

Jak zawsze, chcialbym podziekowac mojej rodzinie za wsparcie i za wytrwanie ze mna podczas pisania tej ksiazki. Chcialbym również podziekowac rozwlosni WWOZ New Orleans za obecnosc w Internecie i doskonala muzyke, ktora towarzyszyla mojemu pisaniu. Trzymajcie tak dalej — Rob Scrimger

Dla Karen, ktora nauczyla mnie jak kochac i uczyc sie na nowo, oraz dla moich chlopcow, Ryana i Shawna, którzy, oprócz dostarczania inspiracji, dali mi wystarczajaco duzo przestrzeni do pracy nad tym projektem przez czas, który musial wydawac sie wiecznoscia — Paul LaSalle

Ta ksiazka jest dedykowana mojej Matce, ktora byla dla mnie, w trakcie pisania, zrodlem olbrzymiego wsparcia i inspiracji. Jak zawsze byla przy mnie. Bez Niej napisanie tej ksiazki nie byloby dla mnie mozliwe — Mridula Parihar

Ta ksiazka jest dla mojej rodziny. Dziekuje, Mamo i Tato, za obecnosc i za to, ze nigdy nie narzekaliscie, mimo ze poswiecalam pracy bardzo duzo czasu. Vikas, mój drogi bracie, nie moglabym ukonczyc tej pracy bez Twojego poczucia humoru i braterskiego wsparcia — Meeta

Serdeczne podziekowania dla Rona Gilstera i Diany McMichael za cenne nauki oraz dla Katie Feltman za umozliwienie powstania tej ksiazki. Dziekuje moim trzem aniołom — Kayli, Lauren i Mitchell — za dostarczanie mi powodów do usmiechu co dzien — Clay Leitzke

Podziękowania

Chciałbym podziękować Gabrielle za jej cieczka pracę, dzięki której to, co napisalem, wygląda na język angielski. Ponieważ jestem nie tylko dyslektykiem, lecz również marnym maszynista, naprawdę było to dla niej wyzwaniem. Chciałbym również podziękować Amandzie za cierpliwość; być może kiedyś ukończy książkę na czas. Dziekuje Nancy, Ami i Katie za urzeczywistnienie projektu. Na koniec chciałbym podziękować Timowi za to, że pomógł mi przez ten cały czas zachować uczciwość — Rob Scrimger

Wielkie dzięki dla „Zabójczej Katie”, Amandy, Nancy „Morderczyny z Long Island” i Scrima za madre porady, których mi nie szczędzili, gdy ich potrzebowałem — Paul LaSalle

Dziekuje Anicie Sastry za współpracę i porady. Chciałbym też podziękować mojej przyjaciółce i współautorce Meecie Gupta za towarzystwo w trakcie pisania tej książki — Mridula Parihar

Suchi, Namrata, Sunil, Anghsuman, Anita, Ashok, Rashim i Kurien zasługują na szczególnie serdeczne podziękowania za nieocenione wsparcie i pomoc, które kazde z nich dalo mi podczas pisania tej książki. Dziekuje — Meeta Gupta

Chciałbym podziękować Ronowi Gilsterowi i Dianie McMichael Gilster za nauki oraz Katie Feltman za umożliwienie powstania tej książki. Przede wszystkim chce podziękować moim dzieciom, Kayli, Lauren i Mitchell, za nieustającą miłość i wsparcie oraz za pokazywanie, że życie powinno być radosne — Clay Leitzke

Wydawnictwo Hungry Minds pragnie podziękować swojej letniej stazystce, Leslie Kersey, za cieczka pracę nad tym projektem.

Rzut oka na ksiazke

O Autorach.....	15
Wstep.....	17
Czesc I Wprowadzenie do transmisji TCP/IP	19
Rozdzial 1. Podstawy dzialania sieci komputerowych	21
Rozdzial 2. Architektura protokolu TCP/IP.....	45
Rozdzial 3. Warstwa fizyczna.....	65
Rozdzial 4. Warstwa interfejsu sieciowego	85
Rozdzial 5. Warstwa internetowa.....	99
Rozdzial 6. Warstwa transportowa.....	125
Rozdzial 7. Warstwa aplikacji.....	143
Czesc II Praca z TCP/IP	157
Rozdzial 8. Instalacja i konfiguracja TCP/IP.....	159
Rozdzial 9. Konfiguracja automatyczna	179
Rozdzial 10. Znajdowanie hostow w sieci IP.....	195
Czesc III Popularne aplikacje TCP/IP	223
Rozdzial 11. Dostep do Internetu.....	225
Rozdzial 12. Narzedzia do obslugi plikow.....	255
Rozdzial 13. Narzedzia zdalnego wykonywania polecen	271
Rozdzial 14. Drukowanie przez siec.....	291
Rozdzial 15. Aplikacje i protokoly WWW	305
Rozdzial 16. Dostep do poczty elektronicznej i grup dyskusyjnych.....	329
Rozdzial 17. Uslugi informacyjne dla przedsiebiorstw.....	345
Czesc IV Tworzenie i utrzymanie sieci TCP/IP	363
Rozdzial 18. Wybór schematu adresowania.....	365
Rozdzial 19. Projektowanie trasowania dla sieci.....	383
Rozdzial 20. Planowanie rozmieszczenia serwerow.....	411
Rozdzial 21. Wprowadzenie do lacznosci.....	429
Rozdzial 22. Planowanie bezpieczenstwa sieci.....	445
Rozdzial 23. Rozwiązywanie problemów z siecią i łącznością	463
Rozdzial 24. Monitorowanie sieci TCP/IP.....	481
Rozdzial 25. Plany na przyszlosc.....	505
Dodatki.....	515
Dodatek A Domeny DNS najwyższego poziomu.....	517
Skorowidz	539

Spis tresci

O Autorach.....	15
Wstep	17
Czesc I Wprowadzenie do transmisi TCP/IP.....	19
Rozdzial 1. Podstawy dzialania sieci komputerowych.....	21
Co to jest siec komputerowa?	21
Elementy składowe sieci.....	22
Rodzaje konfiguracji sieci.....	22
Sieci zdecentralizowane.....	22
Sieci scentralizowane.....	23
Model odniesienia OSI	24
Warstwa aplikacji	26
Warstwa prezentacji.....	26
Warstwa sesji.....	27
Warstwa transportowa.....	28
Warstwa sieciowa.....	28
Warstwa lacza danych	29
Warstwa fizyczna.....	29
Podzial sieci wedlug zasiegu	30
Sieci lokalne.....	30
Sieci rozlegle	30
Model z projektu IEEE 802	31
Topologie sieci	32
Topologia magistrali	32
Topologia gwiazdy	33
Topologia pierscienia.....	33
Topologia oczkowa.....	34
Topologie hybrydowe.....	34
Infrastruktura sieciowa.....	35
Regeneratorы	36
Karta interfejsu sieciowego	36
Koncentrator	36
Przelacznik.....	37
Most.....	37
Ruter	37
Bruter.....	37
Brama	38

Wprowadzenie do TCP/IP	38
Request for Comments	38
Model odniesienia TCP/IP.....	40
Przegląd adresowania IP	42
Aplikacje TCP/IP.....	43
Rozdział 2. Architektura protokołu TCP/IP.....	45
Pieciowarstwowa architektura TCP/IP	45
Warstwa fizyczna.....	47
Warstwa interfejsu sieciowego	53
Warstwa internetowa.....	56
Warstwa transportowa.....	57
Warstwa aplikacji	59
Laczosc pomiedzy warstwami	60
Format nagłówka warstwy transportowej.....	62
Format nagłówka warstwy internetowej.....	63
Rozdział 3. Warstwa fizyczna.....	65
W jaki sposób sygnał przesyłany jest kablem.....	65
Metody transmisji (metody sygnalizacji)	66
Technologie i mechanizmy transmisji.....	67
Nosniki fizyczne.....	70
Modemy	74
Nosniki bezprzewodowe.....	75
Najczesciej stosowane topologie	77
Magistrala	77
Token Ring	79
Gwiazda.....	80
FDDI.....	81
Sieci ATM	82
Rozdział 4. Warstwa interfejsu sieciowego.....	85
Warstwa interfejsu sieciowego — omówienie	85
Zawartosc ramki Ethernet.....	86
Typowe składniki pakietu sieciowego	88
Standardy sterowania dostepem do nosnika	89
Ethernet.....	89
ARCnet	91
Token Ring	91
ATM	92
Odwzorowanie adresów fizycznych na adresy IP.....	94
ARP i RARP.....	94
ATMARP.....	97
Rozdział 5. Warstwa internetowa.....	99
Przeznaczenie warstwy internetowej	99
Ustalenie, czy adres docelowy jest lokalny czy odległy	100
Wprowadzenie do trasowania.....	101
Adresy IP.....	101
Notacja dwójkowa i dziesiętna	102
Identyfikatory sieci i hostów	104

Klasy adresów IPv4.....	105
O czym informuje adres IP	106
Jak stosuje się maskę podsieci	107
Brama domyslna	107
Ustalenie czy adres docelowy jest lokalny, czy zdalny	108
Podstawy trasowania.....	109
Rutery sprzętowe i programowe.....	110
Typy tras	110
Zawartosc datagramu IP.....	115
Nagłówek IP	115
Ladunek IP	116
Protokół ICMP.....	116
Przeznaczenie ICMP.....	116
Pakiety ICMP.....	117
Protokół IGMP.....	119
Wprowadzenie do transmisji grupowych.....	120
Do czego sluzi adresowanie grupowe.....	122
Pakiety IGMP.....	122
Rozdzial 6. Warstwa transportowa.....	125
Typy przesyłu danych.....	125
Dostawy wiarygodne i dostawy nie gwarantowane.....	128
Dostawy stanowe i bezstanowe.....	128
Bezpolaczeniowe przesyłanie danych.....	130
Polaczeniowe przesyłanie danych.....	132
Inicjacja sesji	133
Maksymalny rozmiar segmentu.....	137
Okna nadawania i odbioru TCP.....	137
Okno przeciazenia	138
Algorytm powolnego startu.....	139
Nagłówek TCP.....	139
Rozdzial 7. Warstwa aplikacji	143
Przegląd portów	143
Dobrze znane numery portów.....	144
Gniazda — wprowadzenie	147
Dwukierunkowa łączność oparta na gniazdach	147
RPC.....	153
Czesc II Praca z TCP/IP	157
Rozdzial 8. Instalacja i konfiguracja TCP/IP	159
Konfiguracja TCP/IP.....	159
Informacje potrzebne zawsze.....	159
Informacje potrzebne czasami	160
Konfiguracja TCP/IP w swiecie Linuksa.....	161
Instalacja i konfiguracja TCP/IP w swiecie Microsoftu.....	169
Instalacja TCP/IP w systemach operacyjnych Microsoftu.....	169
Reczna konfiguracja TCP/IP	172
Kontrola konfiguracji IP	177

Rozdział 9. Konfiguracja automatyczna	179
Wprowadzenie do konfiguracji automatycznej.....	179
Korzyści z konfiguracji automatycznej.....	180
Konfiguracja w sieciach wielosegmentowych.....	181
Protokół BOOTP.....	181
Proces ładowania początkowego BOOTP.....	182
Zawartość pakietu BOOTP.....	182
Rutery obsługujące protokół BOOTP.....	184
Wady protokołu BOOTP.....	185
DHCP.....	185
Dzierżawy DHCP.....	186
Opcje zakresu i serwera.....	189
Pakiet DHCP.....	190
Opcje serwera DHCP.....	191
Trasowanie DHCP.....	192
Rozdział 10. Znajdowanie hostów w sieci IP.....	195
Przegląd nazw hostów.....	195
Podstawowe nazwy hostów.....	197
Pełne zlozone nazwy domen.....	197
Nazwy kanoniczne i aliasy	197
Lokalny plik HOSTS.....	199
Format pliku HOSTS.....	199
Rozwiązywanie nazw	200
Wykorzystanie usługi DNS do rozwiązywania nazw hostów.....	200
Czym jest domena?.....	202
Serwery nazw.....	202
Resolwery	202
Przestrzeń nazw	202
Strefy w obrębie przestrzeni nazw.....	205
Tworzenie pliku strefy	207
Zapytania iteracyjne i rekurencyjne.....	210
Konfiguracja DNS-u z wykorzystaniem programu BIND	211
Konfiguracja Windows 2000.....	212
Rozwiązywanie nazw NetBIOS.....	214
Nazwy NetBIOS — co to jest?.....	214
Składniki sieciowe Microsoftu.....	215
Rozwiązywanie nazw NetBIOS przed Windows 2000.....	216
Rozwiązywanie nazw NetBIOS w Windows 2000	220
Część III Popularne aplikacje TCP/IP	223
Rozdział 11. Dostęp do Internetu	225
Przegląd miedzysieci prywatnych i publicznych.....	226
Adresowanie w sieciach prywatnych.....	227
Ograniczenia IPv4	229
Łączenie się z Internetem.....	231
Dostawcy usług internetowych.....	233
Wykorzystanie zapór firewall	234
Rola zapór firewall	234
Typy zapór firewall.....	236
Najczęściej stosowane konfiguracje sieci z zaporami firewall.....	239

Sposobowanie NAT.....	242
Korzyści ze stosowania NAT.....	245
Przeczysty czy nieprzeczysty	246
Wykorzystanie serwera proxy	246
Udostępnianie połączenia internetowego Microsoftu.....	247
Wirtualne sieci prywatne	248
PPTP	251
Layer-2 Tunneling Protocol.....	253
Rozdział 12. Narzedzia do obsługi plików.....	255
NFS	255
Wprowadzenie do NFS.....	255
Usługi NFS	257
Zagadnienia bezpieczeństwa w NFS.....	258
Wersje NFS.....	258
Konfiguracja serwera NFS	260
DFS	263
Wprowadzenie do DFS.....	263
Katalogi główne DFS: autonomiczny i domeny.....	264
Konfiguracja DFS w Windows 2000.....	264
Narzedzia do przesyłania plików	266
FTP	266
TFTP.....	268
Remote Copy Protocol.....	268
Rozdział 13. Narzedzia zdalnego wykonywania poleceń.....	271
Przegląd narzędzi zdalnego wykonywania poleceń	271
Telnet	272
Remote login.....	278
Remote shell (rsh).....	280
Secure shell (ssh)	281
Remote execute (rexec)	284
Serwery terminali	284
Sun Ray	285
Microsoft Terminal Server	287
Citrix	289
Rozdział 14. Drukowanie przez sieć.....	291
Wprowadzenie do drukowania.....	291
Drukowanie w środowisku linuksowym.....	292
Drukowanie w systemach Microsoftu.....	294
Drukowanie z klienta.....	295
Konfiguracja serwera lpd.....	296
Zdalne drukarki w systemach Unix i Linux	296
Narzędzie printtool.....	297
Łączenie z lokalna drukarka	297
Łączenie ze zdalna drukarka	299
Polecenia związane z drukowaniem	301
Internet Printing Protocol Microsoftu.....	304
Administratorzy	304
Pozostali użytkownicy	304

Rozdział 15. Aplikacje i protokoly WWW	305
Podstawy WWW.....	305
Internet — wprowadzenie.....	305
Ewolucja WWW	306
Jak funkcjonuje WWW.....	307
HTML.....	308
HTTP.....	310
World Wide Web Consortium	311
Aplikacje WWW.....	313
Serwery WWW	313
Aplikacje w Internecie	314
Języki.....	316
Bezpieczenstwo w Sieci	322
Handel elektroniczny w Internecie	324
Wideo i inne zaawansowane typy danych.....	325
Potokowa transmisja audio i wideo	325
Co trzeba brac pod uwage przy transmisji potokowej.....	327
Rozdział 16. Dostep do poczty elektronicznej i grup dyskusyjnych.....	329
Wprowadzenie do poczty elektronicznej.....	329
SMTP.....	331
POP.....	332
IMAP.....	333
Czytanie poczty	334
MIME i S/MIME.....	336
PGP.....	339
Grupy dyskusyjne — wprowadzenie	339
Serwery i koncentratory.....	341
NNTP.....	342
Netykieta.....	343
Rozdział 17. Usługi informacyjne dla przedsiębiorstw.....	345
Wprowadzenie do sieciowych uslug katalogowych	345
Standard X.500.....	347
LDAP.....	350
NIS.....	353
NIS+.....	355
STDS.....	356
Network Directory Service Novella	357
Active Directory	360
Czesc IV Tworzenie i utrzymanie sieci TCP/IP.....	363
Rozdział 18. Wybór schematu adresowania.....	365
Szacowanie potrzeb dotyczących adresów	365
Fizyczna konfiguracja sieci	365
Lokalizacje obsługiwane przez siec.....	366
Wymogi wydajnosci	367
Adresy publiczne i prywatne.....	369
Uzyskanie adresu i polaczenia z Internetem.....	369
Obliczanie potrzeb adresowych.....	370

Podzial na podsieci.....	375
Obliczanie ID lokalizacji	375
Obliczanie ID podsieci	379
Ustalenie adresów hostów.....	380
Rzut oka na nadsieci	381
Rozdzial 19. Projektowanie trasowania dla sieci	383
Podstawy trasowania.....	383
Tablica tras.....	384
Budowanie tablicy tras	386
Statyczny wybór trasy.....	388
Tworzenie struktury trasowania.....	389
Laczanie podsieci.....	390
Maski podsieci o zmiennej dlugosci.....	392
Podlaczanie odleglych biur.....	395
Dynamiczny wybór tras	396
ICMP Router Discovery	397
Protokół RIP	398
Protokół IGRP	403
OSPF.....	405
Rozdzial 20. Planowanie rozmieszczenia serwerów	411
Ustalenie uslug potrzebnych w sieci.....	411
Instalowanie uslug w sieci	413
Laczanie uslug.....	415
Planowanie równowazenia obciążenia i nadmiarowosci	419
Dodawanie kolejnych systemów	419
Systemy wieloadresowe.....	421
Serwery hierarchiczne	422
Stosowanie grupowania	425
Rozdzial 21. Wprowadzenie do lacznosci.....	429
Podstawy lacznosci.....	430
Laczanie lokalizacji	430
Budowanie własnej sieci WAN.....	437
Planowanie dostepu zdalnego	441
Wybór strategii polaczen telefonicznych.....	441
Praca zdalna	442
Rozdzial 22. Planowanie bezpieczenstwa sieci.....	445
Szacowanie ryzyka.....	445
Równoważenie bezpieczeństwa i uzytecznosci.....	448
Zabezpieczanie sieci	449
Szyfrowanie transmisji danych.....	449
Uwierzytelnianie uzytkowników.....	451
Jednoczesne stosowanie szyfrowania i uwierzytelniania	457
Rozdzial 23. Rozwiązywanie problemów z siecią i lacznoscia.....	463
Proces rozwiązywania problemów.....	464
Sprawdzenie konfiguracji IP	465
Kontrola konfiguracji IP dla Microsoft Windows.....	465
Kontrola konfiguracji IP w systemach uniksowych.....	467

Testowanie łączności	468
Znajdowanie problemów z rozwiązywaniem nazw	474
Znajdowanie problemów z rozwiązywaniem nazw hostów.....	474
Znajdowanie problemów w rozwiązywaniu nazw NetBIOS.....	477
Weryfikacja klienta i serwera	480
Rozdział 24. Monitorowanie sieci TCP/IP	481
Monitorowanie sprzętu.....	482
Wymogi dla serwerów uwierzytelniających.....	482
Wymogi dla serwerów plików i drukowania.....	483
Wymogi dla serwerów aplikacji.....	483
Narzędzia monitorujące	484
Narzędzia do monitorowania sieci	487
Monitorowanie sieci za pomocą polecenia ping.....	487
Monitorowanie sieci za pomocą polecenia netstat	488
Monitorowanie sesji NetBIOS za pomocą narzędzia nbtstat	493
Przechwytywanie ruchu sieciowego za pomocą analizatorów pakietów.....	494
SNMP.....	498
Community name.....	499
System zarządzania SNMP.....	499
Agent SNMP.....	500
Baza informacji zarządzania.....	500
Regulacja rozmiaru okna TCP/IP	501
Rozdział 25. Plany na przyszłość	505
Wprowadzenie do IPv6.....	506
Zmiany w porównaniu z IPv4.....	507
Adresowanie IPv6.....	508
Bezprzewodowy Internet	508
Wireless Datagram Protocol	510
Wireless Transport Layer Security	510
Wireless Transaction Protocol.....	511
Wireless Session Protocol.....	511
Wireless Application Environment.....	511
Inteligentne urządzenia domowe	512
Planowanie na przyszłość.....	514
Dodatki.....	515
Dodatek A Domeny DNS najwyższego poziomu	517
Ogólne domeny najwyższego poziomu	517
Specjalne domeny najwyższego poziomu.....	517
Narodowe domeny najwyższego poziomu z poddomenami.....	518
Skorowidz	539

O TCP/IP napisano wiele ksiazek — jak implementowac, jak zabezpieczac i jak wyliczac maski podsieci. Jednakze „Biblia TCP/IP” gromadzi w jednym miejscu omówienie wszystkich wzajemnych tematów zwiazanych z TCP/IP, od modelu TCP/IP, az do porad, jak zaimplementowac ten pakiet protokolów. Dane w tej ksiazce sa w miare mozliwosci ogólne, aby mozna bylo wykorzystac je w systemach Solaris, Linux czy nawet Windows 2000.

Tytul oryginalu: TCP/IP Bible
Tłumaczanie: Adam Jarczyk

ISBN: 83-7197-668-2

Original English language edition Copyright © 2002 by Hungry Minds, Inc.
All rights reserved including the right of reproduction in whole or in part of any form. This translation published by arrangement with Hungry Minds, Inc.
Photoshop is a trademark of Adobe System, Inc. The Bible trade dress is a trademark of Hungry Minds, Inc. in the United States and/or other countries. Used by permission.

Polish language edition published by Wydawnictwo Helion.
Copyright © 2002

Wydawnictwo HELION
ul. Chopina 6, 44-100 GLIWICE
tel. (32) 231-22-19, (32) 230-98-63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (ksiegarnia internetowa, katalog ksiazek)

Drogi Czytelniku!
Jezeli chcesz ocenic te ksiazke, zairzyj pod adres
<http://helion.pl/user/opinie?tcpipb>
Mozesz tam wpisac swoje uwagi, spostrzezenia, recenzje.

