencje, 164
      - kodowanie korekcyjne, 165
      - obsługa błędów, 165
      - tłumienie, 164

usunięcie, 165  
zbitka błędów, 165  
zniekształcenia, 164  
bod, baud, 129  
BOOTP, Bootstrap Protocol, 412  
BPSK, Binary Phase Shift Keying, 198  
brama, gateway, 521  
budowa przeglądarki, 88  
budowa przełącznika, 320  
budowa sieci LAN, 313  
buforowanie ARP, 406  
buforowanie danych, 103  
buforowanie stron, 87  
buforowanie treści (Akamai), 572  
bufory fluktuacji opóźnienia, 514

## C

CBR, Constant Bit Rate, 503  
CCITT, Consultative Committee for International Telephone and Telegraph, 349  
CCITT, Consultative Committee for International Telephone and Telegraph, 41  
CDDI, 348  
CDM, Code Division Multiplexing, 216  
CDMA, Code Division Multi-Access, 216, 263  
CDMA 2000, 305  
cechy transmisji radiowej, 308  
centrala sieci komórkowej, 301  
CIDR, Classless Interdomain Routing, 373  
cienki Ethernet, 279  
CMTS, Cable Modem Termination System, 231  
CNAME, 105  
Comer Douglas, 23  
COPS, Common Open Policy Services, 506  
CRC, Cyclic Redundancy Code, 173  
cechy kodu, 173  
implementacja sprzętowa algorytmu, 175  
wielomian generującym kod, 175  
CSMA/CA, 267, 271  
CSMA/CD, 267  
CSU/DSU, Channel Service Unit/Data Service Unit, 233  
cyfrowa linia abonencka, 224  
cyfrowa modulacja wielotonowa, 226  
cyfrowe obwody dzierżawione, 233  
cyfrowe obwody punkt-punkt, 232  
cyfrowy procesor sygnałowy, 308  
cyfrowy przekaz wideo, 575

cyfrowy sygnał informacyjny, 196  
cykliczny kod nadmiarowy, 173, 176  
czarna lista adresów URL, 547  
czas dzierżawy, 413  
czas oczekiwania na potwierdzenie, 461  
czas retransmisji, 460  
czas wstrzymania transmisji, 269  
częstotliwość próbkowania, 136  
częstotliwość radiowa, Radio Frequency, 151  
czujniki, 577

## D

datagram IP, 384  
fragmentowanie, 393  
nagłówek, 385  
przekazywanie datagramu, 387  
rejestrowanie fragmentów, 395  
datagram IPv6, 429, 432  
datagram UDP, 444  
DCF, Distributed Coordination Function, 294  
decyble (dB), 157  
dekoder kanałowy, 118  
dekoder źródłowy, 117  
demodulator, 118, 198  
demultiplexacja, 205, 252  
demultiplexacja FDM, 207  
demultiplexer, 118, 205  
deskryptor, 64  
deszyfrator, 118  
detekcja błędów, 170  
detekcja kolizji, 268  
DHCP, Dynamic Host Configuration Protocol, 412  
diagram konstelacji, 195, 197, 199  
DiffServ, Differentiated Services, 507  
DMT, Discrete Multi Tone, 226  
DNS, Domain Name System, 98  
aliasy nazw, 105  
buforowanie danych, 103  
CNAME, 105  
drzewa nazw, 102  
hierarchia, 101  
odpowiedzi, 103  
odwzorowywanie nazwy na adres, 104  
rodzaje wpisów, 104  
serwery główne, 101  
skróty, 106  
znaki narodowe, 106  
żądania, 103

docelowy adres IP, 544  
 DOCSIS, Data Over Cable System Interface Specification, 231  
 dokument  
   RFC 1889, 528  
   RFC 2663, 421  
   RFC 2766, 421  
   RFC 2916, 528  
   RFC 3216, 528  
 domeny administracyjne telefonii IP, 527  
 domeny najwyższego poziomu, 99  
 domeny rozgłoszeniowe, 321  
 dopasowanie o najdłuższym prefiksie, 389  
 dostarczanie datagramu, 390  
 dostawca treści, 575  
 dostawca usług, 34, 221  
 dostęp do internetu, 222  
   szerokopasmowy, 222  
   wąskopasmowy, 222  
 DPI, Deep Packet Inspection, 546  
 drzewa nazw DNS, 102  
 drzewo rozpinające, 318  
 DSL, Digital Subscriber Line, 224, 346  
 DSL lite, 227  
 DSP, Digital Signal Processors, 308  
 DSSS, 290  
 DST, Distributed Spanning Tree, 318  
 dupleks, 186  
 DVR, Distance Vector Routing, 335, 336  
 DWDM, Dense Wavelength Division Multiplexing, 210  
 dwustronna translacja NAT, Twice NAT, 419  
 dynamiczne aktualizacje informacji o routingu, 332  
 dziedziczenie, 73

## E

efektywna szybkość dostarczania danych, 494  
 EGP, Exterior Gateway Protocols, 474  
 EGPRS, Enhanced GPRS, 304  
 ekran, 143  
 ekranowanie, 145  
 element sieci, 560  
 enkapsulacja  
   ARP, 405  
   ICMP, 410  
   IP, 391  
   RTP, 516  
   UDP, 445

ENUM, 526  
 ethernet, 268, 348  
 ethernet skrątkowy, 280

## F

fala nośna, 191, 196  
 faza, 194  
 FDDI, 348  
 FDM, Frequency Division Multiplexing, 206  
 FDMA, 263  
 FEC, Forward Error Correction, 165  
 FHSS, 290  
 filtrowanie pakietów, 544  
 filtrowanie ramek, 316  
 firma  
   Cisco, 22, 501  
   Linksys, 420  
 fluktuacja opóźnienia, 496, 514  
 format  
   Berkeley, 378  
   komunikatu ARP, 403  
   ramki 802.11, 293, 294  
   ramki ethernetowej, 276  
   ramki IEEE 802.3, 277  
 forum WiMAX, 296  
 Fourier, 130  
 fragment, 394  
 fragmentowanie fragmentów, 396  
 Frame Relay, 350  
 FSK, Frequency Shift Keying, 195  
 FTP, File Transfer Protocol, 89  
   konto anonimowe  
     hasło guest, 91  
     nazwa anonymous, 91  
   sesja, 90  
   ustanawianie połączenia, 91