Wszystkie znaki wystepujace w tekscie sa zastrzezonymi znakami firmowymi badz towarowymi ich wlascicieli.

Autor oraz Wydawnictwo HELION dolozyli wszelkich staran, by zawarte w tej ksiazce informacje byly kompletne i rzetelne. Nie biora jednak zadnej odpowiedzialnosci ani za ich wykorzystanie, ani za zwiazane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponosza również zadnej odpowiedzialnosci za ewentualne szkody wynikle z wykorzystania informacji zawartych w ksiazce.

Wszelkie prawa zastrzezone. Nieautoryzowane rozpowszechnianie calosci lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metoda kserograficzna, fotograficzna, a takze kopiacie ksiazki na nosniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Printed in Poland.

O Autorach

Rob Scrimger pracował jako operator komputera, programista, szkoleniowiec, administrator sieci i kierownik sieci w różnych przedsiębiorstwach. W trakcie swej kariery zdobył rzetelną wiedzę na temat protokołu TCP/IP i pomagał w pisaniu kilku innych książek o TCP/IP, w tym „Networking with Microsoft TCP/IP: Certified Administrator's Resource Edition” oraz „MCSE Training Guide: TCP/IP 2nd Edition”. Rob ma na koncie trzy tytuły MCSE: NT 3.5, NT 4.0 i Windows 2000 (członek statutowy), a poza tym MCT, MCDBA, MCSE+I, MCP+SB, CTT, A+ Certified Technician i Network + Certified.

Paul LaSalle posiada tytuł MCSE (*Microsoft Certified Systems Engineer*) i jest prezesem Enchanted Forest Systems — firmy szkoleniowej i konsultanckiej w zakresie sieci komputerowych z siedzibą w Rockland, Ontario w Kanadzie. Paul, gdy nie jest zajęty pisaniem, konsultacjami lub szkoleniem, chętnie spędza czas z rodziną i poświęca się innym swoim pasjom, do których należą: stolarstwo, wędkowanie, biwakowanie, ogrodnictwo i muzyka. Można się z nim skontaktować pod adresem paul@efs.ca.

Mridula Parihar w swoim życiu zawodowym zajmowała się głównie szkoleniami w wiodącej indyjskiej firmie szkoleniowej zajmującej się tematyką informatyczną — NIIT Ltd., zarówno ucząc, jak i opracowując materiały dla instruktorów. Mridula posiada tytuł Microsoft Certified Solution Developer (MCSD). Jej doświadczenie w nauczaniu obejmuje prowadzenie sesji dotyczących pojednaniowych i zarządzania sieciami lokalnymi.

Meeta Gupta przez trzy lata pracowała jako szkoleniowiec w NIIT i posiada tytuł magistra informatyka. Meeta jest wykwalifikowanym inżynierem z tytułem Certified Novell Engineer. Jest fachowcem w zakresie sieci komputerowych i rozwiązywania problemów z sieciami.

Clay Leitzke jest prezesem i CEO Northwest Computer Training Center w Coeur d'Alene w Idaho. Większość czasu spędza ucząc na „obozach” szkoleniowych Windows 2000 Bootcamp na terenie całych Stanów Zjednoczonych. Jego firma certyfikowała ponad 2000 tytułów MCSE. Clay posiada tytuły MCSE i MCT już od NT 3.5, a jego specjalnością jest Exchange; poza tym posiada certyfikaty firm Cisco, CompTIA i Novell. Clay jest dostępny pod adresem www.nexusww.com.

Wstep

Dwa slowa o tej ksiazce

O TCP/IP napisano wiele ksiazek — jak implementowac, jak zabezpieczac i jak wyliczacz maski podsieci. Jednakze „Biblia TCP/IP” gromadzi w jednym miejscu omówienie wszystkich ważnych tematów związanych z TCP/IP, od modelu TCP/IP, az do porad, jak zaimplementowac ten pakiet protokolów. Dane w tej ksiazce sa w miare mozliwosci ogólne, aby mozna bylo wykorzystac je w systemach Solaris, Linux czy nawet Windows 2000.

Proszę potraktowac te ksiazke jako punkt wyjściowy — jesli ktos woli, wprowadzenie — do protokolu TCP/IP. Nie jest ona kompendium wiadomosci o współpracy protokolu z kazda platforma i Czytelnikowi nie bedzie potrzebny wózek widlowy, by zawiezc ja do domu. Jesli jednak Czytelnik zacjal juz pracowac z TCP/IP, ksiazka ta bedzie dla niego bezcennym leksykonem. Przedstawia narzecza potrzebne do szerszego poznania pakietu protokolów TCP/IP oraz jest punktem wyjścia do uruchomienia serwera WWW, zaimplementowania IPSec, czy wyboru systemu uslug katalogowych dla przedsiebiorstwa. Stawiajac czolo kolejnym wyzwaniom, Czytelnik bedzie mógł wracac do tego tekstu niejeden raz.

Rozklad ksiazki

Ogólnie rzecz biorac, pomyslna implementacja TCP/IP sklada sie z czterech etapów. Przede wszystkim musimy zrozumiec podstawy dzialania protokolów z pakietu TCP/IP, procesy i teorie. Nastepnie powinnismy nauczyc sie, jak w praktyce pracowac z TCP/IP, jak instalowac, konfigurowac i jak odnajdywac inne komputery. Gdy juz zainstalujemy i uruchomimy TCP/IP oraz systemy, pora cos z nimi zrobic. Oznacza to, iz musimy zainstalowac serwery dajace uzytkownikom funkcjonalnosć sieci. Po opanowaniu funkcji TCP/IP oraz zorientowaniu sie, co potrafi TCP/IP, musimy nauczyc sie planowac implementacje TCP/IP. Niniejsza ksiazka jest podzielona na cztery czesci, odzwierciedlające cztery obszary wiedzy, która Czytelnik musi zdobyc.

Czesc I — Wprowadzenie do transmisiJI TCP/IP

W czesci I omówione zostały podstawy TCP/IP: jak dziala stos protokolów i co możemy za jego pomoca zrobic. Ta czesc zawiera wprowadzenie do stosu TCP/IP, a nastepnie omawia kazda warstwe szczegółowo, zaczynajac od warstwy fizycznej. Nastepnie, przechodzacz w góre stosu, poznamy pozostałe warstwy — interfejsu sieciowego, inter-

netowa, transportowa i aplikacji. Informacje te moga wydawac sie nieco ezoteryczne, lecz zrozumienie funkcji warstw i ich wzajemnych interakcji jest kluczem do rozwiązywania trudnych problemów.

Czesc II — Praca z TCP/IP

Te czesc rozpoczyna instrukcja instalowania i konfiguracji TCP/IP. Nastepnie zajmujemy sie tematyka nazewnictwa i rozwiązywania nazw. Poniewaz nazwy sa latwiejsze do zapamietania od numerów — w koncu zwykle nie uczymy sie na pamiec adresów IP, zarówno internetowych, jak i intranetowych — potrzebna jest metoda pozwalajaca użytkownikom znalezc inne systemy w sieci i polaczyc sie z nimi. Tutaj wazne staje sie nazewnictwo. Zrozumienie nazw i rozwiązywania nazw pomoze tez Czytelnikowi rozwiązywac problemy z lacznoscia.

Czesc III — Popularne aplikacje TCP/IP

Czesc III charakteryzuje różnorodne zastosowania TCP/IP i przedstawia przeglad najbardziej aktualnych aplikacji uzywajacych tego protokolu, takich jak NFS i HTTP. Dzieki temu Czytelnik uzyska wiedze na temat uslug, które zaspokajaja okreslone potrzeby w sieci komputerowej.

Czesc IV — Tworzenie i utrzymanie sieci TCP/IP

Czesc IV zawiera informacje, które pomoga zaimplementowac poznane rozwiązania. Tu-taj nasze opisy koncentruja sie raczej na wytycznych niz na faktach — na przykład, jak obliczyc dopuszczalna liczbe hostów w pojedynczym segmencie sieci. Zakonczymy opisem kilku technologii, które beda mialy wpływ na prace Czytelnika w najblizszej przyszlosci, na przykład bezprzewodowego Internetu i inteligentnych urzadzen domowych.

Symbole stosowane w tej ksiazce

W niniejszej ksiazce zastosowalismy kilka symboli obrazkowych, wyróżniajacych wazne informacje.



Ostrzeżenie
Ten symbol oznacza informacje o potencjalnych problemach z planowaniem, implementacją lub funkcjonalnością.



Odnosnik
Ten symbol kieruje do uzytecznych informacji zawartych w innych rozdziałach ksiazki.



Uwaga
Ten symbol oznacza dodatkowe informacje o omawianym właśnie zagadnieniu.



Wskazówka
Ten symbol wskazuje na zyczliwą rade autorów.

Czesc I

Wprowadzenie do transmisji TCP/IP

W tej czesci:

- ◆ Rozdzial 1. Podstawy dzialania sieci komputerowych
- ◆ Rozdzial 2. Architektura protokolu TCP/IP
- ◆ Rozdzial 3. Warstwa fizyczna
- ◆ Rozdzial 4. Warstwa interfejsu sieciowego
- ◆ Rozdzial 5. Warstwa internetowa
- ◆ Rozdzial 6. Warstwa transportowa
- ◆ Rozdzial 7. Warstwa aplikacji

W czesci I zajmiemy sie podstawami transmisji TCP/IP. Omówimy stos protokolu TCP/IP oraz współprace jego warstw podczas przesyłania danych pomiędzy dwoma hostami.

Rozdzial 1. omawia podstawy sieci komputerowych i pokrótce opisuje model OSI. Model ten zostanie porównany z uproszczonym modelem TCP/IP w rozdziale 2., w którym zdefiniujemy warstwy i omówimy przeznaczenie każdej z nich. W rozdziałach od 3. do 7. każda z warstw zostanie omówiona szczegółowo.

Bieżaca czesc zawiera podstawowe wiadomości o TCP/IP — protokole niezbednym w projektowaniu, wdrażaniu i rozwiązywaniu problemów z niemal wszystkimi współczesnymi sieciami komputerowymi.

Rozdział 1.

Podstawy działania sieci komputerowych

W tym rozdziale:

- ◆ Podstawy sieci komputerowych
- ◆ Składniki łączności sieciowej
- ◆ Konfiguracje sieci
- ◆ Model OSI
- ◆ Typy sieci według zasięgu
- ◆ Topologie sieciowe
- ◆ Infrastruktura sieci
- ◆ Wprowadzenie do TCP/IP

Wprowadzenie do TCP/IP Komputery mają wpływ na niemal każdy aspekt naszego życia. Niniejsza książka ma za zadanie umożliwić Czytelnikowi lepsze zrozumienie sposobów komunikowania się komputerów za pomocą protokołów *Transmission Control Protocol* i *Internet Protocol* (TCP/IP).

W ciągu ostatniej dekady nasze społeczeństwo zostało uznane za „pokolenie Internetu”. Nasze życie codzienne ulega wciąż nowym wpływom ze strony rozwijającego się Internetu. Możemy już za pomocą Sieci opłacać rachunki, szukać pracy, dokonywać rezerwacji przed podrózami i robić ponad milion innych rzeczy. Lecz zanim będziemy mogli zaglebić się w tajniki TCP/IP, musimy poznać podstawy działania sieci komputerowych.

W tym rozdziale zostały pokrótko omówione elementy składowe, umożliwiające łączność sieciową. Obejmuje on podstawy sieci komputerowych, w tym model OSI, topologie oraz adresowanie TCP/IP.

Co to jest sieć komputerowa?

W najbardziej podstawowym znaczeniu *sieć* oznacza dwa lub więcej komputerów korzystających ze wspólnych informacji. Sieci mogą być jednak bardzo różnorodne, mogą mieć rozmiary kilku klientów i milionów klientów. *Klient* jest to zasobnictwo zadajające usługi lub danych w sieci — komputerem ubiegającym się o przesył danych przez sieć.

Jedna z funkcji klienta moze byc na przykład sprawdzanie poczty elektronicznej. Klient zada informacji od serwera pocztowego, który z kolei zada informacji od klienta — przez co sam serwer pocztowy również staje sie klientem.

W szkole podstawowej uczyono nas, ze wszystkie rekiny sa rybami, lecz nie wszystkie ryby sa rekinami. Ta sama zasada stosuje sie do klientów i wezłów. Wszystkie klienci sa wezłami, lecz nie kazdy wezel jest klientem. Mimo to pojęcia *klient* i *wezel* sa często używane zamiennie. *Wezel* (ang. *node*) oznacza dowolne urządzenie w sieci, zawierające karte sieciowa aktywna w tejże sieci. Aktywny wezel generuje ruch w sieci w postaci zadan i odpowiedzi. Niektóre urządzenia, jak np. drukarki, routery i przełączniki, zazwyczaj nie wysyłają zadan w sieci. Urządzenia te odpowiadają na zadania innych klientów w sytuacjach, gdy cos jest potrzebne — na przykład połączenie lub plik. Takie urządzenia, jak routery i przełączniki zazwyczaj nie mają o co pytać klientów. Nie znaczy to, iż nie inicjalizują ruchu sieciowego; jedynie nie zadają usług od innych klientów.

Pierwszym wymogiem w sieci jest używanie przez wszystkie klienci (wezły) tego samego języka, czyli protokołu. Na potrzeby łączności sieciowej dostępnych jest mnóstwo protokołów; niniejsza książka koncentruje się jednakże na TCP/IP.

Elementy składowe sieci

Każdy sposób komunikacji — ustna, pisemna czy też elektroniczna — wymaga jakiegoś mechanizmu. Łączność sieciowa nie jest tu wyjątkiem. W podstawowej sieci jednym wymaganym mechanizmem jest karta interfejsu sieciowego (NIC — *Network Interface Card*). Sieć może składać się z dwóch klientów połączonych przewodem skrzyżowanym. Skrzyżowane kable pozwalają klientom nadawać i odbierać informacje pomiędzy sobą bez posrednictwa innych urządzeń łączących — koncentratorów, przełączników czy routerek. Dostępne są karty sieciowe dla wszelkich typów sieci.

W podstawowym znaczeniu topologia sieci to rozkład przewodów sieciowych. W dalszej części rozdziału omówimy różne typy topologii sieciowych i typy kart sieciowych, których można w nich używać.

Rodzaje konfiguracji sieci

Istnieją dwie podstawowe konfiguracje sieci: równorzędna („kazdy z każdym”) oraz klient-serwer. Można jednak dyskutować, czy środowiska sieciowe w pełni równorzędne czy całkowicie typu klient-serwer tak naprawdę istnieją. Wobec tego wprowadzone zostały pojęcia sieci scentralizowanej i zdecentralizowanej.

Sieci zdecentralizowane

Sieci *równorzędne* (zdecentralizowane) zostały kiedyś zdefiniowane jako sieci nie zawsze serwerów, a jedynie klientów. Inaczej mówiąc, każdy klient w sieci był w stanie zadać i dostarczać informacji. Nie istniał żaden centralny serwer, od którego wszystkie klienci zdobywali informacji. Z biegiem czasu pojawiły się tendencje do gromadzenia w pojedynczym kliencie sieci wszystkich plików dla pozostałych klientów. W wyniku

tego klient przechowujacy informacje zaczal byc uwasany za *serwer*. Sytuacja w wielu malych srodowiskach biurowych nadal wyglada podobnie.

Wraz ze zmianami potrzeb w srodowiskach sieciowych i wzrostem rozmiarów programow z pojedynczych megabajtów do setek megabajtów, zaczely sie upowszechniac *serwery specjalistyczne* (komputery sieciowe, pelniace funkcje jedynie serwerów). Specjalizowany serwer stal sie centralnym magazynem danych. Klienty zaczely zadac informacji od serwerów zamiast od siebie nawzajem; jednak zarzadzanie siecia nadal bylo zadaniem zmudnym. Zaden serwer nie zawieral wszystkich kont uzytkownikow — zamiast tego konta uzytkownikow byly utrzymywane w kazdym kliencie.

Gdy uzytkownik loguje sie w sieci, podaje nazwe uzytkownika i haslo. Jesli konta i hasla sa skladowane w roznich miejscach sieci, mamy do czynienia z siecia typu zdecentralizowanego (rownorzedna).

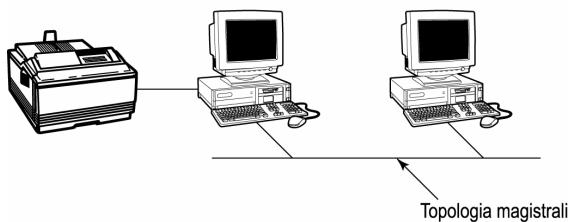
Zdecentralizowana siec posiada wiele dostepnych zasobow: serwery pocztowe, serwery baz danych, skladnice plikow, drukarki, czy tez programy graficzne, lecz obecnosc tych zasobow nie oznacza, iz siec jest scentralizowana; nie oznacza to rowniez konfiguracji klient-serwer. Sieci zdecentralizowane zazwyczaj posiadaja nastepujace właściwości:

- ◆ Male rozmiary, ograniczone do okolo 20 klientow w jednej sieci.
- ◆ Bezpieczenstwo nie jest wzorne.
- ◆ Nie jest wymagane zarzadzanie na poziomie sieci.
- ◆ Jest wymagane zarzadzanie na poziomie klientow.

Rysunek 1.1 przedstawia zdecentralizowane srodowisko sieciowe. Ten typ nosi również nazwy *siec równorzedna (peer-to-peer — dosłownie „kazdy z kazdym”)* lub *grupa robocza*.

Rysunek 1.1.
Siec zdecentralizowana

Środowisko komputerów
rownorzednych



Sieci scentralizowane

Siec *scentralizowana* (typu klient-serwer) jest siecia, w której przynajmniej jeden komputer jest wyznaczony do roli serwera. Serwer ten swiadczy uslugi dla kliencow, na przyklad obsluge poczty elektronicznej lub skladowanie plikow, a ponadto dostarcza informacji zadanych przez klienty.

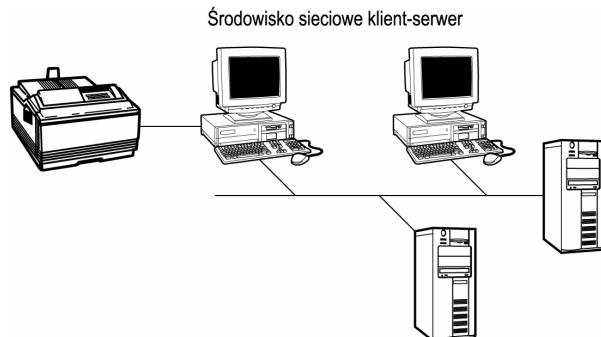
W sieciach scentralizowanych serwer, lub grupa serwerow, zawiera wszystkie informacje o kontach uzytkownikow. Microsoft oferuje Windows NT i Windows 2000 Server, zas Novell — NetWare eDirectory Services (NDS). Gdy konta uzytkownikow sieci przechowywane sa w pojedynczej bazie danych, taka siec nazywana jest scentralizowa-

na. Domeny Windows NT i 2000 oraz Novell Networks skladuja konta uzytkowników w centralnej bazie danych. Naklady pracy administracyjnej wlozone w zarzadzanie siecia scentralizowana sa nizsze niz w przypadku sieci zdecentralizowanej, poniewaz w tym drugim przypadku administrator musi udac sie do kazdego klienta, aby wykonac czynnosci zwiiazane z zarzadzaniem. W sieci scentralizowanej zarzadzanie moze odbywac sie z dowolnego klienta. Sieci takie zwykle charakteryzuja sie nastepujacymi wlasciwosciami:

- ◆ Wymagane jest zarzadzanie na poziomie sieci.
- ◆ Zarzadzanie poszczególnymi klientami jest ograniczone do minimum.
- ◆ Nie jest ograniczona dopuszczalna liczba klientów.

Czesc sieci scentralizowanych posiada mniej niz dziesiec klientów, zas niektóre — na przyklad siec U.S. Postal Service — maja ponad milion klientów. Rysunek 1.2 przedstawia siec scentralizowana. Serwery moga sluzyc do skladowania zasobów i kont uzytkowników na potrzeby wszystkich klientów.

Rysunek 1.2.
Siec scentralizowana



Zaden standard nie definiuje terminologii dla różnych typów sieci — równorzędnych lub scentralizowanych, lecz istnieja standardy określające, w jaki sposób odbywa się komunikacja w sieci.

Model odniesienia OSI

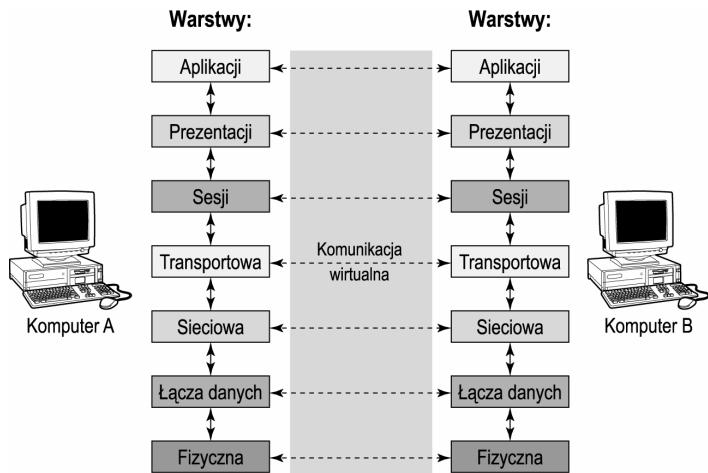
Ruch sieciowy generowany jest przy wyslaniu zadania przez siec. Zadanie musi zostac przekształcone z postaci, jaka widzi uzytkownik, do formatu nadajacego sie do uzycia w sieci. Transformacja ta jest mozliwa dzieki modelowi OSI (*Open Systems Interconnection*), opracowanemu przez ISO — *International Organization for Standardization*.

Dane przesypane sa w sieci w postaci pakietów danych. *Pakiet danych* to dane uzytkownika przekształcone na postac zrozumiala dla sieci. Kazde przekształcenie jest pochodna siedmiowarstwowego modelu OSI, który sluzy twórcom oprogramowania sieciowego za wytyczne. Chociaz wielu producentów manipuluje tym modelem, jest on nadal podstawa prac rozwojowych.