FTTB, 230  
 FTTC, 230  
 FTTH, 230  
 FTTP, 231  
 funkcja  
   accept, 70  
   appname\_to\_apnum, 583, 585  
   await\_contact, 583, 585  
   bind, 69  
   bram sygnalizacji, 522  
   bramy mediów, 522  
   close, 68

**funkcja**

cname\_to\_comp, 583, 586  
 connect, 68  
 gethostbyaddr, 73  
 gethostbyname, 73, 101  
 gethostname, 73  
 getsockopt, 73  
 gniazda, 66  
 interfejsu API gniazd, 65  
 kontrolera bram mediów, 522  
 listen, 70  
 make\_contact, 583, 585  
 modułu połączenia, 522  
 NetFlow, 501  
 obsługi kont, 522  
 read, 68  
 readln, 595  
 recv, 67, 583, 586  
 recvfrom, 72  
 recvln, 583, 586, 595  
 recvmsg, 72  
 routingu, 522  
 rozszyfrująca, 539  
 select, 603  
 send, 67, 583, 586  
 send\_eof, 583, 587  
 sendmsg, 72  
 sendto, 71  
 serwera aplikacji, 522  
 serwera mediów, 522  
 setsockopt, 73  
 skrótu, 537  
 socket, 66  
 sterowania usługami, 522  
 sygnalizacji w bramie dostępowej, 522  
 szyfrująca, 540  
 write, 68  
 współpracy z innymi sieciami, 522

**G**

GEO, Geostationary Earth Orbit, 154  
 geostacjonarne satelity komunikacyjne, 153  
 GET, 85  
 gesta multipleksacja z podziałem długości fali, 210  
 Gig-E, 283  
 globalny system komunikacji mobilnej, 304  
 gniazdo, 64  
 GNU Radio, 309

GPRS, General Packet Radio Service, 304

GPS, Global Positioning System, 307

graf, 333

graf OSPF, 481

graf z wagami przypisanymi do krawędzi, 338

granica stosowania adresów, 408

gruby Ethernet, 278

GSM, Global System for Mobile Communications, 304

**H**

harmonogramowanie ruchu, 507

HEAD, 85

HFC, Hybrid Fiber Coax, 229

hierarchia

DNS, 101

FDM, 209

synchronicznych systemów cyfrowych, 237

TDM, 213

HTML, HyperText Markup Language, 81

HTTP, HyperText Transfer Protocol, 81

HTTPS, 553

hub, 280

**I**

ICANN, Internet Corporation for Assigned Names and Numbers, 99, 370

ICMP, Internet Control Message Protocol, 409

enkapsulacja, 410

komunikaty, 409

konfiguracja, 411

identyfikacja pakietów, 252

identyfikacja serwerów, 61

identyfikator mostu, 319

identyfikatory standardów sieci LAN, 249

IDNA, Internationalizing Domain Names in Applications, 106

IDS, Intrusion Detection System, 545

IEEE, Institute for Electrical and Electronic Engineers, 247

IGMP, Internet Group Multicast Protocol, 484

IGP, Interior Gateway Protocols, 474

iloczyn logiczny maski i adresu docelowego, 388

iloczyn opóźnienia i przepustowości, 498

impuls świetlny, 148

indeks modulacji, 194

InfraRed, 300

instalacja ADSL, 227  
 instalacja modemu kablowego, 229  
 inteligentny interfejs, 320  
 interfejs  
   API, 581, 584  
   API gniazd, 64, 65  
   programistyczny, 583  
   programistyczny aplikacji, API, 64  
   programowania aplikacji, 583  
   przyłączeniowy, 278  
   sieciowy, 380  
 interferencje, 164  
 internet, 357  
 internetowe technologie QoS, 506  
 internetowy protokół grup multiemisji, 484  
 internetowy protokół komunikatów  
   sterujących (ICMP), 409  
 intranet, 557  
 IntServ, Integrated Services, 504  
 IPSec, 553  
 IPTV, 575  
 IPv4, Internet Protocol wersja 4, 365  
 IPv6, 578  
 IR, Infra Red, 150  
 ISDN, Integrated Services Digital Network, 224  
 IS-IS, Intermediate System to Intermediate  
   System, 482  
 ISM, Industrial, Scientific and Medical, 288  
 ISO, International Organization for  
   Standardization, 41  
 ISP, Internet Service Provider, 34, 221  
 ITAD, IP Telephone Administrative  
   Domains, 527  
 ITU, International Telecommunications  
   Union, 349  
 ITU-T, International Telecommunications  
   Union — Telecommunication  
   Standardization Sector, 41

**J**

jakość usługi (QoS), 501  
 jawny i niejawny rozmiar nagłówka Ipv6, 431  
 jednolita usługa, 356  
 jednostki obsługi danych, 233  
 jednostki obsługi kanału, 233  
 język znacznikowy, 82  
 jitter, 186, 496

**K**

kabel  
   miedziany, 149  
   prosty, 283  
   współosiowy, 145  
   z przeplotem, 283  
 kanał, 206  
 kanał fizyczny, 118  
 kanał komunikacyjny, 208  
 kanał ramkowania, 213  
 kanał w dół, downstream, 222  
 kanał w górę, upstream, 222  
 karta sieciowa, 278  
 kategorie parametrów QoS, 503  
 kategorie sieci, 247  
 kategorie skrętek, 146  
 kąt krytyczny, 147  
 klastry, 156, 302  
 klastry komórek, 303  
 klasy adresów IP, 367  
 klient i serwer, 60  
 klucz deszyfrujący, 539  
 klucz prywatny, 539  
 klucz publiczny, 539  
 klucz szyfrujący, 539  
 kluczowanie, 195  
 kluczowanie amplitudy, 196  
 kluczowanie częstotliwości, 196  
 kluczowanie fazy, 195  
 kod CRC, 173  
 kod RAC, 171  
 kod uwierzytelniający wiadomość, 537  
 koder kanałowy, 118  
 koder źródłowy, 117  
 kodowanie  
   liniowe, 132  
   bipolarne, 133  
   unipolarne, 133  
   wielopoziomowe, 133  
   Manchester, 134  
   różnicowe Manchester, 134  
 kodowanie kanałowe, 167  
 kodowanie korekcyjne, 165, 176  
 kodowanie wierszy i kolumn, 171  
 kody blokowe, 166, 176  
   bez pamięci, 166  
   nadmiarowość, 166  
   notacja (n,k), 168  
   pojedyncza kontrola parzystości, 167

kody splotowe, 166  
     z pamięcią, 166  
 kolizja, 268  
 kompresja bezstratna, 137  
 kompresja stratna, 137  
 komunikacja  
     bezpołączeniowa, 441  
     FTP, 89  
     laserowa, 150  
     P2P, 573  
     przezroczysta, 415  
     punkt-punkt, 159  
     radiowa, 151  
     satelitarna, 306  
     sieciowa, 50  
     UWB, 299  
     w paśmie ISM, 288  
 komunikat, 56  
     ARP, 403  
     DHCP, 414  
     DVR, 337  
     sterujący, 463  
     UDP, 444  
 koncentrator, hub, 280  
 konfiguracja systemu pocztowego, 95  
 konfiguracja zapory sieciowej, 545  
 konstelacja modulacji QAM, 201, 202  
 kontrola dostępu, 538  
 korekcja błędów, 170, 175  
 korekcja pojedynczego błędu, 171  
 koszt administracyjny, 475  
 kryptografia, 538  
 książka kodowa, 168  
 kwadraturowa modulacja amplitudy, 198  
 kwantowanie, 135

**L**

LAN, Local Area Network, 42, 247  
 laser, 150  
 LEO, Low Earth Orbit, 154  
 liczba komputerów, 370  
 liczba przeskóków, 475  
 liczba sieci, 370  
 licznik TTL, 410  
 lista kontroli dostępu, 538  
 lista masek podsieci, 373  
 LOS, Line-Of-Sight, 297  
 LSR, Link-State Routing, 335

**L**

łącza ADSL, 225  
 łącza satelitarne VSAT, 347  
 łącza typu trunk, 235  
 łącze abonenckie, 223

**M**

MAC, Media Access Control, 252  
 MAC, Message Authentication Code, 537  
 magistrala, 235  
 maksymalna jednostka transmisyjna, 393  
 MAN, Metropolitan Area Network, 247  
 martwa strefa, 293  
 maska adresu, 371  
 maska podsieci, 371  
 maszyna wirtualna, Virtual Machine, 573  
 MCU, Multipoint Control Unit, 521  
 mechanizm  
     ALOHA, 267  
     CSMA/CA, 270  
     CSMA/CD, 268  
     detekcji błędów, 166  
     dystrybucji kluczów, 542  
     nadziewania bajtami, 256  
     NetFlow, 500  
     połączeniowy, 58  
     start-stop, 456  
 medium transmisyjne, 141  
     opóźnienie propagacyjne, 157  
     parametry, 157  
     pojemność kanału, 157  
     wybór medium, 156  
 menedżer, 567  
 MEO, Medium Earth Orbit, 154  
 metryka routingu, 475  
 MGCP, Media Gateway Control Protocol, 519  
 MIB, Management Information Base, 565  
 MIME, Multi-purpose Internet Mail  
     Extensions, 97  
 MIMO, Multiple-Input Multiple-Output, 309  
 MISTP, Multiple Instance Spanning Tree  
     Protocol, 319  
 mobile IP, 575  
 mobilność, 575  
 mobilny WiMAX, 296  
 model  
     FCAPS, 558  
     IEEE 802, 275  
     klient-serwer, 58, 582

- model  
 OSI, 41  
 powiązań serwerowych, 101  
 transmisji danych, 116  
 warstwowy, 37
- modem, 198  
 czołowy, 231  
 kablowy, 228, 346  
 instalacja, 229  
 przepustowość, 228
- końcowy, 231  
 optyczny, 200, 314  
 radiowy, 200  
 telefoniczny, 200  
 V.32, 201  
 V.32bis, 201  
 wewnętrzny, 201  
 zewnętrzny, 201
- modulacja  
 16QAM, 199  
 amplitudy, 192  
 analogowa, 192  
 częstotliwości, 193  
 delta, 136  
 kumulowanie błędów, 136  
 fazy, 194  
 impulsowo-kodowa (PCM), 135, 518  
 QAM, 198  
 QAM w telefonii, 201
- modulator, 118, 198  
 moduł interfejsu sieciowego, 234  
 monitorowanie sieci, 500, 558  
 most, 315  
 most adaptacyjny, 316  
 most uczący się, 316
- MPLS, Multiprotocol Label Switching, 351, 507
- MSC, Mobile Switching Center, 301
- MSTP, Multiple Spanning Tree Protocol, 319
- MTU, Maximum Transmission Unit, 393
- multicast, 253
- multiemisja, 576
- multiemisja IP, 483
- multimedia, 49, 513
- multimedia czasu rzeczywistego, 513
- multipleksacja, 205  
 kodowa, 216  
 odwrotna, 216  
 przestrzenna, 309  
 statystyczna, 215  
 z podziałem czasu, 211
- z podziałem częstotliwości, 206  
 z podziałem długości fali, 210
- multiplekser, 118, 205  
 multiplekser add/drop, 346
- N**
- naciąganie, 532  
 nadajnik, 192  
 nadpróbkowanie, 136  
 nadziewanie bajtami, 256, 257  
 nadziewanie bitami, 233, 256  
 nadziewanie znakami, 256  
 nadzorca, gatekeeper, 521  
 nagłówek, 40  
 datagramu IP, 386  
 odpowiedzi, 86  
 protokołu IPv6, 429
- najbardziej znaczący bit, MSB, 182  
 najmniej znaczący bit, LSB, 182  
 nakładanie obszarów, 293  
 NAPT, Network Address and Port Translation, 418
- narzędzia do zarządzania siecią, 561
- narzut protokołów, 494
- narzut transmisyjny, 170
- narzut transmisyjny i fragmentacja, 552
- następny skok, next hop, 330
- NAT, Network Address Translation, 91, 415  
 dostęp do serwerów, 419  
 działanie usługi, 416  
 forma podstawowa, 417  
 oprogramowanie, 420  
 tablica translacji, 418  
 warstwa transportowa, 418
- NetFlow, 501
- NIC, Network Interface Card, 252
- nieciągłe wartości sygnału, 194
- niezależna obsługa ruchu, 327
- niezależność od źródła, 332
- niezawodny transport danych, 450
- NLOS, Non-Line-Of-Sight, 297
- notacja (n,k), 168
- notacja ASN.1, 564
- notacja CIDR, 373, 374
- notacja dziesiętna z kropkami, 368, 369
- numer portu, 61, 444
- numeracja telefoniczna, 525
- numerowanie, 452
- Nyquist, 137