Siedem warstw modelu OSI, przedstawionego na rysunku 1.3, pelni funkcje elementów konstrukcyjnych pakietu danych. Kazda warstwa dodaje do pakietu danych informacje, lecz sam pakiet danych pozostaje niezmieniony. Informacje dodane do pakietu nosza

nazwe *nagłówka*. Nagłówek kazdej warstwy jest po prostu informacja, opisujaca formowanie pakietu danych. Nagłówek jest odbierany w odpowiedniej warstwie u klienta odbierajacego dane i sluzy do rozpoznania formatu pakietu. Kazda warstwa komunikuje sie z warstwami sasiednimi, znajdujacymi sie powyzej i ponizej. Rysunek 1.3 przedstawia siedmiowarstwowy model OSI.

Rysunek 1.3.
Model OSI



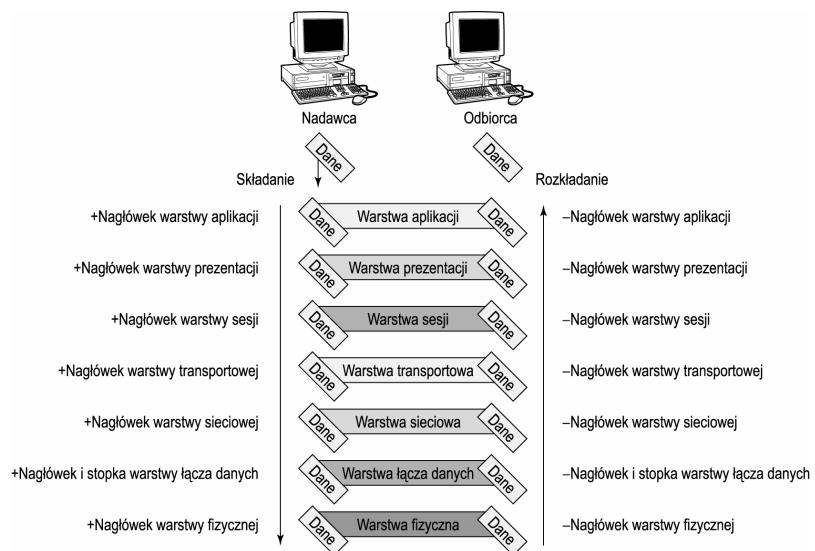
Komunikacja poprzez siedmiowarstwowy model OSI nie przebiega wedlug ostatecznie ustalonej sciezki, lecz zawsze odbywa sie w kierunku pionowym. Pakiety danych nie musza byc wysylane z warstwy 7, ktora jest warstwa najwyzsza — aplikacji. Lacznosc moze rozpoczac sie, na przyklad, w warstwie 3, lecz warstwy 2 i 1 musza zostac uzyte, aby dodac nagłówki.

Zalóżmy, ze Komputer A uzywa narzedzia, które zaczyna dzialanie w warstwie 3. Warstwa 3. dodaje nagłówki i przekazuje calosc do warstwy 2., która również dodaje nagłówki i przekazuje pakiet do warstwy 1. Ta dodaje nagłówek i umieszcza pakiet w sieci. Komputer B odbiera pakiet i przetwarza, zaczynajac od warstwy 1. Warstwa 1. usuwa nagłówek dodany przez warstwe 1. Komputera A i przekazuje pozostałe informacje do warstwy 2. Ta nastepnie usuwa nagłówek dodany przez warstwe 2. Komputera A i przekazuje pozostałe informacje do warstwy 3. Warstwa 3. usuwa nagłówek dodany przez warstwe 3. Komputera A i przetwarza zadanie.

Wszystkie siedem warstw jest w uzytku *jedynie* wtedy, gdy zadanie pochodzi od uzytkownika. Niezależnie od tego, która warstwa rozpoczyna komunikacje, nagłówki dodawane sa na kazdym poziomie i usuwane na odpowiadajacym mu poziomie u klienta odbierajacego pakiet, jak na rysunku 1.4. Pakiet danych jest przesyłany od nadawcy po lewej stronie do odbiorcy po prawej. Podczas przekazywania pakietu w dół z warstwy 7. do 1., kazda z nich dodaje nagłówek. Gdy pakiet dociera do odbiorcy, kazdy nagłówek jest usuwany, zas dane przekazywane sa do nastepnej, wyzszej warstwy.

Rysunek 1.4.

Przepływ danych od nadawcy do odbiorcy w modelu OSI



Warstwa aplikacji

Najwyższa, siódma warstwa w modelu OSI jest *warstwa aplikacji*. Jest ona odpowiedzialna za interakcje z aplikacją użytkownika; przyjmuje dane od programu i świadczy usługe aplikacji sieciowej, odpowiedzialnej za zadanie użytkownika. Kilka przykładów przekształcania danych w warstwie aplikacji:

- ◆ Gdy użytkownik wysyła list e-mail, warstwa aplikacji daje dostęp do usługi SMTP (*Simple Mail Transfer Protocol*).
- ◆ Przesyłu pliku można dokonać za pomocą protokołu FTP (*File Transfer Protocol*). Za usługę FTP odpowiedzialna jest warstwa aplikacji.
- ◆ Zadanie dostępu z przeglądarki do witryny WWW, np. www.nwcomputertraining.com, umieszcza w warstwie aplikacji zadanie rozwiązania nazwy przez usługę DNS oraz zadanie protokołu dla HTTP.

W warstwie aplikacji istotnie znajdują się aplikacje, lecz są one niewidoczne dla użytkownika. Warstwa ta jest *jedyną* warstwą, która bezpośrednio komunikuje się z oprogramowaniem użytkownika.

Warstwa prezentacji

Szósta warstwa modelu OSI jest *warstwa prezentacji*, która przyjmuje dane od warstwy aplikacji. Podstawowym jej zadaniem jest konwersja języka. Jak pamiętamy, językiem używanym w sieci jest protokół. Jeśli dwa klienci nie używają tego samego protokołu, niezbędna jest konwersja protokołu, za którą odpowiada warstwa prezentacji.

Warstwa prezentacji jest również odpowiedzialna za zarządzanie danymi: udostępnia konwersje zestawów znaków, szyfrowanie danych i kompresję danych. Warstwa prezentacji odpowiada za przekierowywanie zadań wejścia i wyjścia.

Przekierowywanie danych to zadanie *sieciowego programu przeadresowujacego (network redirector)*, który funkcjonuje w warstwie prezentacji. Chociaz pojecie brzmi groznie, jest late do zrozumienia. Warstwa prezentacji przyjmuje pakiet danych z warstwy aplikacji i musi wybrac poprawne urzadzenie sieciowe. Jesli klient zada informacji, uzyta zostaje usluga stacji roboczej. Jesli klient dostarcza informacji, uzyta zostaje usluga serwera. Jesli zadanie adresowane jest do innego typu klienta, uzyty zostaje translator protokolu sieciowego. Na przyklad, gdy uzywamy klienta Microsoftu, aby uzyskac dostep do informacji w komputerze uniksowym, role translatora protokolu odegra SAMBA. SAMBA przekształca zadania klientów Microsoftu tak, by ich format byl zrozumialy dla systemu Unix.

Zarowno warstwa aplikacji, jak i prezentacji swiadcza uslugi sieciowe, lecz kazda z nich swiadczy inny typ uslugi, przez co pojecie *uslugi sieciowe* moze byc niejasne. Aby rozwiac watpliwosci, prosze pamietac, ze:

- ◆ Uslugi aplikacji sieciowych wywoływanie sa przez uzytkownika i komunikuja sie bezposrednio z warstwa aplikacji. List e-mail uzytkownika korzysta z uslugi aplikacji SMTP w warstwie aplikacji.
- ◆ Uslugi sieciowe nie sa wywoływanie bezposrednio przez uzytkowników, lecz sa wymagane w lacznosci sieciowej. Uslugi te sa niewidoczne dla uzytkowników. Gdy warstwa aplikacji odbiera od uzytkownika zadanie wyslania wiadomosci e-mail, wówczas warstwa ta uzywa nagłówka SMTP, aby opisac zadanie uzytkownika i przesyła calosc do warstwy prezentacji. Ta z kolei wykorzystuje usluge stacji roboczej, aby zazadac uslugi od serwera pocztowego.
- ◆ Uslugi aplikacji sieciowych sa wywoływanie przez uzytkownika i funkcjonuja w warstwie aplikacji. Uslugi sieciowe sa niewidoczne dla uzytkownika i funkcjonuja w warstwie prezentacji.

Po wybraniu odpowiedniej uslugi sieciowej nalezy ustanowic sesje.

Warstwa sesji

Piąta warstwa modelu OSI jest *warstwa sesji* — chociaz lepsza nazwa bylaby chyba „warstwa polaczenia aplikacji”. Warstwa sesji pozwala na lacznosc pomiedzy identycznymi aplikacjami dzialajacymi w dwóch różnych klientach. Osiaga sie ja przez ustanowienie wirtualnego polaczenia, opartego na nazwie uzytkownika, nazwie komputera lub poswiadczeniach sieciowych klienta.

Warstwa sesji zarządza tym wirtualnym polaczeniem, ustawiajac punkty kontrolne w odbieranych danych. Punkt kontrolny (*checkpoint*) informuje aplikacje, które dane zostaly odebrane. W przypadku zerwania polaczenia warstwa sesji analizuje punkty kontrolne i rozpoczyna transfer od ostatniego punktu kontrolnego. Zalozmy na przyklad, ze Komputer 1 odbiera 10 MB danych od Komputera 2, przy czym polaczenie zostaje przerwane przy 8 MB. Zamiast ponownej transmisji wszystkich danych, warstwa sesji szuka ostatniego punktu kontrolnego i zaczyna retransmisię od niego (w naszym przypadku — 8 MB). Poniewaz warstwa sesji zarządza lacznoscia, transfer danych jest kontynuowany, a nie ponawiany.

Z uwagi na to, iz polaczenie uzywane w warstwie sesji jest polaczeniem wirtualnym, nie daje pewnosci dostarczenia pakietu.

Warstwa transportowa

Czwarta warstwa — *transportowa* — jest odpowiedzialna za sprawdzanie poprawnosci i kontrole przepływu danych. Na poziomie tej warstwy do transmisji danych uzywane sa dwa protokoly: TCP (*Transmission Control Protocol*) oraz UDP (*User Datagram Protocol*).

W tej warstwie, jesli w roli protokolu transportowego wystepuje TCP, dostepny jest dodatkowy poziom polaczenia, który wynika z trójkierunkowego potwierdzenia (*three-way handshake*) i zapewnia dostarczenie pakietu wykorzystujac pakiety potwierdzajace. Trójkierunkowe potwierdzenie jest zestawem komunikatów powitalnych, sluzacych do ustalenia, czy nadawca i odbiorca sa gotowi do transferu danych.

Kontrola przepływu realizowana przez warstwe transportowa korzysta z rozmiaru okna TCP/IP. Rozmiar okna okresla, ile danych nadawca wysle do odbiorcy bez odbierania pakietu potwierdzajacego. Typowym rozmiarem okna jest 4096 bajtów. Warstwa transportowa jest odpowiedzialna za podzial duzych pakietów danych na mniejsze, zwykle wielkosci 1500 bajtów, lecz wartosc ta moze zostac zmieniona. Przy typowym rozmiarze okna wynoszaczym 4096 bajtów oznacza to w sumie cztery nie potwierdzone pakiety w sieci. Generalnie, po otrzymaniu przez odbiorce pakietu, do nadawcy zostaje wyslany pakiet potwierdzajacy. Po otrzymaniu tego pakietu przez nadawce kolejne pakiety danych moga byc wyslane do odbiorcy. W przypadku braku potwierdzenia pakietu moze nastapic retransmisyja, lecz to zależy od uzywanego protokolu. Podstawowa różnica pomiędzy dwoma protokolami warstwy transportowej — TCP i UDP — jest wystepowanie pakietu potwierdzajacego.

TCP

TCP dostarcza pakiety w sposob niezawodny dzieki pakietom potwierdzajacym, lecz jest wolniejszy od UDP. Przykladem aplikacji korzystajacej z TCP jest usluga FTP.

UDP

UDP nie gwarantuje dostarczenia pakietu, lecz oferuje kontrole integralnosci pakietu. Zarówno TCP, jak i UDP sprawdzaja poprawnosc odebranych pakietow. Pakiety zawierajace bledy sa odrzucane. UDP jest zazwyczaj szybszy od TCP, poniewaz przy transmisji danych wymaga transferu mniejszej liczby dodatkowych informacji. Przykladem aplikacji uzywajacej UDP jest usluga TFTP.

Gdy nadawca ustali juz, jak dane maja zostac opakowane, musi jeszcze wiedziec, dokad wyslac dane.

Warstwa sieciowa

Trzecia warstwa modelu OSI jest *warstwa sieciowa*, odpowiedzialna za adresowanie i trasowanie w sieci. Do adresowania pakietów sluzy IP (*Internet Protocol*), który podaje dla pakietów danych adresy: źródłowy (nadawcy) i docelowy (odbiorcy). Podawany jest unikatowy adres 32-bitowy, znany pod nazwą adresu IP. Adresy IP zostana omówione w dalszej czesci rozdziału.

Internet Protocol dokonuje ponadto fragmentacji pakietów i nadaje każdemu unikatowy identyfikator. Po odebraniu pakietu, Internet Protocol w warstwie sieciowej odbiorcy ponownie składa razem podzielony pakiet i przesyła dane do warstwy transportowej.

Aby ustalić najlepszą drogę do miejsca przeznaczenia, w warstwie sieciowej dokonywany jest wybór trasy (*routing*). Do powszechnie stosowanych protokołów trasowania funkcjonujących na poziomie warstwy sieciowej należą *Routing Information Protocol* (RIP), *Open Shortest Path First* (OSPF) oraz *Border Gateway Protocol* (BGP).

Mozna sobie wyobrazić warstwę sieciową jako policjanta kierującego ruchem w sieci. Warstwa sieciowa określa adresy IP nadawcy i odbiorcy oraz ustala najlepszą trasę do celu. Gdy posiadamy adres IP, trzeba ustalić adres fizyczny.

Warstwa lacza danych

Druga warstwa modelu OSI jest *warstwa lacza danych*. Jest ona podzielona na dwie podwarstwy: kontroli lacza logicznego (*Logical Link Control*) i kontroli dostępu do nosnika (*MAC — Media Access Control*).

Podwarstwa kontroli lacza logicznego jest odpowiedzialna za dodanie nagłówka i stopki. Wszystkie warstwy dodają do pakietu danych informacje nagłówka, lecz warstwa lacza danych (w podwarstwie kontroli lacza logicznego) dodaje do pakietu danych również stopkę. Zawiera ona dane cyklicznej kontroli nadmiarowej (CRC — *cyclical redundancy check*), która oblicza parzystość pakietu danych i umieszcza wynik w stopce. Po odebraniu pakietu danych przez klienta wykonywana jest operacja CRC, a jej wynik zostaje porównany z CRC nadawcy. Jeśli wyniki są zgodne, dane zostają uznane za poprawne i przekazane do następnej warstwy. Jeśli wyniki nie zgadzają się, dane, uznane za niepoprawne, są odrzucone.

Podwarstwa kontroli dostępu do nosnika (MAC) umieszcza adres fizyczny karty interfejsu sieciowego w nagłówku, który zostaje dodany do pakietu danych. Adres MAC jest unikatowa, 12-pozycyjna liczba szesnastkowa, zapisana w każdej karcie interfejsu sieciowego. Przykładowy adres MAC może wyglądać następująco: 00-80-C7-4D-B8-26.



Warstwa lacza danych powołuje się na model projektu 802. Model ten został opracowany przez IEEE (*Institute of Electrical and Electronic Engineers*) w celu ustalenia sposobu fizycznego przesyłania danych przez sieć. Projekt 802 definiuje topologie sieciowe omówione w dalszej części tego rozdziału.

Po zdefiniowaniu CRC, MAC i topologii, dane należy przetworzyć i umieścić w sieci.

Warstwa fizyczna

Pierwsza warstwa modelu OSI jest *warstwa fizyczna*, która odpowiada przede wszystkim za umieszczenie danych surowych w sieci. Dane surowe (nie przetworzone) reprezentowane są w formacie dwójkowym, czyli zbiorze jedynek i zer.

Warstwa fizyczna, inaczej zwana warstwą sprzętową, nawiązuje i utrzymuje połączenia pomiędzy nadawcą i odbiorcą. Ponieważ dane mogą istnieć w różnych postaciach (na przykład impulsów elektrycznych, fal radiowych, czy też pulsów świetlnych), warstwa fizyczna określa czas trwania każdego impulsu.

Krótko mówiąc, warstwa fizyczna definiuje sposób przylaczenia przewodu sieciowego do karty interfejsu oraz sposób sformatowania danych do transmisji.

Podzial sieci wedlug zasiegu

Sieci przyjmują różnorodne kształty i rozmiary, lecz zazwyczaj przynależą do jednej z dwóch kategorii: sieci lokalnych (LAN — *local area network*) lub sieci rozległych (WAN — *wide area network*). Pochodnymi tych dwóch typów są sieci osobiste (PAN — *personal area network*), miejskie (MAN — *metropolitan area network*) oraz obejmujące osiedla typu miasteczko akademickie (CAN — *campus area network*).

Sieci lokalne

Dowolnym zadaniem podzielonym na mniejsze części często łatwiej jest zarządzac. Jeśli wobec tego podzielimy na kawałki duża sieć, administratorzy sieci będą mieli łatwiejsze zadanie. Sieć podzielona jest na *segmenty*. LAN może składać się z wielu segmentów, połączonych razem za pomocą urządzeń sieciowego nazwanego *ruterem*. Ruter jest odpowiedzialny za łączenie segmentów sieci; zostanie omówiony bardziej szczegółowo w dalszej części książki.

Gdy połączymy ze sobą segmenty sieci za pomocą trwałego łącza fizycznego, otrzymamy sieć lokalną. LAN nie zawiera połączeń korzystających z linii telefonicznych lub dzierżawionych. Wszystkie przewody sieci lokalnej należą do niej i nie przesyłają sygnałów nie pochodzących od routerów lub klientów tej sieci. Możemy wyobrazić sobie sieć jako dziesięciopiętrowy budynek, w którym każda piętro odpowiada segmentowi sieci. Pomiedzy każdą parą sąsiadujących pięter znajdują się routery zapewniające łączność pomiędzy piętrami. Routery połączone są przewodami należącymi do sieci, nie do zewnętrznej firmy zajmującej się łącznością. Aby się komunikować między sobą nie potrzebują korzystać z łącza dedykowanego lub dzierżawionego, ponieważ wszelka łączność odbywa się wewnątrz budynku. Sieć nie zawiera linii dzierżawionych, więc uznawana jest za sieć lokalną. Gdy łączność pomiędzy dwoma obszarami sieci jest zależna od linii dzierżawionych, wówczas połączenie takie nazywane jest łączem sieci rozległej.

Sieci rozległe

Sieci rozległe (WAN) istnieją w prawie każdym środowisku sieciowym. Niemal wszystkie połączenia internetowe odbywają się poprzez łącza WAN. Łącze WAN jest nosnikiem nie należącym do sieci lokalnej, co oznacza, iż łączność wymaga usług zewnętrznych dostawców. Często jest to łącze szeregowe uzyskane od lokalnego operatora sieci telefonicznej.

Dostępnych jest wiele typów łączy WAN — na przykład, przedsiębiorstwo może zdecydować się na zakup łącza o małej przepustowości. Przepustowość (*bandwidth*) oznacza objętość danych, jaką można przesłać przez łącze w jednostce czasu. Przepustowość można porównać z wodociągiem. W określonej jednostce czasu wodociągiem może przepływać skonczona objętość wody, a jeśli potrzebujemy większego przepływu, potrzebna jest rura o większej średnicy. Podobnie jest z przepustowością łącza. Typowa

wartosc przepustowosci lacza WAN jest wielokrotnoscia 64 kb/s (kilobity na sekunde), przy czym najczesciej spotykane lacza WAN maja przepustowosc 128 kb/s, 256 kb/s, 512 kb/s oraz T1 — 1,544 Mb/s (megabit/s na sekunde).

Czynnikiem, który odróżnia nosniki sieci lokalnych od rozleglych jest fakt, iż lacza WAN nie sa trwale. Jesli lokalny operator przez pomylke nacisnie niewlasciwy przycisk, lacze WAN przestanie istniec i komunikacja zostanie ograniczona do lacznosci lokalnej. Z drugiej strony, jedynym sposobem na utrate lacznosci w sieci lokalnej jest uszkodzenie przewodu lub przerwa w dostawie pradu.

Zalozmy, ze przedsiebiorstwo (Firma B) chce nabyc lacze WAN. Lokalny Operator B dzierzawi Firmie B lacze T1 za 900 dolarów miesiecznie. Firma B posiada dwie lokalizacje funkcjonujace niezaleznie od siebie, lecz zdolne do udostepniania nawzajem danych poprzez lacze WAN. Jesli Firma B zapomni zaplacic lokalnemu Operatorowi B za usluge, lacze WAN przestanie funkcjonowac i Firmie B pozostanie lacznosc lokalna w dwóch odrebnych lokalizacjach.

Sieci lokalne nie sa zalezne od lokalnego operatora i z ich uzywaniem nie wiaza sie opłaty na rzecz firm zewnetrznych za lacza komunikacyjne. Sieci rozlegle sa zazwyczaj laczami komunikacyjnymi oplacanymi comiesiecznie i zaleza od lokalnych operatorow.

Model z projektu IEEE 802

Organizacja IEEE (*Institute of Electrical and Electronic Engineers*) opracowala standary lacznosci sieciowej, ktore rozbudowuja warstwe lacza danych i warstwe fizyczna modelu OSI. W wyniku warstwa lacza danych zostala podzielona na dwie podwarstwy, zas podzial skoncentrowal sie na karcie interfejsu sieciowego (NIC) i sposobie sformowania danych do transmisji przez siec. Wspomnialiśmy wczesniej, iż klienci musza uzywac tego samego protokolu, jesli chca nawiązac lacznosc. Nie jest to jednak jedyny wymóg. Klienci musza również uzywac takiego samego formatu danych, zdefiniowanego przez model z projektu IEEE 802.