**O**

obsługa błędów, 165  
obszary OSPF, 482  
obwiednia, 192  
obwody przełączane, 244  
obwody trwałe, 244  
obwody wirtualne, 244  
OC, Optical Carrier, 235  
odległość do celu, 336  
odległość Hamminga, 168  
odległość Hamminga minimalna, 169  
odmowa obsługi, 532  
odpowiedź  
    DHCP, 412  
    DNS, 103  
    echa, 421  
    HTTP, 86  
odpytywanie, 264  
odtwarzanie datagramu z fragmentów, 394  
odwołania do gniazd, 66  
odwołania peer-to-peer, 63  
odwzorowanie adresów, 401, 402  
odwzorowanie nazw, 101  
OFDM, 290  
ogólna usługa pakietowej transmisji radiowej, 304  
okablowanie, 281  
okno przesuwne, 454, 455  
okno zerowe, 461  
oktet, 369  
operacja przekazania, 328  
operacja zapisu, 328  
opóźnienie, 492, 551  
    0,2 s, 155  
    dostępu do medium, 492, 493  
    kolejkowania, 492, 493  
    propagacyjne, 492  
    przełączania, 492, 493  
    serwera, 492, 493  
    transmisji, 460  
oprogramowanie routingu, 332  
orbita geostacjonarna, 154  
organ zarządzający kluczami, 542  
organizacja  
    IEEE, 247  
    IETF, 106, 436, 518  
    ISO, 41  
    ITU, 41, 518  
    nadzorcza, 473  
    normalizacyjna, 41

OSI, Open System Interconnection, 41  
OSPF, Open Shortest Path First, 479, 480  
    cechy protokołu, 480  
    graf, 481  
    obszary, 482  
otwarty protokół wyznaczania najkrótszych  
    tras (OSPF), 479, 480  
OUI, Organizationally Unique ID, 253

**P**

P2P, peer-to-peer, 574  
pakiet, 245  
pakiet RIP, 479  
PAN, Personal Area Network, 288  
parametr  
    DIFS, 295  
    QoS, 502  
    SIFS, 295  
pasma ochronne, 207  
pasmo danych, 201  
pasmo głosowe, 201  
pasmo ISM, 288  
PCF, Point Coordination Function, 294  
PCM, Pulse Code Modulation, 135  
peer-to-peer (P2P), 64, 574  
pętla routingu, 340  
pętla zwrotna, 233  
PGP, 552  
Phishing, 532  
plik  
    chatclient.c, 595  
    chatserver.c, 593  
    echoclient.c, 590  
    echoserver.c, 589  
    webcontent.c, 597  
    webserver.c, 599  
poczta elektroniczna, 92  
    algorytm, 92  
    aplikacja interfejsu, 93  
    dostęp do poczty, 95  
    IMAP, 96  
    kolejka poczty wychodzącej, 93  
    MIME, 97  
    POP3, 96  
    protoły, 93  
    RFC2822, 97  
    skrzynka pocztowa, 93  
    SMTP, 93

- podczerwień, IR, Infra Red, 150, 288
- podpis cyfrowy, 540
  - autentyczność wiadomości, 541
    - zaufany nadawca, 541
- podpróbkowanie, 136
- podsieć, 370
- podwarstwa
  - LLC, 248
  - MAC, 248, 261
    - sterowania dostępem do medium, 248
    - sterowania połączeniem logicznym, 248
- podział na klasy, 504
- podział protokołów, 262
- podział sieci bezprzewodowych, 287
- podział warstwy 2, 248
- pojedyncza kontrola parzystości, 167
- pojemność, 494
- pojemność kanału, 158
- pojemność warstwy sprzętowej, 494
- pole elektromagnetyczne, 144
- polecenie ping, 105
- polityka bezpieczeństwa, 535, 537
- połączenia
  - danych, 90
  - optyczne (OC), 346
  - routera bezprzewodowego, 420
    - sterujące, 90
    - wirtualne, 451
- połączenie, 57
  - dwóch modemów, 199
  - dwóch sieci fizycznych, 357
  - klient-serwer, 60
- poniar
  - fluktuacji opóźnienia, 499
  - opóźnienia, 499
  - pasywny, 500
  - przepustowości, 499
  - wydajności sieci, 499
- port, 61
  - docelowy, 466
  - źródłowy, 466
- POST, 85
- poszukiwanie MTU trasy, 433
- pośrednik DHCP, 415
- POTS, 225
- potwierdzenie, 453
- powiadomienia ICMP, 421
- półtupleks, 187
- prawo Keplera, 153
- prefiks, 434
- prefiks bezklasowy, 372
- prefiks klasy C, 372
- prefiks sieci, 388
- problem dystrybucji kluczy, 542
- problem ostatniej mili, 231
- problem ukrytej stacji, 270
- procesory sieciowe, 578
- program
  - dostarczania poczty, 93
  - Wireshark, 22, 561
    - zróżnicowanych usług, 507
- programowanie sieciowe, 31, 583
- projektowanie protokołu, 458
- promieniowanie elektromagnetyczne, 143
- propagacja fal elektromagnetycznych, 152
- propagacja sygnału, 152
- prosty protokół zarządzania siecią, 563
- protokoły
  - alokacji kanałów, 263
  - dostępu do poczty, 96
  - dostępu swobodnego, 266
    - ALOHA, 267
    - CSMA/CA, 270
    - CSMA/CD, 268
  - internetowe, 361
  - multiemisji, 486
    - CBT, 486
    - DVMRP, 486
    - MOSPF, 486
    - PIM-DM, 486
    - PIM-SM, 486
  - routingu wewnętrznego, 474
  - routingu zewnętrznego, 474
  - sterowania dostępem, 264
    - odpytywanie, 264
    - przekazywanie znacznika, 266
    - rezerwacja, 265
  - strumieniowania, 496
  - TCP/IP, 362 *Patrz także stos protokołów TCP/IP*
  - transportowe, 439, 452
  - WWW, 81
- protokół
  - ALOHA, 267
  - ARP, 391, 403
  - BGP, bram granicznych, 476
  - CBT, 486
  - CDMA, 263, 264