Model z projektu IEEE 802 jest podzielony na kategorie, które definiują transfer danych do poszczególnych warstw modelu OSI. Najważniejsze kategorie to:

- ♦ 802.1 — definiuje model OSI i zarządzanie siecią.
- ♦ 802.2 — definiuje warstwę lacza danych i dzieli ją na podwarstwy kontroli lacza logicznego i MAC (kontroli dostępu do nosnika).
- ♦ 802.3 — definiuje podwarstwę MAC dla sieci Ethernet, korzystających z techniki *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD) — wykrywanie wielokrotnego dostępu do nosnika i wykrywanie kolizji. Ta kategoria jest powszechnie nazywana kategorią Ethernet. Przed transmisją danych karta sieciowa bada stan sieci i czeka na zwolnienie linii przed wysłaniem danych. Gdy dwa klienci nadają jednocześnie, zachodzi tzw. *kolizja* (ang. *collision*). 802.3 w takich przypadkach odpowiada za ponowną transmisję danych.
- ♦ 802.4 — definiuje warstwę MAC dla sieci typu Token Bus (magistrali o sztafetowym sposobie transmisji). Klient przed wysłaniem danych otrzymuje zeton, zas dane przesyłane są prostą ścieżką.

- ◆ 802.5 — definiuje warstwe MAC dla sieci Token Ring (o architekturze pierscieniowej i szafetowym sposobie transmisji). Klient przed wyslaniem danych otrzymuje zeton, zas dane przesylane sa po logicznym pierscieniu.
- ◆ 802.12 — definiuje priorytet na zadanie. Gdy do transferu danych uzyty jest model 802.12, zostaje ustalony bit priorytetu. Bit ten informuje reszte sieci, ze dany pakiet musi byc zawsze akceptowany. Gdy bit ten jest aktywny, odbiorca pakietu posiadajacego bit priorytetu na zadanie musi zaakceptowac pakiet niezaleZNIE od swojej obecnej konfiguracji. Technika bitu priorytetu na zadanie umoZLIWIWA w przeszlosci przeprowadzenie wielu niebezpiecznych ataków sieciowych, wiec zasadniczo nie jest uzywana do transferu danych.

Oprócz kategorii, nalezy również zdefiniowac topologie sieci.

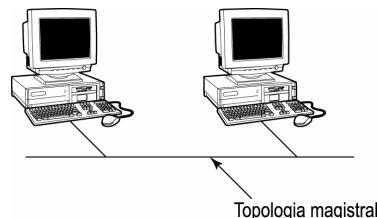
Topologie sieci

Projekt sieci okresla jej topologie, czyli trasy przesyłania danych w sieci. Która droga zostana przeslane dane od Klienta 1 do Klienta 2? Trudno jest znalezc topologie nadajaca sie do wszystkich sieci; w ponizszych punktach omówione zostana podstawowe typy topologii.

Topologia magistrali

W topologii magistrali (ang. *bus topology*), przedstawionej na rysunku 1.5, wszystkie klienty przylaczone sa do pojedynczego przewodu, zwykle kabla koncentrycznego, który pełni funkcje lacza. Jakiś czas temu topologia magistrali byla najczesciej spotykana topologia sieci. Jest latwa do zainstalowania i mozna w niej szybko wykryc usterki — to jej dwie decydujace zalety; jednakże ograniczone sa w niej dopuszczalne odleglosci i liczba klientów. Dominacja topologii magistrali trwala krótko.

Rysunek 1.5.
Liniowa topologia magistrali



Topologia magistrali zazwyczaj korzysta z kabla koncentrycznego — podobnego do stosowanych w domu — któremu IEEE nadala kategorie 802.3 10b2 (10 base 2). Sieci 10b2 zwykle przesyłaja dane z przepustowoscia 4 Mb/s na odleglosci nie przekraczajace 185 metrów. Wszystkie klienty przylaczone sa do jednego kabla przesyłajacego dane — stad wzielo sie pojecie „magistrali”. Siec magistralowa posiada punkt poczatkowy i koncowy, zakonczone opornikami o wartosci 50Ω .

Magistrala moze również posluzyc jako szkielet dla ruchu sieciowego. Siec szkieletowa jest klasyfikowana przez IEEE jako 802.3 10b5 (10 base 5). Szybkosc transmisji zwiększena jest do 10 Mb/s, a maksymalna odleglosc do 500 metrów. Podstawowa różnica pomiedzy 10b2 i 10b5 jest srednica uzywanego kabla koncentrycznego.

Do cech topologii magistrali należą:

- ◆ Łatwość instalacji i znajdowania błędów.
- ◆ Ograniczenia w odległości i przepustowości.
- ◆ Możliwość utraty magistrali — w przypadku przerwy w kablu żaden klient nie jest zdolny do komunikacji.



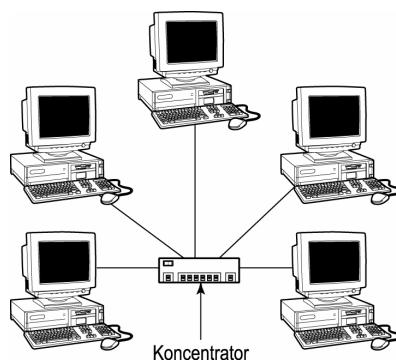
Wprawdzie ograniczenia odległości sprawiają kłopoty, lecz zasięg można zwiększyć stosując regeneratorów.

Topologia gwiazdy

Potrzeby szybszych sieci iłączenia większej liczby klientów doprowadziły do opracowania topologii gwiazdy. W tej strukturze wszystkie klienty łączą się z centralnym urządzeniem, które przyjmuje transmisje od nadawcy i przekierowuje dane do odbiorcy. Urządzeniem centralnym jest zwykły koncentrator lub przełącznik. Topologia gwiazdy została przedstawiona na rysunku 1.6.

Rysunek 1.6.

Topologie gwiazdy można rozpoznać po obecności koncentratora — centralnego urządzenia sieci



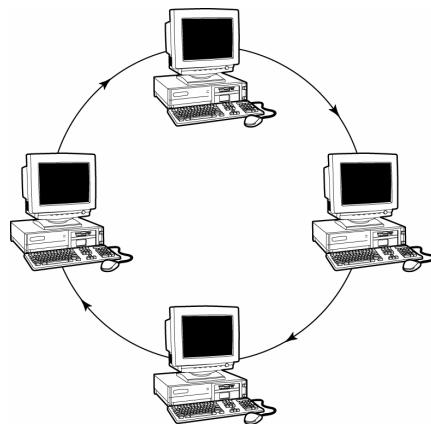
Topologia gwiazdy pozwala przesyłać dane z predkoscią do 1 Gb/s. Topologia ta została przez IEEE sklasyfikowana jako 802.3 10bT. Maksymalna odległość jest ograniczona do 100 metrów, lecz może zostać zwiększone za pomocą regeneratorów. Zarówno topologia magistrali, jak i topologia gwiazdy stosują CSMA/CD w celu dostępu do sieci.

Topologia pierscienia

Podstawowym projektantem technologii Token Ring (lierscienia z przekazywaniem żetonu), która wymusza na klientach „uprzejmie” zachowanie w sieci, była firma IBM. Aby móc umieścić dane w sieci, klient musi posiadać żeton (ang. *token*) dostępu do sieci. Żeton przekazywany jest kolejno w logicy pierscienia. W każdej chwili w sieci dostępny jest tylko jeden żeton, przez co jednocześnie tylko jeden klient może z niego skorzystać. Mechanizm ten może sprawiać wrażenie spowalniającego ruch w sieci, lecz w sieci typu Token Ring żeton może zostać przesłany wzduż pierscienia o długości 2 kilometrów 10 000 razy na sekundę. Rysunek 1.7 przedstawia topologię pierscienia.

Rysunek 1.7.

W logicznej topologii pierscienia ruch sieciowy odbywa sie po obwodzie; przekazywany jest zeton uzytkowany wspólnie przez wszystkie klienty



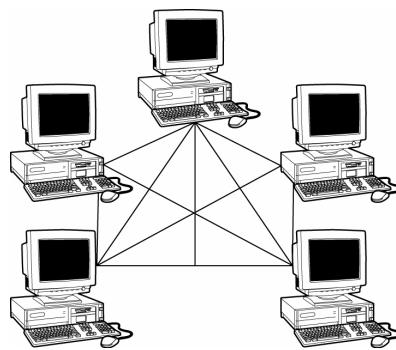
W duzych srodowiskach sieciowych stosowane sa struktury pierscienia odpornego na bledy. W przypadku wylaczenia (uszkodzenia) jednego z pierscieni, siec funkcjonuje dalej korzystajac z drugiego pierscienia. Klienty lacza sie zazwyczaj za pomoca specjalnego koncentratora o nazwie MSAU (*Multi Station Access Unit* — jednostka dostepu do stacji wielotermalowej). Najczesciej spotykana obecnie forma topologii piersciennowej jest pierscien swiatlowodowy (*Fiber-Optic Ring*), zwykle uzywany w funkcji sieci szkieletowej.

Topologia oczkowa

Najbardziej odporna na uszkodzenia struktura sieci jest topologia oczkowa (ang. *Mesh*), przedstawiona na rysunku 1.8. W tej topologii jedyna mozlioscia zakamania pracy sieci jest kleska zywiolowa. Topologie oczkowe stosowane sa zasadniczo w bardzo malych sieciach z uwagi na wymogi sprzętowe. Po zainstalowaniu, nadmiarowe transmisje w tej topologii sa znikome. Dzieki nadmiarowosci w kazdym kliencie, dopuszczalna jest awaria kilku składników.

Rysunek 1.8.

W topologii oczkowej kazdy klient polaczony jest z wszystkimi pozostalymi

**Topologie hybrydowe**

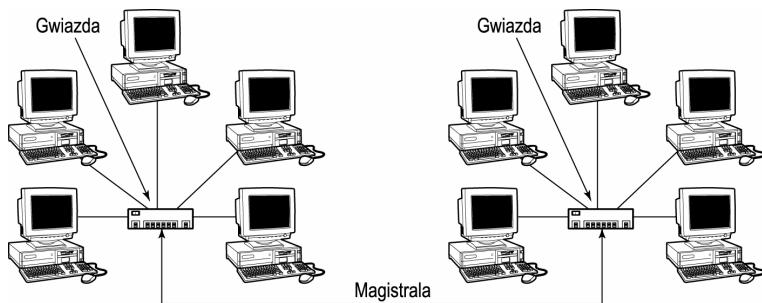
Bardzo rzadko trafiaja sie srodowiska, w których stosuje sie jeden rodzaj topologii. Czesto potrzeby organizacji wymuszaja uzycie kilku różnych typów topologii. Topologia hybrydowa czesto pozwala na niezalezne funkcjonowanie calych obszarów sieci nawet

w przypadku laczacej te obszary sieci szkieletowej. Ponisze podpunkty opisuja dwa typy topologii hybrydowych: gwiazda-magistrala oraz gwiazda-pierscien.

Gwiazda-magistrala

Zbiór sieci o topologii gwiazdzistej, polaczonych magistrala, daje w wyniku topologie hybrydowa gwiazda-magistrala. Chociaz nie daje ona odpornosci na uszkodzenia, nie posiada również pojedynczego punktu awarii. Topologie hybrydowa gwiazda-magistrala przedstawia rysunek 1.9.

Rysunek 1.9.
Topologia gwiazda-magistrala zlozona z dwóch odrebrnych segmentów o topologii gwiazdy, polaczonych magistrala



Jak widac, magistrala laczy dwie sieci o topologii gwiazdy. Jesli siec magistralowa zawiadzie, sieci o topologii gwiazdy moga funkcjonowac niezaleznie. Jesli zawiadzie jeden z koncentratorów, druga siec o topologii gwiazdy bedzie funkcjonowac dalej. Głównymi zaletami topologii gwiazda-magistrala jest prosty projekt i proste rozwiazyanie problemów.

Gwiazda-pierscien

Zbiór sieci o topologii gwiazdy polaczonych pierscieniem daje w wyniku topologie hybrydowa gwiazda-pierscien, która nalezy do najczesciej stosowanych topologii hybrydowych. Ma przewage nad topologią gwiazda-magistrala, gdyż jest odporna na uszkodzenia.

W tym rozwiazaniu topologia pierscienia zawiera pierscien podstawowy i zapasowy. Jesli którys z nich zostanie uszkodzony, siec funkcjonuje nadal. Jesli zawiadzie jedna z sieci gwiazdzistych, reszta sieci również funkcjonuje dalej.

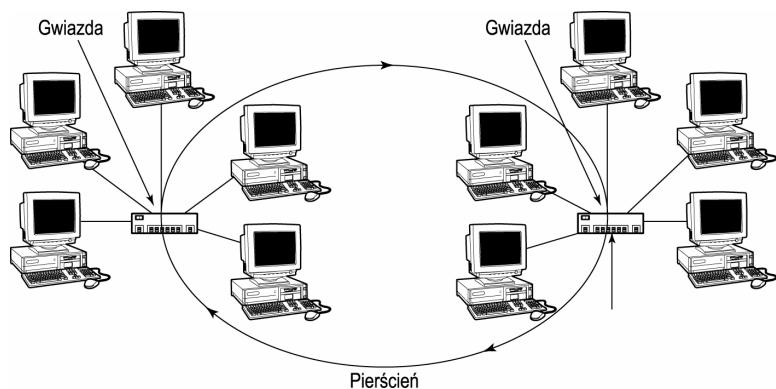
W typowej topologii hybrydowej gwiazda-pierscien sieci gwiazdziste sa typu Ethernet, zas pierscien jest swiatlowodowy. Taki projekt zwiększa szybkosc przesyłania danych pomiedzy sieciami gwiazdzistymi. Rysunek 1.10 przedstawia topologie hybrydowa gwiazda-pierscien.

Infrastruktura sieciowa

Dotychczas zdefiniowalismy komunikacje sieciowa i objasniliśmy zasady teoretyczne. Biezacy podrozdzial zajmuje sie infrastruktura sieci i składnikami sprzętowymi potrzebnymi do utrzymywania lacznosci w sieci. Składniki te zostana przedstawione według kolejnosci wystepowania warstw funkcjonalnych OSI, zaczynajac od warstwy 1.

Rysunek 1.10.

Topologia gwiazda-pierscien zlozona z dwóch odrebrnych segmentów o topologii gwiazdy, polaczonych piersciem



Regeneratory

Wszystkie topologie sieciowe maja ograniczenia odleglosci. Niektóre z nich uzywaja technologii 10bT o ograniczeniu do 100 metrów, inne 10bF (swiatlowodów) o teoretycznym ograniczeniu do 2000 metrów.

Ograniczenie odleglosci czesto utrudnia dzialanie sieci. Sposobem na rozwiazanie problemu moze byc zastosowanie *regeneratora* (ang. *repeater*), funkcjonujacego w warstwie fizycznej modelu OSI. Regenerator sluzi do wzmacnienia sygnalu, gdyz jego osłabienie na skutek tlumienia przewodu moze spowodowac uszkodzenie danych i utrate pakietów. Regenerator wzmacnia jedynie elektrycznie sygnal w przewodzie. Niepisana regula mówi, by umieszczac regeneratory 15 metrów przed punktem oddalonym na maksymalna odleglosc dla danej topologii. Jesli siec zbudowana jest na podstawie 10b2, regenerator powinien zostac umieszczony w okolicy 170 metra przewodu.

Karta interfejsu sieciowego

Karta interfejsu sieciowego (NIC — *Network Interface Card*) funkcjonuje zarówno w warstwie fizycznej, jak i w warstwie lacza danych modelu OSI. NIC uzywa adresu sprzetowego MAC z warstwy lacza danych oraz topologii warstwy fizycznej.

Koncentrator

W sieci gwiazdzistej koncentrator jest centralnym miejscem podlaczenia wszystkich klientów. Koncentrator dziala w warstwie lacza danych modelu OSI i „interesuja” go jedynie adresy MAC. Koncentrator nie sluzi do tworzenia dodatkowych segmentów sieci; sluzi jedynie jako miejsce podlaczenia.

Siec mozemy rozbudowac, laczac koncentratory kablem skrzyzowanym. Przy laczeniu koncentratorów czesto stosowane sa regeneratory, aby zwięksyc odleglosc pomiedzy koncentratorami. Przepustowosc koncentratora liczona jest w sposób zbiorowy. Jesli przepustowosc znamionowa koncentratora wynosi 100 Mb/s, oznacza to sume wszystkich równoczesnych przesyłów danych przez koncentrator.

Przelacznik

Przelacznik również jest obecny w warstwie lacza danych i przypomina koncentrator, gdyz laczy ze soba klienty w punkcie centralnym. Funkcjonowanie przelacznika opiera sie na adresach MAC, jednakze przelacznik uzywa tych adresów do segmentacji sieci. Utworzone za pomoca przelacznika segmenty nosza nazwe *wirtualnych sieci lokalnych* (ang. *Virtual LAN*). Oprócz zdolnosci do wirtualnej segmentacji sieci, przelacznik udogodnia maksymalna przepustowosc na kazdym porcie.

Jesli przepustowosc znamionowa przelacznika wynosi 100 Mb/s, kazdy klient moze potencjalnie komunikowac sie z szybkoscia 100 Mb/s. Koncentrator i dobry przelacznik różnia sie przede wszystkim cena.

Most

Most sieciowy (ang. *bridge*) funkcjonuje podobnie do mostu laczacego dwa odrebrene obszary ladowe. Most sieciowy po prostu laczy różne typy sieci i funkcjonuje w warstwie lacza danych modelu OSI, sluzac do translacji topologii.

Jak juz wspomniano w tym rozdziale, klienty, aby sie komunikowac, musza uzywac tego samego protokolu i tej samej topologii. Gdy klienty podlaczone sa do sieci o różnych topologiach, lecz uzywaja wspólnego protokolu, do ich polaczenia moze posluzyc most, który „rozumie” obie topologie i dokonuje pomiedzy nimi translacji.

Mostu mozna tez uzyc do ograniczenia propagacji *ruchu sieciowego rozwloszen*. Sa to transmisijsie sieciowe wysylane do wszystkich klientow w sieci, które sa wrogiem numer jeden kazdego administratora sieciowego.

Most nie analizuje adresów sieciowych i nie zajmuje sie nimi. Uznaje sie, iz odczytanie adresu sieciowego przekracza mozliwosc mostu. Most „przeszkolony” w zakresie adresów sieciowych zostaje ruterem.

Ruter

Ruter (ang. *router*), który funkcjonuje w warstwie sieciowej, kieruje ruchem sieciowym wszystkich klientow. Poniewaz ruter zna polozenie innych sieci, mo ze skierowac ruch sieciowy do odpowiedniej lokalizacji.

Kazdy segment sieci musi byc w stanie komunikowac sie z innymi segmentami, co jest możliwe dzieki uzytku rutera. Ruter kieruje ruchem, lecz nie dokonuje translacji. Jego zakres dzialania ograniczony jest do warstwy 3. Rutery nie zajmuja sie adresami sprzetowymi MAC klientow — to nalezy do urzadzen z warstwy 2.

Bruter

Bruter (*bridge/router*) jest urzadzeniem zdolnym do trasowania i translacji, laczacy zalety mostu i rutera.

Bruter funkcjonuje w warstwach 2. i 3. modelu OSI. Jego typowe zastosowania obejmuja srodowiska, w których znajdują się segmenty o różnych topologiach i różnych adresach sieciowych (które zostaną omówione w dalszej części rozdziału). Bruter tłumaczy topologie i trasuje pakiety do miejsca przeznaczenia.

Wiekszosć współczesnych bruterów potrafi funkcjonować w roli mostów.

Brama

Brama sieciowa (ang. *network gateway*) nie spełnia tej samej funkcji, co brama domyslna (ang. *default gateway*). Adres bramy domyslnej oznacza adres routera. Brama sieciowa służy do tłumaczenia protokołów i może też posłużyć do tłumaczenia adresów pomiędzy różnymi protokołami.

Bramy mogą pracować we wszystkich siedmiu warstwach, lecz najczęściej spotyka się bramy funkcjonujące w warstwie 4. i wyższych. W tych warstwach brama sieciowa może przyjmować dane od klientów używających TCP/IP oraz IPX/SPX (używany przez Novellę) protokół *Internet Packet Exchange/Sequence Packet Exchange*) i tłumaczyć te protokoły tak, by klienci mogli się ze sobą komunikować.

Wprowadzenie do TCP/IP

Internet powstał ponad 40 lat temu w Department of Defense/Advanced Research Project Agency (DOD/ARPA) USA z myślą o ogólnokrajowym systemie łączności, który pozwoliłby komputerom w kraju i na całym świecie wysyłać i odbierać informacje. Z technicznego punktu widzenia rząd USA nie zamierzał stworzyć tego, co obecnie nazywamy Internetem; chodziło po prostu o obronny system łączności. Mimo to prawie trzydzieści lat, zanim utworzono rade zarządzającą nadzorującą rozwój Internetu.

Internet Architecture Board (IAB) jest rada nadzorcza kierująca opracowywaniem standardów internetowych. IAB obejmuje dwie istotne grupy: Internet Research Task Force (IRTF) oraz Internet Engineering Task Force (IETF):

- ♦ *IETF* — koncentruje się przede wszystkim na krótkoterminowych problemach technicznych. IETF dzieli się na około dziesięć grup, które wspólnie tworzą Internet Engineering Steering Group (IESG).
- ♦ *IRTF* — koncentruje się na długofalowych strategiach technicznych Internetu. Jako odpowiednik IETF, IRTF dzieli się na grupy tworzące Internet Research Steering Group (IRSG).

Request for Comments

Projekt DOD/ARPA miał skromne początki, lecz w następnych latach włączały się do pracy nad nim kolejne agencje. Na samym początku został stworzony proces zapytan o komentarze (RFC — *Request for Comments*; wcześniej służył do opracowania standardów TCP/IP, natomiast obecnie służy do tworzenia standardów internetowych. Proces RFC opiera się na recenzjach i segregacji na kategorie. Nowe dokumenty RFC sa

przedstawiane do oceny IETF i oceniane przez grupę redaktorów RFC. Po ocenie zaaproponowanego RFC, dokument zostaje zaszeregowany przez IETF do kategorii określającej, jak z danego dokumentu należy korzystać.

RFC są numerowane i każdy może zgłosić propozycję dokumentu. Reguły zgłoszania propozycji opisane zostały w RFC 1543. Najwięcej nieporozumień związanych z RFC dotyczy ich zawartości. Dokumenty RFC obejmują cały pakiet łączności TCP/IP, nie tylko protokole. Protokole, topologie, narzędzia i standardy to tylko część zagadnień objętych procesem RFC.



Dokument RFC po opublikowaniu nie podlega już żadnym modyfikacjom. Wszelkie korekty i zastąpienia publikowane są jako nowe RFC. Nowy dokument RFC korygujący lub zastępujący dokument, który już istnieje, według stosowanej terminologii aktualizuje (*update*) lub wycofuje (*obsolete*) starszą wersję.

Część RFC to dokumenty informacyjne, zas inne opisują protokole internetowe. IAB utrzymuje listę RFC opisujących pakiet protokołów. Do każdego z dokumentów przypisany jest status.

Protokół internetowy może posiadać jeden z następujących statusów:

- ♦ *Standard* — gdy IAB otrzymuje dokument opisujący ewentualny nowy standard lub modyfikacje RFC, propozycja oceniana jest przez ekspertów technicznych, grupy robocze lub redaktora RFC. IAB następnie przyznaje klasyfikację, aby ustalić, czy dokument ma być uznany za standard.
- ♦ *Wymagany* (ang. *Required*) — protokole wymagane muszą być implementowane przez wszystkie systemy.
- ♦ *Zalecany* (ang. *Recommended*) — protokole zalecane powinny być implementowane przez wszystkie systemy.
- ♦ *Fakultatywny* (ang. *Elective*) — wszystkie systemy mogą implementować protokół fakultatywny, lecz nie muszą tego robić. Jest to zwykła oznaka, iż dany protokół na etapie „dojrzałania” cieszy się ograniczonym zainteresowaniem.
- ♦ *Do ograniczonego zastosowania* (ang. *Limited use*) — te protokole przeznaczone są do stosowania w niektórych okolicznościach, na przykład z uwagi na wyspecjalizowany charakter, ograniczona funkcjonalność lub pozostawanie na etapie eksperymentalnym albo historycznym.
- ♦ *Nie zalecany* (ang. *Not recommended*) — te protokole nie są zalecane do powszechnego użytku — np. z uwagi na ograniczoną funkcjonalność, wyspecjalizowany charakter lub pozostawanie na etapie eksperymentalnym albo historycznym.

Jeżeli IAB ustali, iż protokół ma możliwość stanie się standardem, wówczas przechodzi przez kilka etapów rozwoju, testowania i akceptacji:

- ♦ *Propozycja standardu* (ang. *Proposed standard*) — te protokole mogą być rozważane przez IAB jako potencjalne standardy w przyszłości. Ocena ta zwykle wskazuje, iż specyfikacja jest ogólnie stabilna i dobrze zrozumiała. Pozadane są implementacje i testowanie przez kilka grup.

- ◆ *Szkic standardu* (ang. *Draft standard*) — IAB aktywnie rozwaza przyjecie tego protokolu jako ewentualnego standardu. Pozadane jest szeroko pojete testowanie. Komentarze i wyniki testów powinny byc zgłoszane do IAB. Istnieje mozliosc wprowadzenia zmian w projekcie protokolu, zanim stanie sie standardem.
- ◆ *Standard internetowy* (ang. *Internet standard*) — oznacza, iz proponowany RFC uzyskal aprobatę w roli aktywnego standardu do przyjecia przez spolecznosc internetowa. Uznanie za standard internetowy oznacza, iz protokol przeszedl rygorystyczne testy i osiągnal wysoki poziom dojrzalosci technicznej.

RFC po zaklasifikowaniu do roli standardu otrzymuje unikatowy numer. Pelna liste dokumentow RFC mozna pobrac pod adresem <http://www.rfc-editor.org>.

Model odniesienia TCP/IP

TCP/IP nie jest pojedynczym protokołem, lecz pakietem protokołów. Z uwagi na to TCP/IP nie korzysta bezpośrednio z modelu OSI, stosujac zamiast tego czterowarstwowy model lacznosci, przedstawiony na rysunku 1.11.

Rysunek 1.11.

*Modele odniesienia
OSI i TCP/IP
nie odpowiadaja sobie
w stosunku jeden do
jednego. Kazda
warstwa modelu
TCP/IP
jest odwzorowana
na jedna lub wiecej
warstw modelu OSI*

Model OSI	Pakiet protokołów
	TCP/IP
Warstwa aplikacji	Warstwa aplikacji:
Warstwa prezentacji	Telnet, FTP i inne
Warstwa sesji	Warstwa transportowa: TCP
Warstwa transportowa	Warstwa sieciowa:
Warstwa sieciowa	Warstwa łącza danych: IP, ARP, ICMP
Warstwa łącza danych	Warstwa fizyczna
Warstwa fizyczna	

Warstwa aplikacji

Czwarta warstwa modelu odniesienia TCP/IP jest *warstwa aplikacji*, odpowiedzialna za aplikacje TCP/IP. W tej warstwie funkcjonuja dwa typy aplikacji: oparte na gniazdach (ang. *socket*) oraz aplikacje NetBIOS (ang. *Network Basic Input Output System*).

Aplikacje oparte na gniazdach istnieja we wszystkich klientach uzywajacych TCP/IP. Dla takich aplikacji wymagane sa trzy elementy: adres IP, port i typ uslugi. Jak juz wspomniano, kazdy klient uzywajacy TCP/IP posiada unikatowy adres 32-bitowy. Kazdy adres posiada 65 536 punktów wejsciowych zwanych *portami*. Aplikacje TCP/IP dzialaja na określonych portach (najczesciej spotykane aplikacje TCP/IP zostana omówione w dalszej czesci tego podrozdzialu, laczenie z portami, których uzywaja do lacznosci).

Aplikacje NetBIOS sa powszechnie spotykane w systemach operacyjnych Microsoftu. NetBIOS jest jednym z najgorzej rozumianych aspektów sieci tego producenta. Najczesciej spotykanym nieporozumieniem jest zalozenie, iz nazwa NetBIOS oznacza nazwe komputera. Dopiero nazwa komputera ze wskaznikiem uslugi tworzy nazwe NetBIOS. Sieci oparte na NetBIOS-ie nie tylko generuja niepowazne ilosci ruchu sieciowego, lecz również wymagaja ogromnych nakladow pracy administracyjnej. Kolejnym nieporozumieniem, dotyczacym NetBIOS-u, jest uznanie go za protokol. NetBIOS jest narze-

dziem transportu w warstwie sesji, zapewniającym wirtualną linię z aplikacjami w różnych klientach. Oznacza to, iż aplikacje sprawiają wrażenie zdolnych do komunikacji na podstawie samych tylko nazw komputerów. Wskaznik usługi w nazwie NetBIOS posiada format szesnastkowy. Najczęściej spotykane wskazniki usług NetBIOS w sieciach Microsoftu to:

- ◆ *Nazwa_komputera[00h]* oznaczający usługę Stacja robocza
- ◆ *Nazwa_komputera[03h]* oznaczający usługę Messenger
- ◆ *Nazwa_komputera[20h]* oznaczający usługę Serwer

Podstawowa wada aplikacji NetBIOS jest liczba rozgłoszeń wysyłanych w sieci w celu ogłaszenia usług i przeglądania.

Warstwa transportowa

Trzecia warstwa modelu TCP/IP jest *warstwa transportowa*. W warstwie transportowej używane są dwa protokole: TCP (ang. *Transmission Control Protocol* — protokół kontroli transmisji) oraz UDP (ang. *User Datagram Protocol* — protokół datagramów użytkownika). TCP jest protokołem zorientowanym na połączenie i wiarygodnym, lecz wolniejszym w transmisji. UDP jest protokołem bezpołączniowym, bez gwarancji dostawy, szybszym w transmisji.

Gdy aplikacja używa do linii TCP/IP, uruchamiany jest mechanizm potwierdzenia trójstronnego (ang. *three-way handshake*), które zapewnia dostarczenie pakietów bez błędów, we właściwej kolejności i bez utraty lub powielania danych. Rozmiar okna TCP/IP jest definiowany w warstwie transportowej za pomocą TCP. TCP gwarantuje dostarczenie pakietu, lecz transfer jest wolniejszy.

Aplikacja korzystająca z UDP nie stosuje trójstronnego potwierdzenia i nie oferuje gwarancji dostarczenia pakietu. W zasadzie UDP wysyła dane do klienta-odbiorcy z nadzieją, iż zostaną one odebrane. Nie jest stosowana linia uzupełniająca dla retransmisji danych. Protokół UDP jest o wiele szybszy, lecz nie gwarantuje dostawy.

Klient nie ma możliwości wyboru UDP lub TCP. Decyzja podejmowana jest przez twórcę aplikacji w trakcie jej pisania.

Warstwa internetowa

Druga warstwa modelu TCP/IP jest *warstwa internetowa*, która funkcjonuje bardzo podobnie do warstwy sieciowej modelu OSI. Warstwa internetowa jest przede wszystkim odpowiedzialna za adresowanie i trasowanie w sieci, a ponadto za fragmentację pakietów. W tej warstwie pakietów są składane i dzielone na potrzeby transmisji.

W warstwie internetowej działa kilka protokołów, z których najczęściej spotykane to:

- ◆ *Internet Protocol (IP)* — bezpołączniowy protokół, który zapewnia adresowanie i wybór trasy. Informacje nagłówka dodanego do pakietu danych obejmują adresy źródłowy i docelowy; na podstawie tych adresów wybierana jest trasa. IP dokonuje ponadto łączenia i podziału pakietów, czasem nazywanego fragmentacją,

dla warstwy interfejsu sieciowego. IP pomaga takze kontrolowac ruch przechodzacy przez rutery, korygujac w pakietach wartosc czasu zycia (TTL — *time to live*) podczas ich przechodzenia przez ruter. TTL ustala, jak dlugo pakiet moze przebywac w sieci. Przy kazdym przejsciu pakietu przez ruter TTL zmniejszany jest o 1, a gdy wartosc TTL spadnie do zera, pakiet zostaje odrzucony.

- ◆ *Internet Control Message Protocol (ICMP)* — najczesciej uzywany z narzedziem PING (ang. *Packet Internet Groper*). PING najczesciej sluzy do rozwiązywania problemów z polaczeniami. ICMP jest wykorzystywany do wysylania pakietów tlumienia źródła rutera, które powiadamiaja klienty o zbyt szybkim nadchodeniu dużego ruchu sieciowego i zagrożeniu wypadaniem pakietów. Bardziej zaawansowanym zastosowaniem ICMP jest zabieganie o rutery. Klienty moga stosowac ICMP Router Discovery Protocol do lokalizowania rutерów w sieci.
- ◆ *Address Resolution Protocol (ARP)* — sluzy do rozwiązywania adresów IP na adresy MAC. Gdy adres MAC jest juz znany, pakiety moga byc przesyłane bezposrednio od nadawcy do odbiorcy, o ile oba klienty znajdują sie w tym samym segmencie. Jesli klienty znajdują sie w różnych segmentach, pakiet zostaje wysłany do rutera.
- ◆ *Internet Group Management Protocol (IGMP)* — czasem nazywany *Internet Group Messaging/Membership Protocol*; sluzy do identyfikacji członków grupy, która przyjmuje pakiety grupowe (ang. *multicast packet*). Pakiet grupowy wysyłany jest do grupy klientów, zamiast do wszystkich (jak dzieje się w przypadku rozmów). Unicast oznacza wysyłanie pakietu tylko do jednego klienta. IGMP ma wiele zastosowań w sieci, lecz do najczęstszych naleza wideokonferencje, pogadki internetowe i dynamiczne aktualizacje ruterów.

Warstwa interfejsu sieciowego

Pierwsza warstwa modelu TCP/IP — *warstwa interfejsu sieciowego* — odpowiada warstwom lacza danych i fizycznej modelu OSI i realizuje dostep do sieci. Warstwa interfejsu sieciowego komunikuje sie bezposrednio z siecią — jest posrednikiem pomiedzy topologią sieci a warstwą internetową.

Przegląd adresowania IP

Kazdy wezel w sieci TCP/IP musi posiadac unikatowy adres 32-bitowy. Adres IP jest bardzo podobny do adresu domowego lub biurowego. Adres domowy wyszczególnia kraj, stan (województwo), miasto, ulice i lokalizacje przy ulicy. Adres IP identyfikuje wezel poprzez adres sieci, adres podsieci i adres wezla.

Siec opisana jest przez adres sieci. Segment sieci nazywany jest podsiecią i opisuje go adres podsieci. Kazdy składnik segmentu określany jest mianem wezla (ang. *node*) i opisyany jest przez adres wezla.

Laczność z wykorzystaniem TCP/IP wymaga podania dwóch parametrów: adresu IP i maski podsieci. Maska podsieci zostanie omówiona bardziej szczegółowo w dalszych rozdziałach; jednakże ogólne zrozumienie tego pojęcia jest wymagane, by przyswoić

sobie informacje zawarte w bieżącym rozdziale. Duża sieć może zostać podzielona na podsieci poprzez manipulowanie maską podsieci. Zmiana maski podsieci powoduje zmianę liczby podsieci w sieci i liczby wezłów w każdej podsieci.

Adresy IP składają się z czterech części, nazywanych oktetami, ponieważ każda część ma osiem bitów. Cztery części po osiem bitów dają w sumie 32-bitowy adres. Pierwsza część adresu IP zawsze identyfikuje klasę sieci. Istnieją cztery klasy adresów (trzy z nich przedstawia tabela 1.1), zasada każdej klasy posiada odmienną liczbę adresów, jakie można w niej przydzielić:

- ◆ Sieci klasy A mają w pierwszym oktacie wartości od 1 do 126. Sieci klasy A używają pierwszego oktetu jedynie do identyfikacji adresu sieci. Poczta USA (U.S. Postal Service) otrzymała sieć 56 — w tym przypadku adres sieci to 56.0.0.0. W sieciach klasy A stosowana jest domyslna maska podsieci 255.0.0.0.
- ◆ Sieci klasy B identyfikują w pierwszym oktacie wartości z zakresu od 128 do 191. W sieciach klasy B dwa pierwsze oktety służą do identyfikacji adresu sieciowego. Na przykład, linie lotnicze Delta Air Lines posiadają wewnętrzna sieć o adresie 172.16.0.0. Domyslna maska podsieci dla klasy B jest 255.255.0.0.
- ◆ Sieci klasy C identyfikują w pierwszym oktacie wartości z zakresu od 192 do 223. W sieciach klasy C do identyfikacji adresu sieciowego służą trzy pierwsze oktety. Na przykład, firma Northwest Computer Training przyznana została sieć 216.18.17.0. Domyslna maska podsieci dla klasy C jest 255.255.255.0.
- ◆ Sieci klasy D w pierwszym oktacie mają wartości z zakresu od 224 do 239. Sieci te stosowane są jedynie do adresowania grupowego i stosują domyslną maskę podsieci 255.255.255.255.
- ◆ Sieci klasy E w pierwszym oktacie mają wartości z zakresu od 240 do 255. Sieci klasy E zarządzane są do przyszłych zastosowań.

Gdy sieć stosuje maskę podsieci domyslną dla swojej klasy, oznacza to, że nie jest podzielona na podsieci. Tabela 1.1 opisuje domyslne właściwości rutownalnych klas adresów.

Tabela 1.1. Domyslne właściwości rutownalnych klas adresów

Klasa	Zakres pierwszego oktetu	Domyslna maska podsieci	Liczba wezłów
A	1-126	255.0.0.0	16 777 214
B	128-191	255.255.0.0	65 534
C	192-223	255.255.255.0	254

Aplikacje TCP/IP

Warstwa aplikacji modelu TCP/IP oferuje wiele aplikacji służących do łączności sieciowej i niemal niemożliwe byłoby wymienić je wszystkie. Do najczęściej stosowanych aplikacji należą:

- ◆ *Domain Name System (DNS)* — sluzy do rozwiazywania nazw na adresy IP. DNS uruchomiony jest na porcie 53. Zanim polaczmy sie z witryna WWW, jej adres musi zostac rozwiazany na adres IP. Usluge te swiadczy DNS.
- ◆ *File Transfer Protocol (FTP)* — sluzy do pobierania i wysylania plikow na zdalne komputery. FTP uzywa portu 21. dla serwera i portu 20. dla klienta.
- ◆ *Dynamic Host Configuration Protocol (DHCP)* — sluzy do dynamicznego przydzielania klientom adresow IP z centralnego serwera. DHCP korzysta z portu 67. dla serwera i 68. dla klienta.
- ◆ *Simple Mail Transport Protocol (SMTP)* — sluzy do przesyłania poczty elektronicznej. SMTP korzysta z portu 25.
- ◆ *Post Office Protocol (POP3)* — sluzy do odbierania poczty elektronicznej. POP uzywa portu 110.
- ◆ *Telnet* — emulacja terminala sluzaca do uruchamiania polecen na zdalnych komputerach, korzystajaca z portu 23.
- ◆ *Hyper Text Transfer Protocol (HTTP)* — sluzy do zadania uslug dzialajacych na porcie 80. HTTP jest wykorzystywany do dostepu do stron WWW.
- ◆ *Secure Sockets Layer (SSL)* — sluzy do dokonywania bezpiecznych transakcji danych pomiedzy klientami i serwerami. SSL uzywa portu 443.
- ◆ *Network Basic Input-Output System (NetBIOS)* — sluzy do rozwiazywania nazw, przedw wszystkim nazw komputerow w Microsoft Network. NetBIOS wykorzystuje porty 137., 138. i 139.

Rozdział 2.

Architektura protokolu TCP/IP

W tym rozdziale:

- ◆ Pieciowarstwowa architektura TCP/IP
- ◆ Lacznosc pomiedzy warstwami

To, co znamy obecnie pod nazwa „Internet”, zaistniało w roku 1968 jako projekt sponsorowany przez Departament Obrony (*Department of Defense*) rządu USA. Projekt ten usiłował połączyć różne centra badawcze wspierane przez Departament Obrony siecią o nazwie ARPANET (*Advanced Research Projects Agency Network*). Na początku funkcje standardowego protokołu połączeniowego pełniły *Network Control Protocol* (NCP). Jednak protokół ten okazał się niewystarczający dla sieci ARPANET, której rozmiary rosły w olbrzymim tempie, wobec tego w roku 1974 opracowany został TCP/IP. Nazwa TCP/IP (*Transmission Control Protocol and Internet Protocol*) w rzeczywistości odnosi się do dwóch protokołów, z których żaden nie jest używany samodzielnie. Tworzą one *pakiet protokołów* (ang. *protocol suite*), co oznacza hierarchiczny zbiór powiązanych protokołów. Z uwagi na rewolucyjną rolę, jaką TCP oraz IP odegrali w rozwoju sieci komputerowych, cały pakiet nosi nazwę pakietu protokołów TCP/IP.



Historia TCP/IP została opisana w rozdziale 1.

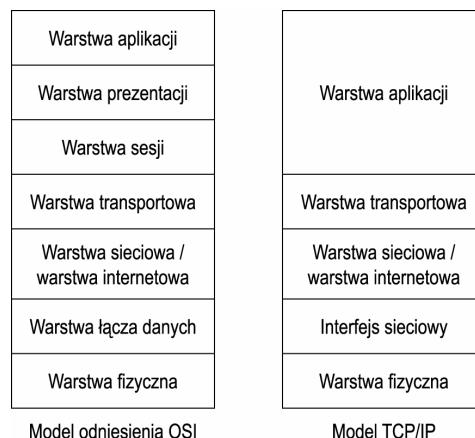
W niniejszym rozdziale poznamy pięć warstw, składających się na architekturę TCP/IP: fizyczna, sieciowa, internetowa, transportowa i aplikacji. Czytelnik zapozna się z rolą, jaką te warstwy odgrywają w pomyslnym przesyłaniu danych z jednego komputera do drugiego. Przedstawimy również proces komunikacji pomiędzy warstwami.

Pieciowarstwowa architektura TCP/IP

W ciągu ostatniej dekady wielu producentów sprzętu i oprogramowania dołączyło do swej oferty produkty pracujące w sieciach komputerowych. Aby uniknąć niezgodności pomiędzy różnymi produktami sieciowymi wprowadzonymi na rynek, opracowane zostały standardy otwartych systemów komputerowych (ang. *open computing*). Rozwój TCP/IP od zawsze odbywał się w środowisku otwartym, wobec tego TCP/IP nadal

uznawany jest za prawdziwy protokół polaczeniowy systemów otwartych, pomimo prób popularyzacji przez rząd USA protokółów *Open Systems Interconnection* (OSI). Z upływem lat, w odpowiedzi na istniejący siedmiowarstwowy model odniesienia OSI, rozwinał się współczesny pieciowarstwowy model architektury TCP/IP. Podstawowym zadaniem tego modelu jest zdefiniowanie zbioru otwartych standardów dla wszelkich obecnych lub przyszłych zmian rozwojowych w dziedzinie TCP/IP. Rysunek 2.1 przedstawia poglądowe porównanie modeli odniesienia OSI oraz TCP/IP.

Rysunek 2.1.
*Modele odniesienia
OSI i TCP/IP
— porównanie*



Czasami można natknąć się na czterowarstwowy model architektury TCP/IP. W uproszczonej wersji dwie pierwsze warstwy — fizyczna i interfejsu sieciowego — zostały połączone w jedną, nazywaną warstwą dostępu do sieci (*Network layer*) lub po prostu warstwą fizyczną (*Physical layer*). Zdarza się również przypadki, gdy warstwa internetowa nazywana jest warstwą sieciową (*Network layer*).

Model odniesienia pełni funkcje wytycznych funkcjonalnych w podziale procesów i zadań łączności sieciowej:

- ◆ pozwala producentom tworzyć produkty zgodne z pozostałymi,
- ◆ ułatwia zrozumienie złożonych operacji,
- ◆ dzieli na kategorie technologie sieciowe i implementacje ich protokołów, co pozwala na wyspecjalizowane tworzenie projektów funkcji modułowych.