- CSMA/CA, 267, 272  
CSMA/CD, 267  
datagramów użytkownika, UDP, 419, 440  
DHCP, dynamicznej konfiguracji stacji, 411–414  
format komunikatu, 414  
dostarczania poczty, 93  
drzewa rozpinającego, 318  
DVMRP, 486  
EGP, 474  
FDMA, 263  
FTP, 89  
H.323, 519, 521  
brama, 521  
cechy protokołu, 523  
nadzorca, 521  
terminal, 521  
warstwy, 524  
HTTP, 84, 108  
format nagłówka odpowiedzi, 86  
kody statusowe, 86  
żądania HTTP, 85  
IGMP, 484  
IGP, 474, 476, 479  
IMAP, 96  
informowania o politykach, 506  
informowania o trasach (RIP), 478  
inicjowania sesji, 519  
internetowy, 494  
internetowy IPv4, 425  
IPv6, 428  
adresacja, 434  
datagram, 431  
format datagramu, 429  
format nagłówka, 428, 429  
fragmentacja, 431  
MTU trasy, 431  
nagłówki rozszerzające, 428  
obsługa ruchu, 429  
rodzaje adresów, 435  
rozmiar adresu, 428  
rozmiar nagłówka, 431  
roszzerzalność protokołu, 429  
zapis adresów, 435  
komunikacyjny, 36  
MOSPF, 486  
odwrotnego odwzorowania adresów, 411  
odwzorowania adresu (ARP), 391, 403  
PIM-DM, 486  
PIM-SM, 486  
POP, 96  
rezerwacji zasobów, 506  
RIP, 478  
routingu telefonicznego, 527  
routingu wewnętrznego (IGP), 474, 476, 479  
routingu zewnętrznego (EGP), 474  
RSVP, 506  
RTP, 515, 518  
sieciowy, 36, 494  
SIP, 520  
cechy protokołu, 524  
metody protokołu, 525  
moduł użytkownika, 521  
przebieg sesji, 525  
serwer lokalizacji, 521  
serwer pośredniczący, 521  
serwer rejestrujący, 521  
serwery przekierowań, 521  
SMTP, 93, 109  
sterowania bramami mediów, 519  
sterowania transmisją, TCP, 440, 450  
STP, Spanning Tree Protocol, 318  
systemów pośrednich (IS-IS), 482  
TCP, Patrz TCP, 440, 450  
TDMA, 263, 264  
transferu plików, Patrz FTP  
transportowy, 494  
transportowy czasu rzeczywistego (RTP), 515, 518  
UDP, Patrz UDP  
uruchomieniowy, 412  
warstwy aplikacji, 79, 494  
reprezentacja danych, 80  
transfer danych, 80  
wielu drzew rozpinających, 319  
próbkowanie, 135  
przeciążenie, congestion, 456, 457  
przejmowanie pakietów, 534  
przekazywanie danych między warstwami, 39  
przekazywanie znacznika, 266

przełącznik VLAN, 321  
 przełącznik warstwy 2, 319  
 przepływność strumieni danych, 454  
 przepustowość, 494, 497, 552  
 przepustowość systemu, 495  
 przestrzeń adresowa, 369  
 przesunięcie fazy, 195  
 przesyłanie datagramów, 485  
     konfiguracja i tunelowanie, 485  
     wyszukiwanie w rdzeniu, 485  
     zalej i odetnij, 485  
 przetwarzanie pakietów, 254  
 przetwarzanie żądań, 62  
 przydział adresów IP, 379  
 pseudonagłówek, 445  
 PSTN, Public Switched Telephone Network, 518  
 publiczne sieci telefoniczne, 518  
 punkt demarkacyjny, 234  
 punkt dostępowy, 292  
 punkt końcowy, 440  
 PUT, 85  
 PVST, Per-VLAN Spanning Tree, 319  
 PVST+, 319

## Q

QAM, Quadrature Amplitude Modulation, 198  
 QoS, Quality of Service, 502  
     ABR, 503  
     CBR, 503  
     ogólna specyfikacja, 502, 504  
     przetwarzanie pakietów, 505  
     szczegółowa specyfikacja, 502  
     technologie internetowe, 506  
     UBR, 503  
     VBR, 503

## R

RAC, Row And Column, 170  
 RADIUS, 553  
 ramka, 185  
     nagłówek, 255  
     pole danych, 255  
     znak EOT, 256  
     znak SOH, 256  
 ramka ethernetowa, 276  
 ramka SONET, 237  
 ramkowanie, 185, 212, 255

RARP, Reverse Address Resolution Protocol, 411  
 rdzeń, 232  
 reasemblacja, 394  
 regenerator, 315  
 reguły filtrowania, 546  
 rejestracja domen, 99  
 replikacja, 101  
 reprezentacja bitu, 133  
 retransmisja, 453  
 retransmisja pakietu, 268, 459  
 rezerwacja, 265  
 rezerwacja zasobów sieciowych, 501  
 RF, Radio Frequency, 151  
 RFC, Request for Comments, 97  
 RFID, Radio Frequency Identification, 300  
 RIP, Routing Information Protocol, 478  
     cechy protokołu, 478  
     format pakietu, 479  
 RJ45, 283  
 rodzaje  
     adresów IPv6, 435  
      błędów, 165  
     interakcji UDP, 443  
     modulowania fali nośnej, 192  
     multipleksacji, 206  
     okablowania, 143  
     opóźnienie, 492  
     protokołów alokacji kanałów, 263  
     protokołów sterowania dostępem, 264  
     przesyłanych danych, 49  
     transmisji, 186  
     włókien optycznych, 148  
 router, 357  
 router bezprzewodowy, 420  
 routing, 332, 388, 472  
     koszt administracyjny, 475  
     liczba przeskoków, 475  
     metryka routingu, 475  
 routing dynamiczny, 334, 469, 471  
 routing LSR, 335  
 routing na bazie informacji o stanie łączy, 335  
 routing SPF, 335  
 routing statyczny, 334, 469, 470  
 routing w multiemisji, 483  
 routing z wykorzystaniem wektorów  
     odległości, 335  
 rozgłasianie na podstawie tras powrotnych, 485  
 rozgłoszenie okna, 461  
 rozgłoszenie w formacie Berkeley, 378