Podobnie jak model odniesienia OSI, model architektury TCP/IP składa się ze zbioru warstw, z których każda reprezentuje grupę określonych zadań i aspektów procesu łączności. Ponieważ model TCP/IP jest teoretyczny, warstwy te nie istnieją fizycznie, ani nie wykonują w rzeczywistości żadnych funkcji. Dopiero implementacje protokołu, stanowiące połączenie sprzętu i oprogramowania, wykonują funkcje przypisane do odpowiadających im warstw. Model TCP/IP składa się z następujących pięciu warstw:

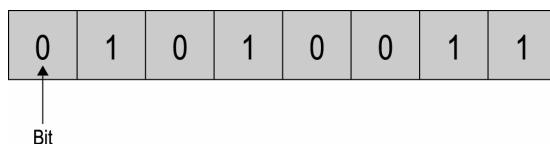
- ◆ *Warstwa fizyczna* — udostępnia noszarkę fizyczną (np. przewody), służącą do transmisji danych z jednego komputera do drugiego.
- ◆ *Warstwa interfejsu sieciowego* — odpowiada za identyfikację urządzeń w sieci w celu kontroli przepływu danych, na podstawie ich adresów sieciowych, oraz za organizację bitów z warstwy fizycznej w ramki.

- ♦ *Warstwa internetowa* (inaczej *miedzysieciowa*) — odpowiada za przesyłanie (trasowanie) danych pomiędzy różnymi sieciami.
- ♦ *Warstwa transportowa* — odpowiada za organizację w segmenty komunikatów odebranych z wyższych warstw, za kontrolę błędów oraz za kontrolę przepływu między dwoma punktami końcowymi.
- ♦ *Warstwa aplikacji* — udostępnia interfejs w postaci aplikacji i usług sieciowych pomiędzy siecią a użytkownikiem.

Warstwa fizyczna

Warstwa fizyczna jest najniższa warstwa modelu TCP/IP i odpowiada za fizyczną transmisję danych przez *nosnik transmisji*. Nazwa nosnika transmisji określana jest ścieżką fizyczną (przewód elektryczny, światłowód, fale radiowe itp.), która przesyłane są dane w postaci sygnałów elektrycznych lub fal elektromagnetycznych. Warstwa fizyczna odbiera dane od wyższych warstw i przetwarza je w ciąg bitów, który można z powodzeniem przesłać nosnikiem transmisji. Bit, przedstawiony na rysunku 2.2, jest podstawową jednostką komunikacji pomiędzy komputerami i urządzeniami sieciowymi i może przyjmować tylko jedną z dwóch wartości: 0 lub 1. 0 reprezentuje nieobecność sygnału w nosziku transmisji, zaś 1 oznacza obecność tego sygnału.

Rysunek 2.2.
Bit w ciągu sygnałów

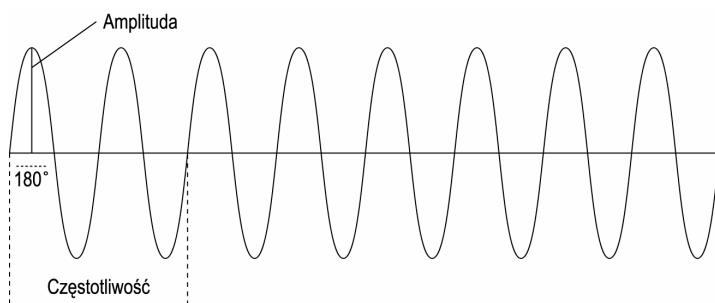


Przesyłanie sygnałów

Dane w sieci przesyłane są z jednego komputera do drugiego w postaci sygnałów. W zależności od użytego nosnika transmisji, sygnały dzieli się na dwie kategorie:

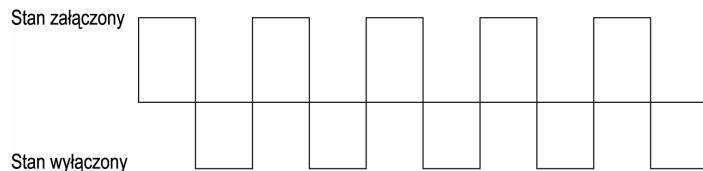
- ♦ *Sygnały analogowe* — przypominają ciąg fal sinusoidalnych, w których stan fali ulega ciągłym zmianom i przechodzi przez wszystkie wartości z dozwolonego zakresu. Rysunek 2.3 przedstawia sygnał analogowy.

Rysunek 2.3.
Sygnał analogowy



- ♦ *Sygnały cyfrowe* — posiadają tylko dwa stany: obecność danych (1) i nieobecność danych (0). „1” powszechnie nazywa się stanem złączonym (ON), zaś „0” stanem wyłączonym (OFF). Rysunek 2.4 przedstawia sygnał cyfrowy.

Rysunek 2.4.
Sygnal cyfrowy



W sygnalach analogowych mierzone sa: amplituda, częstotliwość i faza. Amplituda oznacza wartość maksymalną sygnalu, mierzoną w woltach (jesli mierzmy moc napiecia), watach (jesli mierzmy moc sygnalu) lub decybelach (jesli mierzmy stosunek mocy dwóch sygnalów). Częstotliwość oznacza liczbę pełnych okresów sygnalu w jednostce czasu i mierzoną jest w hercach (okresach na sekunde). Faza oznacza stan wzgledny sygnalu w chwili pomiaru i podawana jest w stopniach lub radianach.



Szczegółowe informacje o nosnikach transmisji i sposobach przechodzenia przez nie sygnału zawiera rozdział 3.

Typy polaczen fizycznych

Istnieją następujące sposoby łączenia komputerów w sieci przez nosnik transmisji:

- ◆ *Polaczenie dwupunktowe* — w polaczeniu tego typu jeden nosnik transmisji tworzy bezpośrednie łącze pomiędzy dwoma komunikującymi się urządzeniami (patrz rysunek 2.5). Polaczenie dwupunktowe jest szybsze, lecz droższe od wielopunktowego. Przykładem polaczenia dwupunktowego jest linia dzierżawiona, łącząca bezpośrednio organizacje z jej dostawcą usług internetowych (ISP — *Internet Service Provider*).

Rysunek 2.5.
Polaczenie dwupunktowe



Dodatkowe informacje o dostawcach usług internetowych zawiera rozdział 11.

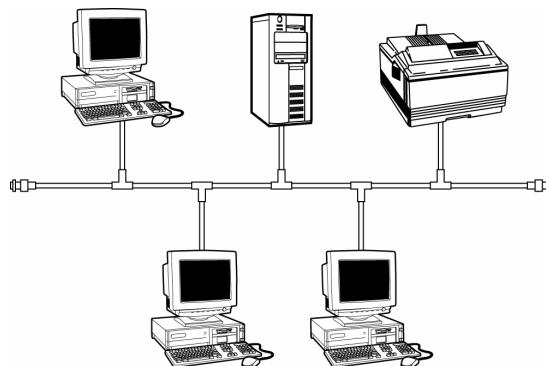
- ◆ *Polaczenie wielopunktowe* — w polaczeniu tego typu pojedynczy nosnik transmisji jest użytkowany wspólnie przez trzy lub więcej urządzeń sieciowych (patrz rysunek 2.6). W rezultacie polaczenie jest stosunkowo wolniejsze, lecz tansze od polaczeń dwupunktowych. Na przykład, możemy wiele urządzeń sieciowych połączyć z serwerem za pomocą pojedynczego kabla.

Topologie fizyczne

Fizyczny układ nosnika transmisji w sieci nazywany jest *topologia fizyczna sieci*. Do najpopularniejszych obecnie topologii sieci lokalnych (LAN) zaliczają się:

Rysunek 2.6.

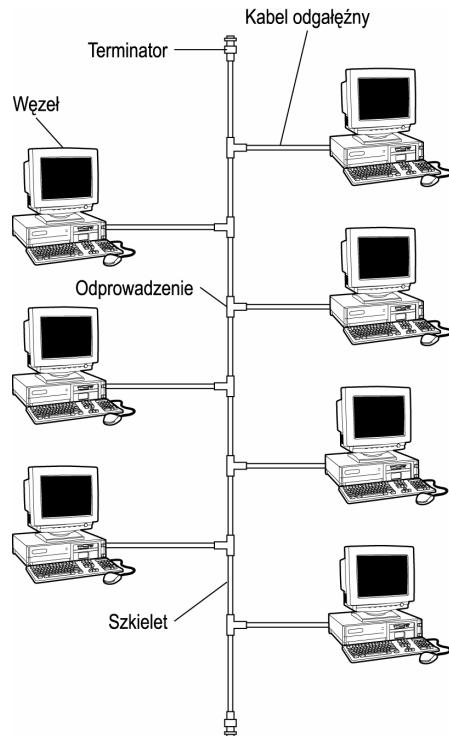
Polaczenie wielopunktowe



- ◆ *Topologia magistrali* — w tej topologii (przedstawionej na rysunku 2.7) wszystkie urządzenia sieciowe podłączone są do głównego kabla, zwanego szkieletem (ang. *backbone*), albo za pomocą krótkich kabli zwanych odgałęziami, albo bezpośrednio przez trójniki. Aby zapobiec odbiciom sygnału od końców magistrali, kabel szkieletowy musi być zakończony z obu stron terminatorami. Z wszystkich stosowanych topologii magistrala uznawana jest za najłatwiejszą i najtanszą w implementacji. Jednakże topologia ta jest wolniejsza od pozostałych.

Rysunek 2.7.

Topologia magistrali

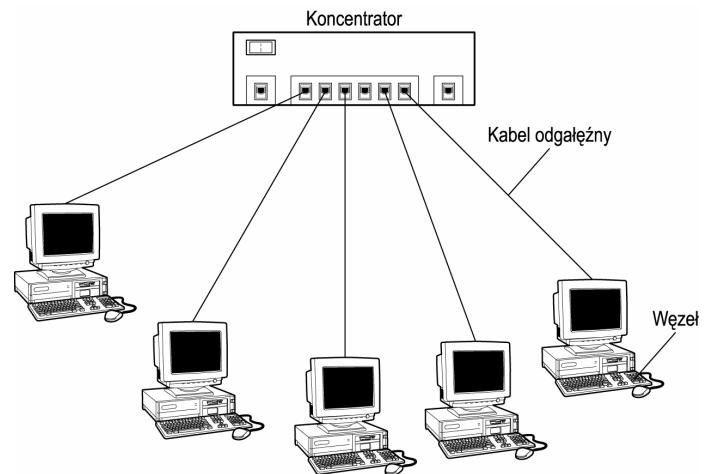


- ◆ *Topologia gwiazdy* — w tej topologii (przedstawionej na rysunku 2.8) wszystkie urządzenia sieciowe podłączone są za pomocą kabli odgałęznych do urządzenia centralnego, zwanego koncentratorem. W wyniku tego każde urządzenie posiada

dwupunktowe połaczenie z koncentratorem. Topologia ta łatwo zarządzac, łatwo ją rozbudowywac i znajdująca w niej problemy. Jednakże w przypadku awarii koncentratora cała sieć przestaje działać.

Rysunek 2.8.

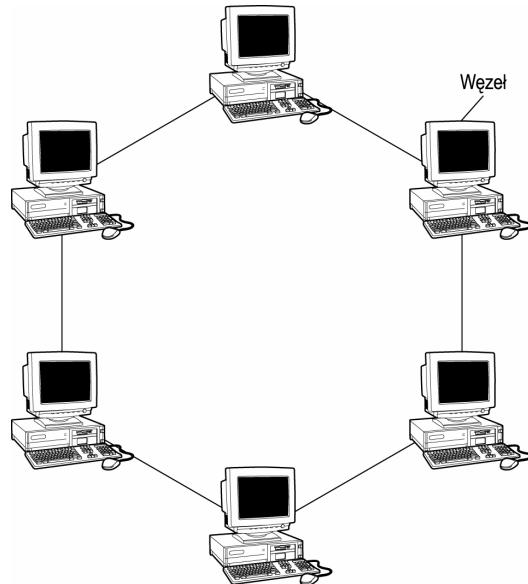
Topologia gwiazdy



- ◆ *Topologia pierscienia* — w tej topologii (przedstawionej na rysunku 2.9) każdy urządzenie sieciowe połączone jest z następnym tak, iż tworzą zamknięta pętle (pierscieni). Latwo nia zarządzac i rozwiązywać problemy, jednakże jest bardzo droga w implementacji, a zmiany konfiguracji są w niej trudne.

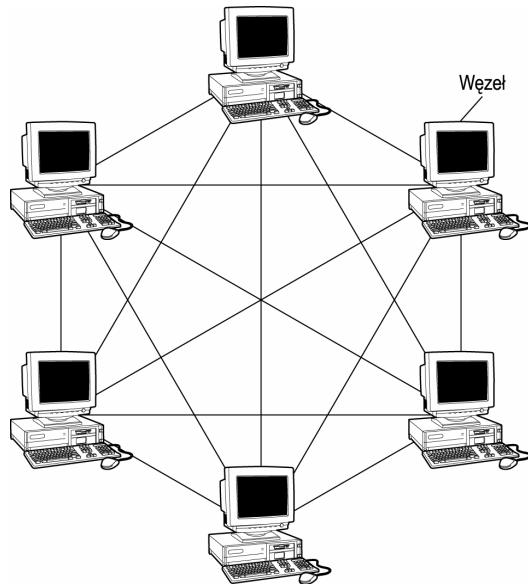
Rysunek 2.9.

Topologia pierscienia



- ◆ *Topologia oczkowa* — w tej topologii (przedstawionej na rysunku 2.10) każdy węzeł połączony jest bezpośrednio z wszystkimi pozostałymi węzłami sieci za pomocą połączeń dwupunktowych. Topologia ta jest zarówno wyjątkowo odporna na uszkodzenia, jak i wyjątkowo kosztowna w implementacji.

Rysunek 2.10.
Topologia oczkowa



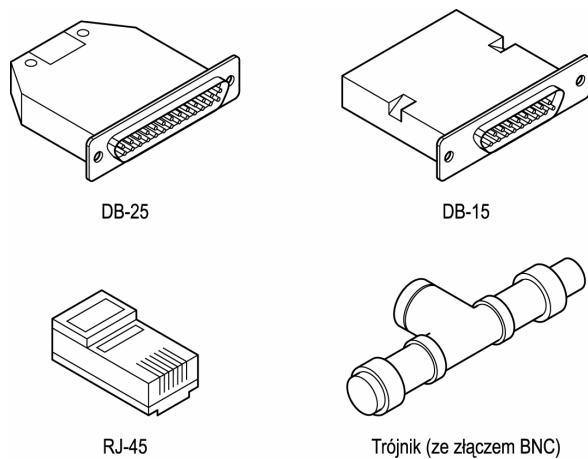
Rozdział 3. zawiera wiecej informacji o różnych topologiiach.

Urządzenia sieciowe warstwy fizycznej

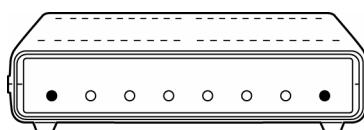
Aby zbudować sieć i połączyć każdy komputer z nośnikiem transmisji, potrzebujemy szeregu urządzeń sieciowych. Do sprzętu zwykle kojarzonego z warstwą fizyczną modelu TCP/IP należą:

- ♦ **Złącza** — złącza nośnika transmisji zapewniają połączenie pomiędzy urządzeniami sieciowymi i nośnikiem transmisji. Dla każdego nośnika transmisji istnieje jeden lub kilka typów złączy, które mogą posłużyć do przyłączenia urządzenia. Do najczęściej używanych złączy fizycznych należą (przedstawione na rysunku 2.11):

Rysunek 2.11.
Najczęściej stosowane złącza



- ◆ Trójniki i zlaczka BNC
- ◆ Zlaczka RJ-45
- ◆ Zlaczka DB-25 (inaczej RS-232)
- ◆ Zlaczka DB-15
- ◆ *Regeneratory* — im dluzsza droga ma do przebycia sygnal, tym bardziej jest tlumiony, wobec czego kazdy nosnik transmisji moze byc uzyty na ograniczona odleglosc. Nosnik mozna jednakze przedluzyc za pomoca regeneratorow. Urzadzenia te po prostu wzmacniaja sygnaly do oryginalnego poziomu. Regenerator przedstawiony jest na rysunku 2.12.

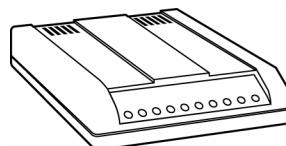
Rysunek 2.12.*Regenerator*

- ◆ *Koncentratory* — koncentrator gra role centralnego wezla do przylaczenia wielu urzadzen sieciowych. Rysunek 2.13 przedstawia typowy koncentrator. Do warstwy fizycznej naleza dwa typy koncentratorow:
 - ◆ *Koncentratory aktywne* — oprócz pelnienia funkcji centralnego punktu polaczenia, regeneruja sygnal.
 - ◆ *Koncentratory pasywne* — jedynie rozsyлаaja sygnal otrzymany od przylaczonego urzadzenia, bez regeneracji sygnalu.

Rysunek 2.13.*Koncentrator*

Istnieje jeszcze trzeci typ koncentratorow — *koncentratory inteligentne*. Jednakze te urzadzenia funkcjonuja w warstwie interfejsu sieciowego.

- ◆ *Modemy* — gdyby polaczyc komputer (który uzywa sygnalów cyfrowych) bezposrednio z analogowa linia telefoniczna (która przenosi jedynie sygnaly analogowe), lacznosc bylaby niemozliwa. Modem (ang. *M*odulator/*D*EModulator) — taki, jak na rysunku 2.14 — przekształca odebrane z komputera sygnaly cyfrowe na analogowe, które mozna przeslac analogowa linia telefoniczna. Sygnaly odebrane z analogowej linii telefonicznej sa przekształcane przez modem na cyfrowe, aby komputer móg³ je przetworzyc.

Rysunek 2.14.*Modem*

Warstwa interfejsu sieciowego

Do podstawowych zadań warstwy interfejsu sieciowego należą:

- ◆ unikatowa identyfikacja urządzeń w sieci lokalnej (LAN) za pomocą adresów sprzętowych MAC (*Media Access Control*),
- ◆ organizacja bitów otrzymanych z warstwy fizycznej w ramki,
- ◆ konwersja adresów IP na adresy LAN i vice versa,
- ◆ wykrywanie błędów i zgłoszenie ich do wyższych warstw,
- ◆ kontrola przepływu danych.

Urządzenia warstwy interfejsu sieciowego

Do urządzeń powszechnie kojarzonych z warstwą interfejsu sieciowego należą:

- ◆ *Karty interfejsu sieciowego (NIC — Network Interface Card)* — sprzętowe karty rozszerzeń, które po instalacji zapewniają komputeromłączność sieciową przez połączenie z nosnikiem transmisji.
- ◆ *Mosty* — w dużych sieciach (zwłaszcza o topologii magistrali) wszystkie urządzenia podłączone do szkieletu odbierają sygnały w nim obecne, co powoduje zbędny ruch w sieci. Możemy jednak uzyc mostu, by podzielić dużą sieć na mniejsze segmenty, wydajnie redukując niepotrzebny ruch. Gdy most odbiera sygnał, wtedy sprawdza, czy odbiorca znajduje się w lokalnym segmencie. Jeśli tak, wówczas most rozgłasza odebrany sygnał w segmencie i nie przekazuje go do innych segmentów. Jeśli odbiorca nie należy do lokalnego segmentu, wówczas most przekazuje sygnał jedynie do segmentu, w którym mieści się adresat, co efektywnie zmniejsza ruch sieciowy. Rysunek 2.15 przedstawia funkcjonowanie typowego mostu.
- ◆ *Inteligentne koncentratory* — oprócz tego, że są centralnym punktem podłączenia w łączności sieciowej i regenerują sygnał, inteligentne koncentratory przekazują sygnały tylko do urządzeń-odbiorców, nie rozgłaszając ich do wszystkich podłączonych urządzeń.

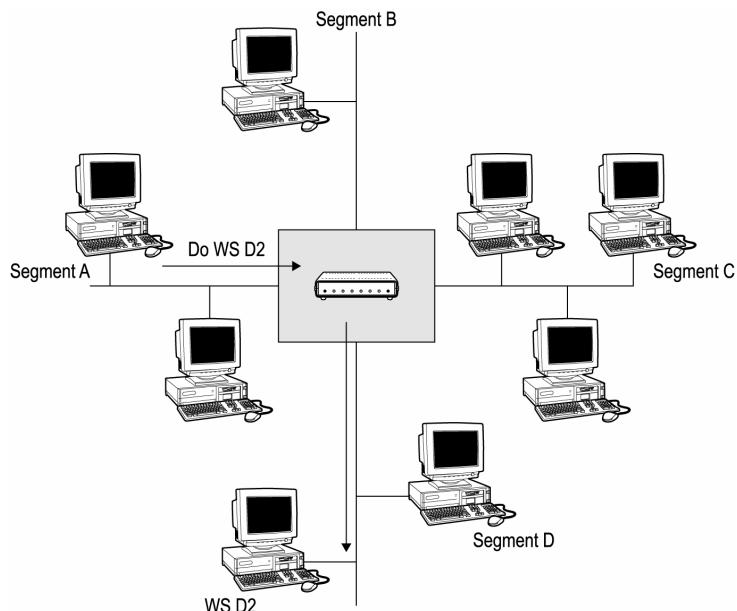
Standardy kontroli dostępu do nosnika

Aby zapewnić poprawne funkcjonowanie sieci, należy zminimalizować lub całkowicie wyeliminować możliwość równoczesnego wysłania do nosnika transmisji dwóch lub więcej sygnałów. Sieci używają reguł kontrolujących, kiedy urządzenie sieciowe może nadawać pakiety danych. Reguły te noszą nazwę *standardów kontroli dostępu do nosnika*.

W zależności od używanej topologii fizycznej, stosowane są różne standardy kontroli dostępu do nosnika:

- ◆ *Rywaliizacja* — w tej metodzie każde urządzenie w sieci rywalizuje o umieszczenie jako pierwsze swojego sygnału w nosniku transmisji. Jeśli dwa lub kilka urządzeń równoczesnie umieści swoje sygnały w nosniku, zachodzi kolizja sygnałów i zostają one odrzucone (zniszczone). Metoda ta jest powszechnie stosowana w topologii magistrali.