- rozkładanie obciążenia serwerów WWW, 572  
rozłączanie połączenia TCP, 464  
rozpraszania widma, 209, 290  
    DSSS, 290  
    FHSS, 290  
    OFDM, 290  
rozproszone centra danych, 574  
rozproszone drzewo rozpinające, 318  
rozproszony atak DoS, 533  
rozszerzony GPRS, 304  
rozwój internetu, 47  
różnicowe kodowanie Manchester, 134  
RPB, Reverse Path Broadcasting, 485  
RS-232, 183  
RSVP, Resource ReSerVation Protocol, 506  
RTP, Real-time Transport Protocol, 515, 518  
    enkapsulacja, 516  
    nagłówek, 515
- S**
- satelity GPS, 307  
SDH, Synchronous Digital Hierarchy, 237  
segment TCP, 465  
serwer  
    DHCP, 415  
    DNS, 101, 109  
    FTP, 100  
    lokalizacji, location server, 521  
    pocztowy, 93, 95  
    pośredniczący, proxy server, 521  
    przekierowań, redirect server, 521  
    rejestrujący, registrar server, 521  
    WWW, 100  
serwery główne, root servers, 101  
sesja FTP, 90  
sesja SMTP, 94  
sieci  
    ad hoc, 578  
    bezprzewodowe, 575  
    domowe, 420  
    energetyczne (PLC), 347  
    izochroniczne, 496  
    korporacyjne, 35  
    lokalne, 247  
    małych biur, 35  
    małych przedsiębiorstw, 35  
    metropolitarne, 247  
    odbiorców prywatnych, 35  
    osobiste, PAN, 288  
    Bluetooth, 288  
    pasmo ISM, 288  
    Podczerwień, 288  
    pakietowe, 246  
    punkt-punkt, 250  
    rozległe, 247, 325  
    społecznościowe, 575  
    wielodostępne, 250  
sieć  
    ALOHA.net, 267  
    cyfrowa z integracją usług (ISDN), 352  
    Ethernet, 275  
    HFC, 229  
    internetowa, 33  
    komórkowa, 301  
    LAN, 247  
    MAN, 247  
    optyczna (SONET/SDH), 346  
    połączzeń, fabric, 320  
    prywatna, 35  
    PSTN, 525  
    publiczna, 34  
    w biznesie, 577  
    WAN, 247  
        adresacja, 329  
        cele sieci, 328  
        graf sieci, 333  
        tradycyjna architektura, 326  
    wirtualna, 359  
    z funkcją NAT, 416  
    z przełączaniem obwodów, 244  
siedmiowarstwowy model OSI, 40, 41  
simplex, 186  
SIP, Session Initiation Protocol, 519  
skalowalne usługi internetowe, 571  
skanowanie plików, 546  
skanowanie portów, 546  
skrątka, 280  
skrątka ekranowana, 145  
skrątka miedziana, 143  
słowa danych, 168  
słowa kodowe, 168  
SMB, Small-To-Medium Business, 35  
SMDS, 350  
SMTP, Simple Mail Transfer Protocol, 93  
SNMP, Simple Network Management  
    Protocol, 563  
SOHO, Small Office/Home Office, 35

- SONET, Synchronous Optical NETwork, 237  
SPC, Single Parity Check, 167  
specyfikacja DECNET V, 482  
specyfikacje, 36  
spektrum, 151  
społność sieci, 340  
SSH, 552  
SSL, 552  
stacje sieciowe, 362  
stacje ukryte, 272  
Stał WiMAX, 296  
standardy  
    ASCII, 256  
    ATP, 503  
    DIX, 268  
    DS, 235  
    E.164, 525  
    HTML, 81  
    IEEE, 289  
    IEEE 802.15, 299  
    IEEE 802.1q-2003, 319  
    IEEE 802.11, 291  
    IEEE 802.16, 296  
    IETF, 501  
    IP, 380  
    kodowania Base64, 97  
    Megaco (H.248), 519  
    MIME, 94, 98  
    RFC2822, 97  
    SNMP, 564  
    SONET, 238  
    STS, 235  
    WiMAX, 296  
    Zigbee, 299  
standardy  
    adresowania, 252  
    bezprzewodowych sieci LAN, 291  
    bezprzewodowych sieci MAN, 296  
    komunikacji bezprzewodowej, 289  
    komunikacji telefonicznej, 518  
    łącze cyfrowych, 234  
    łącze optycznych, 235  
    sieci PAN, 298  
    usługi WWW, 81  
    zapisu wiadomości e-mail, 97  
statystyczna multipleksacja, 215  
sterowanie przepływem, 454  
stopień wykorzystania sieci, 497  
stos, 37  
stos protokołów, 37  
stos protokołów sieciowych, 247  
stos protokołów TCP/IP, 33, 42, 361–363  
STP, Spanning Tree Protocol, 318  
strumień, 56, 503  
STS, Synchronous Transport Signal, 235  
sufiks, 434  
sufiks C, 236  
suma kontrolna, 171  
suma kontrolna UDP, 445  
sygnalizacja, 518  
sygnał analogowy, 122  
    szerokość pasma, 127  
sygnał cyfrowy, 122, 127  
    poziomy napięć, 128  
    szerokość pasma, 131  
sygnał nieokresowy, 122  
sygnał okresowy, 122  
sygnał sinusoidalny, 123  
    amplituda, 123  
    częstotliwość, 123  
    długość fali, 123  
    faza, 123  
sygnał zespłonny, 124  
sygnały elementarne, 125  
synchroniczna sieć optyczna, 236  
synchroniczne sygnały transportowe, 235  
synchroniczne zwielokrotnienie TDM, 211  
synchronizacja nadajnika z odbiornikiem, 131, 184  
systemy  
    analizy treści, 546  
    autonomiczny, Autonomous System, 473  
    CATV, 228  
    EDGE, 304  
    EDGE Evolution, 304  
    IDS, 546  
    ISC, 522  
        AGS-F, 522  
        AS-F, 522  
        CA-F, 522  
        IW-F, 522  
        MGC-F, 522  
        MG-F, 522  
        MS-F, 522  
        R-F, 522  
        SC-F, 522  
        SG-F, 522  
    komunikacyjny, 116

- komórkowy, 300  
nazw domenowych, 98  
priorytetowania, 502  
PSTN, 521  
satelitarny, 300  
sygnalizacji 7, 518  
telefoniczny, 212  
wykrywania włamań, 545  
zintegrowanych usług, 504  
szczegółowa inspekcja pakietów, 546  
szczelina nadawcza, 295  
szerokopasmowa technika CDMA, 305  
szerokość pasma, 127, 131, 495  
szum, 142, 157, 198  
szybki protokół drzewa rozpinającego, 319  
szybkość dostarczania danych, 494  
szybkość transmisji, 495  
szifrator, 118  
szifrowanie, 538  
    klucz deszyfrujący, 538, 539  
    klucz szyfrujący, 538, 539  
    szfyrogram, 538  
    tekst jawnym, 538  
szifrowanie pola danych, 550
- T**
- tablica przekazywania, forwarding table, 330  
tablica routingu, 330, 387  
tablica translacji, 418  
TCP, Transmission Control Protocol, 440, 450  
    adaptacyjne retransmisje, 460  
    cechy protokołu, 450  
    format segmentu, 465  
    obsługa ultraconnych pakietów, 458  
    okno, 461  
    opóźnienie transmisji w obie strony, 460  
    rozgłoszenie okna, 461  
    segment FIN, 463  
    segment SYN, 463  
    sterowanie przepływem, 461  
    trójetapowe porozumienie, 462  
TDM, Time Division Multiplexing, 211  
    hierarchia, 213  
    ramkowanie, 212  
    systemy telefoniczne, 212  
    wady systemu, 214  
zwielokrotnienie statyczne, 215  
zwielokrotnienie synchroniczne, 211
- TDMA, 263  
technika dzielenego horyzontu, 340  
technika wieloprotokołowego przełączania etykiet, 507  
techniki ataków, 533  
    DoS i DDoS, 533  
    fałszowanie adresu, 533  
    fałszowanie nazwy, 533  
    łamanie kluczy, 533  
    podслушаńwanie, 533  
    powtarzanie pakietów, 533  
    przejmowanie pakietów, 533  
    przepełnianie bufora, 533  
    skanowanie portów, 533  
    zalewanie pakietami SYN, 533  
techniki multimedialne w sieciach Wi-Fi, 290  
techniki przesyłania pakietów, 484  
techniki unikania przeciążeń, 456  
technologia  
    ADSL, 225  
    dostępu do internetu, 221  
    DSL, 224  
    EVDO, 305  
    EVDV, 305  
    HSDPA, 305  
    RFID, 300, 577  
    VSAT, 306, 307  
    WiMAX, 297, 298
- technologie  
    bezprzewodowych sieci WAN, 300  
    dostępu bezprzewodowego, 231  
    komórkowe, 303  
    komórkowe drugiej generacji, 305  
    komórkowe trzeciej generacji, 305  
    łączy dostępowych, 345  
    rdzeniowe, 232  
    sieci LAN, 347  
    sieci PAN, 298  
    sieci WAN, 349  
    sieciowe, 33  
    szerokopasmowe, 223  
    światłowodowe, 230  
        FTTB, 230  
        FTTC, 230  
        FTTH, 230  
        FTTP, 231  
    wąskopasmowe, 223  
telefonia IP, 517  
    aparat telefoniczny, 519  
    brama mediów, 520

- telefonia**
- brama sygnalizacji, 520
  - komponenty systemu, 519
  - kontroler bram mediów, 520
  - lokalizacja użytkowników, 525
  - PCM, 518
  - połączenia między komponentami, 520
  - protokół RTP, 518
  - zestawienie protokołów, 523
- telewizja kablowa**, 228
- TLD, Top-level domain**, 99
- TLS**, 553
- łumienie**, 164
- Token Ring**, 347
- topologia**
- fizyczna Ethernetu, 281
  - gwiazdy, 251
  - logiczna Ethernetu, 281
  - magistrali, 251
  - pierścienia, 251
  - siatki, 251
- topologie sieci LAN**, 250
- transfer plików**, 89
- translacja adresów sieciowych (NAT)**, 415
- translacja adresów sieciowych i portów**, 418
- translacja NAPT**, 419
- transmisja**
- asynchroniczna, 182, 183
  - bezprzewodowa, 141
  - danych, 114, 118
  - danych, data communication, 31
  - datagramu, 391
  - izochroniczna, 182, 186
  - naziemna, 153
  - przewodowa, 141
  - radiowa, 308
  - RF, 151
  - rodzaje energii, 142
  - równoległa, 180
  - spoza Ziemi, 153
  - synchroniczna, 182, 184
  - szeregową, 179, 181
  - w podczerwieni, 150
  - znaków, 183
- transport danych**, 476
- transport komunikatów**, 57
- transport strumieni**, 56
- trasa domyślana**, 333, 470
- trendy**, 50
- TRIP, Telephone Routing over IP**, 527
- trójetapowe porozumienie**, 462
- tryb transmisji**, 179
- tunel MPLS**, 560
- tunelowanie**, 550
- tunelowanie IP-w-IP**, 550
- tunelowanie IP-w-TCP**, 551
- twierdzenie Nyquista**, 136
- twierdzenie Shannona**, 157, 194
- U**
- UART**, 181
- UBR, Unspecified Bit Rate**, 503
- UDP, User Datagram Protocol**, 419, 440
- cechy protokołu, 440
  - enkapsulacja komunikatu, 445
  - format datagramu, 444
  - identyfikacja punktów końcowych, 444
  - przebieg komunikacji, 442
  - przetwarzanie komunikatów, 441
  - pseudonagłówek, 445
  - rodzaje interakcji, 443
  - suma kontrolna, 445
- ujednolicone identyfikatory zasobów**, 526
- ujednolicony format adresowania zasobów**, 83
- UMTS**, 305
- unicast**, 253
- Universal Software Radio Peripheral**, 309
- uniwersalny pakiet wirtualny**, 384
- URI, Uniform Resource Identifier**, 526
- URL, Uniform Resource Locator**, 81
- Urząd Komunikacji Elektronicznej**, 207
- urządzenie DCE**, 187
- urządzenie DTE**, 187
- urządzenie terminalowe**, 187
- urządzenie transmisji danych**, 187
- USART**, 181
- usługa**
- CLNS, 483
  - DAYTIME, 80
  - FTP, 89
  - MMS, 304
  - NAPT, 419
  - NAT, 416
  - SMS, 304
  - VBR, 503
  - PBR, 503
  - PBS, 504
  - SBR, 503

SBS, 504  
 WAP, 304  
 WWW, 81  
**usługi**  
 bezpołączeniowe, 383  
 połączeniowe, 383  
 sieciowe, 31  
 wielopoziomowe, 502  
 ustanawianie połączenia TCP, 463  
 usuwanie gniazda, 73  
 utrata danych, 532  
 utrata kontroli, 532  
 utrata pakietu, 395, 458  
 uwierzytelnianie, 540

**V**

VBR, Variable Bit Rate, 503  
 Voice over IP (VoIP), 517  
**VPN, Virtual Private Network, 547**  
 fragmentacja, 552  
 internet, 548  
 narzut transmisyjny, 552  
 niezależne urządzenia, 549  
 obwody dzierżawione, 548  
 opóźnienie, 551  
 oprogramowanie, 549  
 praca zdalna, 549  
 przepustowość, 552  
 Szyfrowanie pola danych, 550  
 Tunelowanie IP-w-IP, 550  
 Tunelowanie IP-w-TCP, 551  
**VSAT, Very Small Aperture Terminal, 306**