Rysunek 2.15.
Sposób działania mostu



- ◆ *Przekazywanie zetonu* — w tej metodzie nieustannie krazy w sieci specjalna ramka, zwana *zetonem* (ang. *token*). Dowolne urządzenie, które chce nadawać dane, przechwytuje zeton i umieszcza dane w jego ramce. Po zakończeniu transmisji urządzenie zwalnia zeton. Ta metoda stosowana jest w topologiiach pierścieniowych.
- ◆ *Odpypywanie* — w tej metodzie urządzenie nadziedne odpytuje urządzenia sieciowe w regularnych odstępach czasu. Gdy określone urządzenie chce wysłać dane, wówczas urządzenie nadziedne wysyła do niego pakiet zadania. Urządzenie umieszcza dane w ramce zadania i zwraca pakiet do urządzenia nadziednego, które następnie wysyła ramkę do odpowiedniego odbiorcy. Ta metoda dostępu stosuje powszechnie inteligentne koncentratory w topologii gwiazdy.



Szczegółowe informacje o metodach dostępu do nosnika zawiera rozdział 4.

Sterowanie przepływem

Siec składa się z urządzeń obsługujących różne predkosci transmisji — na przykład przełączniki sa znacznie szybsze od koncentratorów. Z reguły drukarki sa jednymi z najwolniejszych urządzeń sieciowych. Gdyby nadawca wysyłał ramki szybciej niż odbiorca jest w stanie je przyjmować, nadawca zarzuciłby odbiorce ramkami. Nawet gdyby transmisje były wolne od błędów, w pewnym momencie odbiorca nie byłby w stanie ich przyjmować w miarę nadsyłania i zacząłby tracić ramki. Wobec tego ilość objętości danych, która można wysłać jednokrotnie podczas komunikacji dwóch tożsamosci sieciowych, jest bardzo ważnym zagadnieniem.

Wstępnie zdefiniowane reguły sterowania przepływem zapewniają, że szybsze urządzenia nie zalewają wolniejszych danymi podczas transakcji. Sterowanie przepływem zwalnia szybkosc transmisiji nadawcy do tempa, z jakim odbiorca może sobie poradzić.

W sterowaniu przepływem stosowany jest mechanizm sprzezenia zwrotnego, za pomocą którego odbiorca może poinformować nadawcę, czy jest w stanie poradzić sobie z predkoscią transmisji. Na przykład, przy nawiązaniu połączenia odbiorca może poinformować nadawcę, aby po wysłaniu n ramek zatrzymał się i poczekał, aż do otrzymania od odbiorcy wyraznej lub pośredniej zgody na kontynuację.



Sterowanie przepływem jest zasadniczo wbudowane w różne protokoły w postaci dobrze zdefiniowanych reguł i ma wpływ zarówno na urządzenia końcowe (np. stacje robocze), jak i urządzenia pośredniczące (np. routery).

Sterowanie przepływem w warstwie interfejsu sieciowego może się odbywać według dwóch strategii:

- ◆ *Sterowanie z gwarantowaną szybkoscia przepływu* — w tej strategii nadawca i odbiorca negocjują akceptowalną szybkość transmisji dla całej sesji, jeszcze przed rozpoczęciem transmisji. Niezmienność tej szybkości jest gwarantowana na okres całej sesji.
- ◆ *Sterowanie przepływem za pomocą okien* — takie sterowanie przepływem pozwala dwóm połączonym urządzeniom wynegocjować rozmiary bufora (okna), w którym można umieścić zadana liczbę ramek. Istnieją dwa typy sterowania przepływem w oknach:
 - ◆ *Statyczne* — w chwili nawiązania połączenia ustalają się na jego koncach i wspólnie rozmiary okna i używają ich przez całą sesję, aż do jej zamknięcia. Założymy, że na początku sesji nadawca i odbiorca umawiają się na rozmiar okna wynoszący osiem ramek. Wówczas nadawca zbiera osiem ramek danych, przydziela do każdej tymczasowy numer i umieszcza ramki w nosniku transmisji. W tym przypadku numer okna będzie z przedziału od jeden do osiem. Po odebraniu ramki odbiorca musi wysłać potwierdzenie. Jeśli nadawca wysłał wszystkie osiem ramek, musi czekać na potwierdzenie odbioru przynajmniej jednego z przydzielonych numerów, a następnie powtarza cały proces dla kolejnych ośmiu ramek. Proces ten zapewnia, iż w każdej chwili nie zalega więcej niż osiem ramek.



Ta strategia powoduje marnowanie przepustowości łącza, ponieważ każda wysłana rama musi zostać potwierdzona.

- ◆ *Dynamiczne* — podczas nawiązywania połączenia ustalone zostają rozmiary okna. Jednakże ten typ sterowania przepływem pozwala urządzeniom sieciowym dostosowywać rozmiary okna do wymogów chwili, zgodnie ze statusem odbiorcy. Na początku połączenia ustalany jest maksymalny rozmiar okna. Gdy w czasie transmisji bufor odbiorcy zacznie się przepelnić, wówczas wysyła on natychmiast *pakiet tlumienia*. Pakiet ten jest dla nadawcy sygnałem, by zwolnić. Po jakimś czasie nadawca zaczyna powoli zwiększać szybkość transmisji, aż do odebrania kolejnego pakietu tlumienia. W ten sposób rozmiar okna jest nieustajaco regulowany podczas samej transmisji. Dynamiczne sterowanie przepływem za pomocą okien nazywane jest również *sterowaniem z oknem płynącym lub przesuwającym*.

Warstwa internetowa

Warstwa interfejsu sieciowego identyfikuje unikalowo urządzenie w sieci lokalnej za pomocą adresów fizycznych, zakodowanych na trwale w kartach interfejsów sieciowych. Noszą one inaczej nazwę adresów sterowania dostępu do nosnika (MAC — *Media Access Control*). Jednakże ta metoda unikalowej identyfikacji urządzeń nie jest skuteczna, gdy liczność zachodzi pomiędzy dwoma urządzeniami położonymi w różnych sieciach. Do przesyłania pakietów pomiędzy sieciami warstwa internetowa używa adresów IP.

Adres IP jest 32-bitowa binarna konwencja nazewnicza, która została opracowana na potrzeby globalnej komunikacji. Adresy IP, w celu łatwego zapamiętania, notowane są w postaci czterech dziesiętnych liczb całkowitych oddzielonych kropkami. Na przykład, 23.33.71.11 jest adresem IP.

W zależności od liczby hostów i sieci, które mogą być obsługiwane w danym zakresie adresów, istnieje pięć klas adresów IP:

- ◆ klasa A, obejmująca adresy IP od 0.1.0.0 do 126.0.0.0
- ◆ klasa B, obejmująca adresy IP od 128.0.0.0 do 191.255.0.0
- ◆ klasa C, obejmująca adresy IP od 192.0.1.0 do 223.255.255.0
- ◆ klasa D, obejmująca adresy IP od 224.0.0.0 do 239.255.255.255
- ◆ klasa E, obejmująca adresy IP od 240.0.0.0 do 247.255.255.255



Szczegółowe informacje o adresowaniu IP oraz klasach adresów IP znajdują się w rozdziale 5.

Komutacja

Pomiędzy dwoma urządzeniami komunikującymi się ze sobą w sieci może istnieć wiele niż jedna łączaca je ścieżka. Aby zapewnić szybkie dostarczenie danych, sygnały mogą w miarę potrzeb być przelaczany (komutowany) pomiędzy tymi ścieżkami, za pomocą poniższych trzech technik komutacji:

- ◆ *Komutacja obwodów* — w tej metodzie wymagany jest dedykowany kanał (obwód) łączności pomiędzy dwoma komunikującymi się urządzeniami.
- ◆ *Komutacja komunikatów* — w tej metodzie komutacji nie trzeba nawiązywać dedykowanego fizycznego połączenia pomiędzy punktami końcowymi łączności. Komunikat jest dzielony na małe części, którym zostają przydzielone numery. Część jest traktowana jak niezależna całość; wszystkie zawierają też informacje o adresie docelowym. Komunikaty są składane w każdym przełączniku przed przesaniem do następnego przełącznika na trasie.
- ◆ *Komutacja pakietów* — w tej metodzie komunikaty dzielone są na segmenty zwane pakietami, które następnie są przesyłane niezależnie przez sieć, własnymi trasami. Kazdy pakiet zawiera oprócz właściwych danych adres źródłowy i docelowy.



Jest jedna podstawowa różnica pomiędzy dwiema ostatnimi metodami. W komutacji komunikatów nie istnieje górna granica rozmiarów bloku komunikatów, zas w komutacji pakietów rozmiar pakietu ograniczony jest do ustalonej wartości.

Wykrywanie i wybór tras

Ruterami są urządzeniami sieciowymi skojarzonymi z funkcjami warstwy internetowej. Aby zapewnić najszybsze dostarczenie danych z jednego urządzenia do drugiego, ruter musi wykryć najkrótszą i najszybszą trasę. Ta metoda ustalania tras do sieci docelowej nosi nazwę *wykrywania tras* (ang. *route discovery*). Istnieją dwie metody wykrywania tras:

- ◆ *Metoda wektora odległości* — w tej metodzie każdy ruter utrzymuje tablice tras (ang. *routing table*), która rozglasza w regularnych odstępach czasu. Dzięki rozgłoszeniom innych ruterów, każdy ruter regularnie aktualizuje informacje o wszelkich nowych trasach. Chociaż ta metoda zapewnia każdemu ruterowi posiadanie najswiezszych tablic tras, generuje bardzo wysokie obciążenie łączy.
- ◆ *Metoda stanu połączenia* — w tej metodzie rozgłoszenia generowane są tylko wtedy, gdy nastąpi dowolna zmiana w istniejącej tablicy tras rутera. Pozostałe ruter, które odbierają rozgłoszenie, odpowiednio aktualizują swoje tablice tras. W rezultacie metoda ta generuje znacznie mniejszy ruch w sieci.



Rozdział 5. zawiera bardziej szczegółowe informacje o wyznaczaniu tras.

Ruter po zbudowaniu tablicy tras, przez wykrycie tras do sieci docelowych, wybiera właściwą trasę do sieci docelowej, obliczając najlepszą ścieżkę transmisji. Wybór może odbywać się zarówno *dynamycznie*, jak i *statycznie*:

- ◆ *Dynamiczny wybór tras* — jeśli w dowolnej chwili dostępnych jest wiele tras do urządzenia docelowego, ruter ustala najlepszą z nich. Ten wybór odbywa się w każdym ruterze po drodze do urządzenia docelowego. Inaczej mówiąc, tablica tras jest utrzymywana automatycznie, bez interwencji administratora sieci.
- ◆ *Statyczny wybór tras* — nawet jeśli dostępnych jest wiele tras do urządzenia docelowego, do przesłania pakietów uzyta zostaje jedynie trasa wyznaczona przez administratora sieci. Ruter po drodze do urządzenia docelowego nie może podejmować decyzji o wyznaczaniu tras. Inaczej mówiąc, tablica tras jest tworzona i utrzymywana przez administratora sieci.



Szczegółowe informacje o statycznym i dynamicznym wyborze tras znajdują się w rozdziale 19.

Warstwa transportowa

Czwarta warstwa modelu TCP/IP — transportowa — jest przede wszystkim odpowiedzialna za:

- ◆ udostępnienie interfejsu pomiędzy warstwami niższymi (internetowa, interfejsu sieciowego i fizycznego) a warstwą aplikacji,
- ◆ dostarczenie danych od nadawcy do odbiorcy.

Nizsze warstwy moga zlokalizowac zamierzonego odbiorce (w tej samej sieci lub w innych sieciach) i wyslac do niego dane. Jednakze warstwy te nie moga zapewnic wiarygodnych uslug polaczeniowych. Warstwa transportowa spełnia powyzsze wymagania. Uzywa ona do celów lacznosci dwóch protokolów — TCP i UDP (*User Datagram Protocol* — protokół datagramów uzytkownika). TCP swiadczy uslugi polaczeniowe, zas UDP bezpolaczeniowe.

**Uwaga**

Wiarygodnosc uslug polaczeniowych nie oznacza, iz dane zostana przeslane bez wzgledu na okolicznosci. Pojecie polaczenia wiarygodnego (*reliable*) oznacza, iz protokoly warstwy transportowej potrafia potwierdzic pomyslny odbior danych lub poinformowac o niepowodzeniu. Jesli dane nie dotarly do odbiorcy lub ulegly uszkodzeniu w trakcie transmisji, wówczas warstwa transportowa moze zainicjowac retransmisje. Warstwa aplikacji również jest informowana o niepowodzeniach, dzieki czemu moze zainicjowac dzialania korekcyjne lub powiadomic uzytkownika.

Uslugi polaczeniowe

Warstwa transportowa udostepnia dwa typy uslug polaczeniowych:

- ♦ *Zorientowane na polaczenie (polaczeniowe)* — gdy dane przesypane sa z jednego urzadzenia sieciowego do innego, kazda pomyslnie przeslana porcja nie uszkodzonych danych jest potwierdzana przez odbiorce. Nadawca nie wysle nastepnych danych, dopoki nie odbierze pozytywnego potwierdzenia dotyczacego ostatniej wyslanej porcji. Jesli dane podczas transmisji ulegna zagubieniu lub uszkodzeniu, nadawca nie otrzyma od odbiorcy odpowiedniego potwierdzenia. Nadawca musi ponownie wyslac albo utracony pakiet, albo cala porcje, w zaleznosci od implementacji protokolu. Uslugi zorientowane na polaczenia udostepniaja również sterowanie przeplywem i kontrole bledów.
- ♦ *Bezpolaczeniowe* — urzadzenie nadajace wysyla dane do odbiorcy i nie odpowiada za retransmisyje wszelkich danych uszkodzonych lub utraconych podczas transmisji do odbiorcy. Istnieja dwa typy uslug bezpolaczeniowych:
 - ♦ *Potwierdzane uslugi bezpolaczeniowe* — komunikaty potwierdzajace sa wymieniane, jesli transmisja jest dwupunktowa. Tego typu uslugi również zapewniaja kontrole bledów i sterowanie przeplywem, o ile transmisja odbywa sie dwupunktowo.
 - ♦ *Nie potwierdzane uslugi bezpolaczeniowe* — transmisje nie sa potwierdzane i nie sa dostepne zadne metody kontroli bledów, sterowania przeplywem, czy tez kontroli sekwencji pakietów.



Szczegolowe informacje o uslugach zorientowanych na polaczenie i bezpolaczeniowych zawiera rozdzial 6.

Obsluga segmentow

Oprócz wiarygodnych uslug polaczeniowych warstwa transportowa odpowiada takze za podzial duzych komunikatow warstwy aplikacji na segmenty, ktore mozna przeslac no-snikiem transmisji. Proces ten nosi nazwe *fragmentacji*. Gdy urzadzenie sieciowe od-biera komunikat w postaci kilku segmentow, warstwa transportowa odpowiada za po-

prawne złożenie tych segmentów w oryginalny komunikat — ten proces nazwany jest *defragmentacją*.

Sterowanie przepływem w warstwie transportowej

Sterowanie przepływem w warstwie transportowej nazywane jest również dwupunktowym sterowaniem przepływem (ang. *end-to-end flow control*), ponieważ zajmuje się połączeniami pomiędzy wezlami nadawcy i odbiorcy. Warstwa transportowa dokonuje sterowania przepływem za pomocą poniższych typów potwierdzeń:

- ♦ *Potwierdzenia pozytywne i negatywne* — gdy przesłane dane są odebrane bez strat i uszkodzeń, odbiorca wysyła do nadawcy potwierdzenie pozytywne. Jeśli jednak dane ulegną uszkodzeniu, odbiorca wysyła potwierdzenie negatywne. W drugim przypadku warstwa transportowa albo warstwa aplikacji, która zainicjowała transakcję, podejmuje działania korekcyjne.
- ♦ *Potwierdzenie „wróć do n”* — potwierdzenie „wróć do n” („go back n”) oznacza, iż nadawca musi ponownie przesłać część komunikatu, zaczynając od pakietu o numerze n z ostatniej transakcji.
- ♦ *Potwierdzenie z selektywnym powtórzeniem* — oznacza, iż ciąg pakietów został odebrany poprawnie, lecz kilka zawartych w nim pakietów zostało podczas transmisji uszkodzonych lub uszkodzonych. Potwierdzenie takie mówi nadawcy, aby zamiast całego ciągu wysłał ponownie jedynie pakiety brakujące i uszkodzone.

Kontrola błędów

Utrata danych podczas transmisji jest niekiedy nieunikniona, a ponadto istnieje możliwość dotarcia do celu danych uszkodzonych w procesie transmisji. Warstwa transportowa naprawia te błędy w następujący sposób:

- ♦ Podczas transmisji segmentom przydzielane są unikatowe numery, aby zapobiec wystąpieniu podwójnych numerów segmentów, a co za tym idzie — utracie pakietów.
- ♦ Pakiety, których dopuszczalny czas istnienia został przekroczony (co ustala się na podstawie wartości TTL, używanej przez warstwę internetową), są odrzucone, ponieważ im dłużej pakiet danych podróżuje w sieci, tym większe jest prawdopodobienstwo jego uszkodzenia.
- ♦ Podczas sesji używana jest tylko jedna wirtualna trasa, aby zminimalizować szansę utraty pakietów danych.



Szczegółowe informacje o warstwie transportowej znajdują się w rozdziale 6.

Warstwa aplikacji

Warstwa aplikacji mieści się na szczycie modelu architektury TCP/IP. Jest warstwa najważniejszą, ponieważ użytkownik pracuje z nią bezpośrednio. Warstwa aplikacji obsługuje wszystkie niezbędne protokole, aby świadczyć usługi sieciowe: na przykład, usługi

plikowe, przesyłanie wiadomości, usługi baz danych, czy też usługi drukowania. W istocie wszystkie pozostałe warstwy istnieją tylko po to, by obsługiwać warstwę aplikacji.



Pakietы oprogramowania, na przykład Microsoft Word, Excel i tak dalej, nie należą do warstwy aplikacji. Jedyne aplikacje inicjujące zadania, które mogą być obsługiwane przez inne urządzenia sieciowe — na przykład poczta elektroniczna — uznawane są za składniki warstwy aplikacji.

Do najczęściej używanych protokołów warstwy aplikacji zaliczają się:

- ◆ *FTP (File Transfer Protocol — protokół transferu plików)* — bezpieczny i niezawodny protokół, służący do przesyłania plików ze zdalnego komputera do lokalnego i odwrotnie. Aby umożliwić transfer plików, użytkownik musi nawiązać połączenie ze zdalnym komputerem.
- ◆ *TFTP (Trivial File Transfer Protocol — prosty protokół transferu plików)* — protokół, który używa UDP w roli swojego protokołu transportowego. Dzięki temu użytkownik, aby przesyłać pliki, nie musi nawiązywać połączenia z drugim urządzeniem ani logować się do zdalnego systemu.



Dodatkowe informacje o FTP i TFTP znajdują się rozdział 12.

- ◆ *Telnet (TELecommunication NETwork)* — protokół, który pozwala użytkownikom pracować ze zdalnym systemem tak, jak z lokalnym. Jest to możliwe, ponieważ Telnet przejmuje lokalną interpretację informacji wprowadzanych z klawiatury.



Dodatkowe informacje o usłudze Telnet znajdują się rozdział 13.

- ◆ *SMTP (Simple Mail Transfer Protocol — prosty protokół przesyłania poczty)* — protokół, który używany z aplikacją poczty elektronicznej pozwala użytkownikom odbierając i wysyłając pocztę elektroniczną (e-mail) przez sieć.



Dodatkowe informacje o SMTP znajdują się rozdział 16.

- ◆ *SNMP (Simple Network Management Protocol — prosty protokół zarządzania siecią)* — protokół służący do zarządzania siecią. SNMP przede wszystkim zbiera, analizuje i raportuje dane związane z działaniem różnych składników sieci na potrzeby aplikacji służących do zarządzania siecią.

Lacznosc pomiedzy warstwami

Według modelu architektury TCP/IP warstwa może w stosie komunikować się z warstwą równorzędną w innych urządzeniach. W tym celu jednak musi przesłać dane lub komunikaty przez niższe warstwy stoso, do którego należy. Warstwa może skorzystać z usług warstwy znajdującej się bezpośrednio pod nią, a zarazem musi świadczyć usługi warstwie bezpośrednio nad sobą.

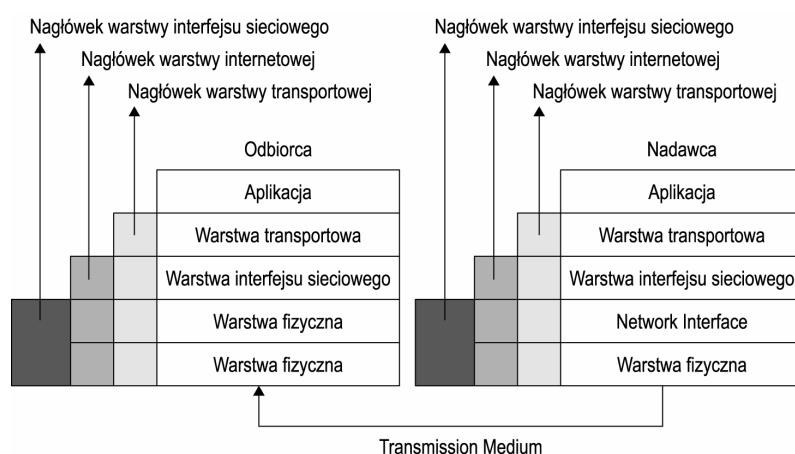
Gdy warstwa przekazuje dane do nizszej, dodacza do tych danych własny *nagłówek* (ang. *header*). Nagłówek zawiera informacje sterujące danej warstwy. Jedynie równorzędną warstwę w innym stanie jest w stanie przetworzyć te informacje. Ogólnie rzecz biorąc, zadania usług sieciowych pochodzą z warstwy aplikacji. W takim przypadku komunikat zostaje przesłany w dół, do warstwy transportowej, która dzieli komunikat na mniejsze segmenty, które można przesłać noszkiem transmisji. Warstwa transportowa, podobnie jak jej poprzednik, również dodaje własny nagłówek do każdego segmentu i przesyła segmenty dalej, do warstwy internetowej. Ten proces dzielenia dużych komunikatów na segmenty nosi nazwę *fragmentacji*. Następnie warstwa internetowa dodaje swój nagłówek do każdego segmentu i przekazuje pakiet do warstwy interfejsu sieciowego. Podobnie jak wszystkie wyższe warstwy, ta dodaje własny nagłówek do datagramów otrzymanych z warstwy internetowej i wysyła ramki do warstwy fizycznej. Warstwa fizyczna dzieli każdą ramkę na sekwencje bitów i umieszcza te sygnały w nośniku transmisji.