**W**

**WAN, Wide Area Network, 42**  
**warstwa, 37**  
 warstwa 1 — fizyczna, 38  
 warstwa 2 — interfejsu sieciowego, 38  
 warstwa 3 — internetowa, 38  
 warstwa 4 — transportowa, 38  
 warstwa 5 — aplikacji, 38  
 warstwy stosu TCP/IP, 361  
     warstwa 3. — internetowa, 361  
     warstwa 4. — transportowa, 361  
 wartość MTU, 394  
 wątek, 62  
 wątek główny, 74  
 wątek potomny, 74

WCDMA, Wideband CDMA, 305  
 WDM, Wavelength Division Multiplexing, 210  
 wektor odległości, 336  
 WEP, 553  
 węzeł bezprzewodowy, 292  
 wielodostęp, 270  
 wielodostęp kodowy, 216  
 wielozadaniowe rozszerzenia poczty  
     internetowej, 97  
 Wi-Fi, 289, 346  
 WiMAX, 296, 346  
 wirtualizacja serwerów, 573  
 wirtualna sieć prywatna (VPN), 547  
 wirtualne pakiety, 384  
 włókna światłowodowe, 146, 149  
 wskazanie komputera, 61  
 wskazanie usługi, 61  
 współczynnik kodu, 170  
 współdzielenie medium transmisyjnego, 206  
 współdzielenie zasobów, 45  
 wydajność sieci, 499  
     asymetryczne trasy, 499  
     technika pomiarowa, 499  
     warunki transmisji, 499  
     zbitki danych, 499  
 wykres sygnału w funkcji czasu, 126  
 wykres sygnału w funkcji częstotliwości, 126  
 wykrywanie nośnej, 268  
 wykrywanie stacji, 484  
 wyłudzenia, 532  
 wysyłanie bitów, 182  
 wyznaczanie tras w sieci WAN, 332

**X**

X.25, 349  
**XML, Extensible Markup Language, 108, 574**

**Z**

zabezpieczenie sieci przed przeciążeniem, 546  
 zabronione witryny, 547  
 zagnieździenie nagłówków, 40  
 zagrożenia internetowe, 532  
     naciąganie, 532  
     odmowa obsługi, 532  
     Phishing, 532  
     utrata danych, 532  
     utrata kontroli, 532  
     wyłudzenia, 532

- zalewanie pakietami SYN, 534, 546  
zapis adresów IPv6, 435  
zapory sieciowe, 543  
zarezerwowane przedziały częstotliwościowe, 288  
zarządzanie elementem, 560  
zarządzanie intranetem, 557  
zarządzanie siecią, 558, 561  
    agent, 567  
    aplikacje, 562  
    menedżer, 567  
    narzędzia, 561  
    protokół SNMP, 563  
zasada działania protokołów, 39  
zasada zapisz i przekaź, 328  
zasady adresowania IP, 379  
zasady komunikacji, 582  
zasięg sieci  
    LAN, 325  
    MAN, 325  
    PAN, 325  
    WAN, 325  
zasoby adresowe, 427  
zator, 63, 457  
zaufany nadawca, 541  
zdolność do współdziałania, 36  
zegar odtwarzania, reassembly timer, 396
- Zigbee, 299  
zliczanie pakietów, 501  
zliczanie referencji, 73  
zmienne MIB, 565  
zmienne tablicowe, 566  
znacznik czasu, 308, 514  
znacznik końca pliku, 80, 582  
znacznik, tags, 82  
znak EOT, 256  
znak SOH, 256  
znaki narodowe, 106  
znieksztalcenia, 164

## Ź

- źródła informacji, 116, 121  
źródłowy adres IP, 544

## Ż

- żądania współbieżnie, 62  
żądanie DHCP, 412  
żądanie DNS, 103  
żądanie echa, 421  
żądanie HTTP, 84  
    GET, 85  
    HEAD, 85  
    POST, 85  
    PUT, 85



Autor bestsellerów i jeden z największych autorytetów w dziedzinie sieci komputerowych — Douglas E. Comer — przedstawia wszechstronny i kompletny przegląd technologii internetowych, które umożliwiają korzystanie z różnorodnych aplikacji, od przeglądarek internetowych, przez systemy telefonii IP, po programy multimedialne. Piąte wydanie obejmuje wiele nowych zagadnień, od protokołów komunikacji bezprzewodowej po problematykę wydajności sieci.

W tej książce znajdziesz odpowiedzi na niemal wszystkie pytania dotyczące funkcjonowania sieci komputerowych. Poznasz fundamenty ich działania (wielowarstwowy model ISO OSI) oraz zaznajomisz się z ich historią, rodzajami czy dostępnymi protokołami. Ponadto dowiesz się więcej o sposobach programowania aplikacji intensywnie korzystających z sieci, organizacji sieci Internet oraz najlepszych praktykach tworzenia aplikacji webowych. W części drugiej autor skupia się na fizycznych aspektach transmisji danych. Zrozumiesz, jak przesyłane są sygnały oraz jakie media transmisyjne masz do dyspozycji. To tylko niektóre zagadnienia poruszone w tym niezwykłym kompendium wiedzy na temat sieci komputerowych, będącym lekturą obowiązkową dla każdego administratora.

#### W TEJ KSIĄŻCE ZNAJDZIESZ:

- szereg interesujących informacji na temat historii i rozwoju sieci komputerowych
- omówienie zagadnień związanych z aplikacjami internetowymi i programowaniem sieciowym
- bogaty zbiór informacji na temat przesyłania sygnałów i informacji
- prezentację dostępnych topologii sieci komputerowych
- opis zalet i wad sieci bezprzewodowych

#### Bogate i kompletne źródło informacji o sieciach komputerowych!

Nr katalogowy: 6816



Księgarnia internetowa

<http://helion.pl>



Zamówienia telefoniczne:

0 801 339900



0 601 339900



**Helion**

Sprawdź najnowsze promocje:

• <http://helion.pl/promocje>

Książki najczęściej czytane:

• <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

• <http://helion.pl/nowosci>

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

<http://helion.pl>

**helion.pl**  
księgarnia  
internetowa

Cena 89,00 zł

ISBN 978-83-246-3607-5



9 788324 636075

Informatyka w najlepszym wydaniu