Dane w warstwie aplikacji są określone mianem *komunikatu* (*message*). W warstwie transportowej dane noszą nazwę *segmentów* lub *datagramów*. W warstwie internetowej segmenty nazywane są *pakietami*. Dane przesypane do warstwy interfejsu sieciowego noszą nazwę *ramek*, zas w warstwie fizycznej — *bitów* lub *sygnałów*.

Po przesłaniu sygnałów do zamierzonego odbiorcy, do czego służy nagłówki warstw internetowej i interfejsu fizycznego, proces przetwarzania sygnałów po stronie odbiorcy jest dokładnie odwrotny do procesu po stronie nadawcy. Warstwa fizyczna odbiorcy odbiera sygnały z nośnika transmisji i przekazuje je do warstwy interfejsu sieciowego. Ta z kolei, używając danych sterujących zamieszczonych przez swojego odpowiednika u nadawcy, przekształca ciągi bitów w ramki i przekazuje je do warstwy internetowej. Warstwa internetowa usuwa odpowiadający jej nagłówek i przekazuje pakiety do warstwy transportowej. Ta z kolei, używając danych sterujących zamieszczonych w nagłówku przez swojego odpowiednika u nadawcy, składają segmenty w komunikat. Proces laczenia segmentów w komunikat nosi nazwę *defragmentacji*. Następnie warstwa transportowa przesyła komunikat do warstwy aplikacji, która go przetwarza oraz, w razie potrzeby, wyświetla informacje dla użytkownika. Cały proces opakowywania i rozpakowywania został przedstawiony na rysunku 2.16.

Rysunek 2.16.
Proces opakowywania i
rozpakowywania



Format nagłówka warstwy transportowej

W zaleznosci od typu lacznosci — gwarantowanej lub nie — nagłówek warstwy transportowej moze nalezec do jednego z dwóch typów: TCP lub UDP.

Format nagłówka TCP

Nagłówek TCP, przedstawiony na rysunku 2.17, sklada sie z nastepujacych pól:

Rysunek 2.17.
Format nagłówka TCP

Port źródłowy	Port docelowy		
Numer kolejny			
Numer potwierdzenia			
HLEN	Zarezerowane	Bity sterujace	Okno
Suma kontrolna			Wskaźnik pilnosci
Opcje (ewentualne)			Wypełnienie

- ◆ *Adres portu źródłowego* — zawiera adres portu TCP aplikacji po stronie nadawcy, która zainicjowala zadanie. Pole to ma dlugosc dwóch bajtów.
- ◆ *Adres portu docelowego* — zawiera adres portu TCP aplikacji po stronie odbiorcy, która musi odpowiedziec na zadanie. Pole o dlugosci dwóch bajtów.
- ◆ *Numer kolejny* — zawiera numer porządkowy segmentu przydzielony podczas podzialu komunikatu na segmenty. Pole o dlugosci czterech bajtów.
- ◆ *Numer potwierdzenia* — zawiera numer nastepnego segmentu, który powinien dotrzec do odbiorcy. Pole o dlugosci czterech bajtów.
- ◆ *HLEN* — zawiera dlugosc nagłówka segmentu. Pole o dlugosci czterech bitów.
- ◆ *Zarezerowane* — jego wartosc musi byc równa zeru, poniewaz to pole jest zarezerowane do wykorzystania w przyszlosci. Pole o dlugosci szesciu bitów.
- ◆ *Bity sterujace* — zawiera szesc ponizszych jednobitowych pól, które wskazuja, jak nalezy interpretowac pozostale pola nagłówka:
 - ◆ *URG* — jesli wartosc jest równa 0, pole *Wskaźnik pilnosci* powinno zostac zignorowane. Jesli wartosc jest równa 1, pole jest obowiazujace.
 - ◆ *ACK* — jesli równe 0, pole *Numer potwierdzenia* powinno zostac zignorowane. Jesli 1, pole jest obowiazujace.
 - ◆ *PSH* — jesli równe 0, to pole powinno zostac zignorowane. Jesli 1, segment inicjuje funkcje *push*.
 - ◆ *RST* — jesli równe 0, to pole powinno zostac zignorowane. Jesli równe 1, polaczenie jest zerowane.
 - ◆ *SYN* — jesli równe 0, segment zada nawiazania nowego polaczenia.
 - ◆ *FIN* — jesli równe 1, oznacza, iż nadawca nie ma wiecej danych do wyslania i polaczenie musi zostac zamkniête po biezacym segmencie.
- ◆ *Okno* — zawiera rozmiar bufora nadawcy i ustala liczbe bajtów, jaka nadawca segmentu jest obecnie w stanie przyjac. Pole o dlugosci dwóch bajtów.

- ♦ *Suma kontrolna* — zawiera sumę kontrolną, służącą do weryfikacji poprawności odebranych danych. Pole to zawiera również pseudonagłówek, który pomaga odbiorcy stwierdzić, czy segment dotarł do właściwego celu. Pole o długości dwóch bajtów.
- ♦ *Wskaznik pełnosci* — zawiera informacje określające pozycję w segmencie, na której kończy się pełne dane. Pole to przetwarzane jest tylko wtedy, gdy pole URG w bitach sterujących ma wartość 1. Pole o długości dwóch bajtów.
- ♦ *Opcje* — zawiera informacje o kilku funkcjach, na przykład maksymalnym rozmiarze segmentu (MSS — *Maximum Segment Size*), jaki punkty końcowe połączenia mogą odebrać, pole końca opcji i tak dalej. Pole o zmiennej długości.
- ♦ *Wypełnienie* — zawiera ciąg zer dodanych do nagłówka, aby jego długość wynosiła 32 bajty. Pole o zmiennej długości.

Format nagłówka UDP

Nagłówek UDP, przedstawiony na rysunku 2.18, składa się z następujących pól:

Rysunek 2.18.

Format nagłówka UDP

0	16	31
Źródłowy port UDP	Docelowy port UDP	
Długość komunikatu UDP	Suma kontrolna UDP	

- ♦ *Adres portu źródłowego* — zawiera adres portu UDP aplikacji po stronie nadawcy, która zainicjowała zadanie. Pole to ma długość dwóch bajtów.
- ♦ *Adres portu docelowego* — zawiera adres portu UDP aplikacji po stronie odbiorcy, która musi odpowiedzieć na zadanie. Pole o długości dwóch bajtów.
- ♦ *Długość* — podaje długość segmentu. Pole o długości dwóch bajtów.
- ♦ *Suma kontrolna* — zawiera pseudonagłówek, który pomaga odbiorcy stwierdzić, czy segment dotarł do właściwego celu. Pole opcjonalne, o długości dwóch bajtów.

Format nagłówka warstwy internetowej

Nagłówek warstwy internetowej, przedstawiony na rysunku 2.19, składa się z następujących pól:

Rysunek 2.19.

Format nagłówka warstwy internetowej

Wersja	HLEN	Typ usługi	Długość całkowita
Identyfikacja			Flagi Przesunięcie fragmentu
Czas życia	Protokół	Suma kontrolna nagłówka	
Źródłowy adres IP			
Docelowy adres IP			
Opcje IP		Wypełnienie	

- ♦ *Wersja* — określa wersję protokołu IP. Obecnie stosowana jest wersja 4 (*IPv4*). Pole o długości czterech bitów.
- ♦ *Długość* — zawiera długość nagłówka warstwy internetowej. Pole o długości czterech bitów.

- ♦ *Typ uslugi* — zawiera informacje, jak nalezy przetwarzac datagram oraz o pozadanej jakosci uslug (QoS— *Quality of Service*). Pole o dlugosci jednego bajta.
- ♦ *Dlugosc całkowita* — zawiera całkowita dlugosc datagramu, lacznie z nagłówkiem i zawartymi danymi. Pole o dlugosci dwóch bajtów.



Dlugosc tego pola — 16 bitów wskazuje, iz maksymalna dlugosc datagramu (pakietu) IP moze wynosic 65 535 bajtow (216). Minimalna dlugosc pakietu IP wynosi 576 bajtow.

- ♦ *Identyfikacja* — zawiera informacje sluzace do ponownego złożenia datagramu z fragmentów. Pole o dlugosci dwóch bajtów.
- ♦ *Flagi* — zawiera trzy ponizsze flagi sterujace:
 - ♦ *Bit 0* — zarezerwowany; jego wartosc musi zawsze wynosic 0.
 - ♦ *Bit 1* — jesli jego wartosc wynosi 0, datagram mozna pofragmentowac. Jesli jest rowna 1, datagra mu fragmentowac nie wolno.
 - ♦ *Bit 2* — jesli jego wartosc wynosi 0, fragment jest ostatni w strumieniu danych i nie nastepuja po nim zadne dalsze. Jesli wynosi 1, po fragmencie nastepuja kolejne.
- ♦ *Przesuniecie fragmentu* — zawiera pozycje fragmentu w datagramie, jesli jest on podzielony na fragmenty. Pole o dlugosci trzynastu bitów.
- ♦ *Czas zycia (TTL — Time to Live)* — zawiera maksymalny czas zycia (w sekundach), przez jaki datagram moze istniec. Kazdy ruter, przez który datagram przechodzi po drodze do celu, zmniejsza te wartosc o 1. Gdy wartosc w polu spadnie do zera, datagram zostaje odrzucony. Pole o dlugosci jednego bajta.
- ♦ *Protokol* — zawiera informacje o protokole warstwy aplikacji, który zapoczatkował zadanie. Pole o dlugosci jednego bajta.



Wartosci odpowiadajace poszczególnym protokołom wyszczególnione sa w RFC 1700.

- ♦ *Suma kontrolna nagłówka* — zawiera sume kontrolna jedynie z samego nagłówka IP. Po kazdej modyfikacji nagłówka te wartosc trzeba obliczyc na nowo. Pole o dlugosci dwóch bajtów.
- ♦ *Zródłowy adres IP* — zawiera adres IP urzadzenia nadawczego. Pole o dlugosci czterech bajtów.
- ♦ *Docelowy adres IP* — zawiera adres IP urzadzenia odbiorczego. Pole o dlugosci czterech bajtów.
- ♦ *Opcje IP* — zawiera informacje o kilku funkcjach IP. Pole ma zmienna dlugosc.
- ♦ *Wypełnienie* — zawiera ciąg zer, dodanych do nagłówka, aby jego dlugosc wynosila 32 bajty. Pole o zmiennej dlugosci.



Format nagłówka warstwy interfejsu sieciowego opisany jest w rozdziale 4.

Rozdział 3.

Warstwa fizyczna

W tym rozdziale:

- ◆ Przesyłanie danych po kablu
- ◆ Przegląd najczęściej stosowanych topologii

Każda warstwa w systemie komunikacyjnym odgrywa decydującą rolę w udanej laczności sieciowej. Niepowodzenie w jednej warstwie prowadzi do niesprawności całego systemu, wobec tego aby sieciowy system komunikacyjny działał poprawnie, wszystkie warstwy komunikacyjne muszą funkcjonować poprawnie. Warstwa fizyczna, położona najbliżej w pięciowarstwowej architekturze TCP/IP, zajmuje się fizyczną transmisją danych w sieci komputerowej. Warstwa fizyczna odbiera dane przekazywane z warstw wyższych i formuluje je do postaci, która można przesłać nosnikiem transmisyjnym — na przykład kablem, światłowodem, sygnałem mikrofalowym lub radiowym. Niniejszy rozdział przedstawia charakterystykę warstwy fizycznej, różne nosniki, których można użyć do transmisji danych, oraz topologie powszechnie stosowane do komunikacji.

W jaki sposób sygnał przesyłany jest kablem

Warstwa fizyczna odpowiada przede wszystkim za wysyłanie i odbieranie bitów. Warstwa ta formuluje komunikaty odebrane z wyższej warstwy i wysyła je nosnikiem w postaci bitów (zer i jedynek). W różnych typach nosników bity reprezentowane są w różny sposób, na przykład w postaci różnych częstotliwości sygnałów dźwiękowych lub różnych napięć. Warstwa fizyczna definiuje specyfikacje implementacji korzystających z określonego nosnika transmisyjnego. Do implementacji warstwy fizycznej należą Ethernet, Token Ring, ARCnet, FDDI i łączność bezprzewodowa. Dla każdej implementacji warstwa fizyczna posiada w określonym nosniku zestaw protokołów, które opisują wzorce układu bitów, sposób kodowania danych na sygnały w nosniku oraz interfejs łączący z fizycznym nosnikiem.

Warstwowa architektura systemu komunikacyjnego daje możliwość wprowadzania zmian w jednej warstwie bez wpływu na pozostałe. W miarę rozwoju technologii nosników fizycznych, można odpowiednio modyfikować warstwę fizyczną. Ponieważ TCP/IP posiada architekturę warstwową, można zmieniać warstwę fizyczną tak, by skorzystać z nowych technologii, bez wpływu na funkcjonowanie pozostałych warstw komunikacyjnych.

Metody transmisji (metody sygnalizacji)

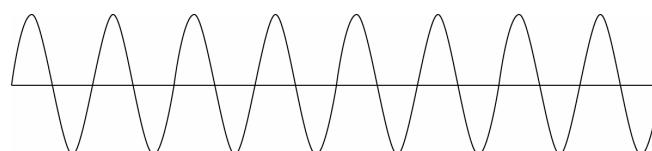
Metoda sygnalizacji oznacza sposób, w jaki dane przesyłane są przez nosnik. Sygnały korzystają z energii elektrycznej. W zależności od używanego nosnika transmisji, mogą być przesyłane sygnały analogowe lub cyfrowe.

Transmisja analogowa

W transmisji analogowej dane wymieniane pomiędzy komputerami mają postać sygnałów audio. Noszą one nazwę sygnałów analogowych. Ich cecha charakterystyczna są zmiany poziomu w całym zakresie wartości pomiędzy wartościami skrajnymi.

Sygnały analogowe są zwykle reprezentowane przez ciągi fal sinusoidalnych, jak na rysunku 3.1. Każda fala składa się z grzbietów (górnego połowyki) oraz niecek (dolne połowyki). Jeden grzbiet i jedna niecka tworzą razem okres sygnału. Każda fala charakteryzuje określone parametry: amplituda, częstotliwość i faza. Amplituda oznacza wielkość grzbietu lub niecki, odległość pomiędzy skrajnymi wartościami sygnału: najwyższą i najniższą. Częstotliwość oznacza liczbę okresów w jednostce czasu, zaś faza oznacza kąt fali, licząc od punktu początkowego. Każdy sygnał identyfikowany jest przez te parametry fali, które reprezentują faktyczne dane.

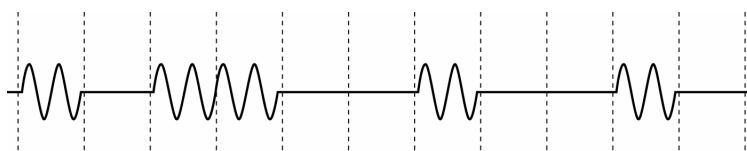
Rysunek 3.1.
Sygnał analogowy



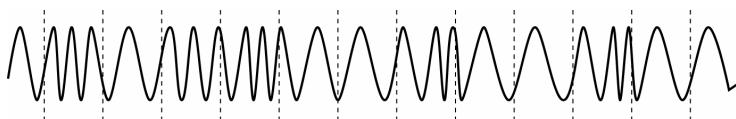
Warstwa fizyczna wysyła dane binarne analogowym nosnikiem transmisji. Te dane binarne przetwarzane są na sygnały o zmieniającej się częstotliwości i amplitudzie; zostają one następnie nalozone na elektromagnetyczną falę nosną. Fale nosne są elektromagnetycznymi falami analogowymi, które przenoszą sygnały z jednego punktu koncowego komunikacji do drugiego. Podczas transmisji dodane sygnały zmieniają jeden lub kilka parametrów fali nosnej: amplitudę, częstotliwość lub fazę. Ten proces modyfikacji parametrów fali nosnej nosi nazwę modulacji lub kluczowania. Istnieją trzy typy kluczowania:

- ◆ *Kluczowanie amplitudy (ASK — Amplitude Shift Keying)* — amplituda fali nosnej jest zmieniana pomiędzy różnymi stałymi wartościami, aby przenieść dane cyfrowe. Dla danych binarnych używane są dwa poziomy napięcia sygnału: jeden dla „0”, drugi dla „1”. Rysunek 3.2 przedstawia kluczowanie ze zmianą amplitudy.

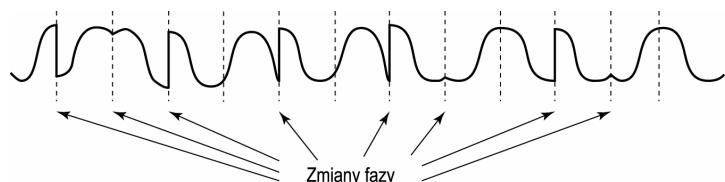
Rysunek 3.2.
Kluczowanie amplitudy



- ◆ *Kluczowanie częstotliwości (FSK — Frequency Shift Keying)* — częstotliwość fali nosnej jest zmieniana pomiędzy stałymi wartościami. Dla danych binarnych stosowane są dwie częstotliwości: dla „0” i „1”. Rysunek 3.3 przedstawia kluczowanie ze zmianą częstotliwości.

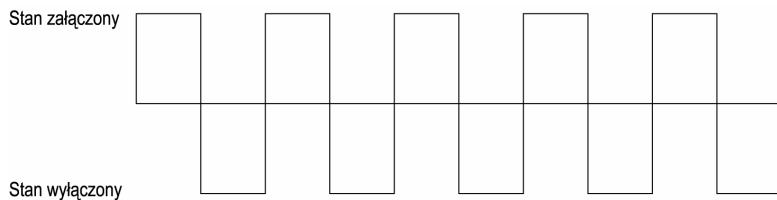
Rysunek 3.3.*Kluczowanie częstotliwości*

- ◆ *Kluczowanie fazy (PSK — Phase Shift Keying)* — na początku okresu faza fali nosnej jest zmieniana pomiędzy wartościami dyskretnymi. Dla danych binarnych fala nosna jest przesuwana systematycznie o 45, 135, 225 i 315 stopni w równomiernych odstępach czasu. Kazda zmiana fazy przenosi 2 bity danych. Kluczowanie fazy jest przedstawione na rysunku 3.4.

Rysunek 3.4.*Kluczowanie fazy*

Transmisja cyfrowa

Transmisja cyfrowa to wymiana pomiędzy komputerami danych w postaci dyskretnych jednostek — jedynek i zer. Sygnały reprezentujące stany dyskretne, przedstawione na rysunku 3.5, noszą nazwę sygnałów cyfrowych. Zmiany stanów dyskretnych są praktycznie natychmiastowe.

Rysunek 3.5.*Sygnały cyfrowe*

W transmisji cyfrowej komputery, które chcą się ze sobą komunikować, muszą ustalić wspólny format danych. Sposób, w jaki dane binarne są formatowane, nosi nazwę *modulacji impulsowo-kodowej* (PCM — Pulse Code Modulation). Do formatów PCM należą NRZ-L, NRZ-M, NRZ-S, Bi-Phase-L, Bi-Phase-M, Bi-Phase-S, DBi-Phase-M i DBi-Phase-S.

Błędы występujące w procesie transmisji PCM można wykryć za pomocą metody *sum kontrolnych parzystości*. W tej metodzie do każdej porcji przesyłanych danych dodawana jest *suma kontrolna*, która może być parzysta lub nieparzysta. W *parzystej sumie kontrolnej* dodawane jest zero lub jedynka, tak aby całkowita liczba jedynek była parzysta. W *nieparzystej sumie kontrolnej* dodawane jest zero lub jedynka, tak aby całkowita liczba jedynek była nieparzysta.

Technologie i mechanizmy transmisji

Komunikacja w sieci może wykorzystywać różne technologie i mechanizmy transmisji. Do przesyłania sygnałów, zarówno analogowych, jak i cyfrowych, służą dwie techniki:

- ◆ *Transmisja w pasmie podstawowym* — ten typ transmisji stosuje sygnalizacje cyfrowa na jednej częstotliwości. Pojedynczy kanał zajmuje całe pasmo nosnika. Kazde urządzenie w sieci z transmisją w pasmie podstawowym wysyła sygnały w dwóch kierunkach. Ten system stosuje w regularnych odstępach odległości regeneratory do przywracania oryginalnego poziomu sygnału.
- ◆ *Transmisja szerokopasmowa* — ten typ transmisji stosuje sygnały analogowe. Jego funkcjonalność pozwala podzielić całe dostępne pasmo na wiele kanałów. Ponieważ każdy kanał może przekonywać osobny sygnał analogowy, sieć szerokopasmowa umożliwia wiele równoczesnych transmisji przez pojedynczy kanał. W transmisji szerokopasmowej przepływ sygnału jest jednokierunkowy. System szerokopasmowy do przywracania oryginalnego poziomu sygnału stosuje wzmacniacze, rozmieszczone w stałych odległościach jeden od drugiego. Ponieważ przepływ sygnału jest jednokierunkowy, wymagane są dwie ścieżki przepływu danych, aby sygnał dotarł do wszystkich urządzeń. Powszechnie stosowane są dwie metody tworzenia dwóch tras przepływu danych:
 - ◆ *Podział pasma* — pasmo dzielone jest na dwa kanały, każdy o innym zakresie częstotliwości. Jeden kanał służy do wysyłania, drugi do odbierania sygnałów.
 - ◆ *Dwuprzewodowa transmisja szerokopasmowa* — do każdego urządzenia przyłączone są dwa kabły: jeden do wysyłania i jeden do odbioru sygnałów.

Komutacja obwodów

Mechanizm *komutacji obwodów* (inaczej *zorientowany na połączenie*) wymaga dedykowanego połączenia (obwodu) pomiędzy dwoma punktami końcowymi komunikacji. Ten mechanizm stosowany jest przy analogowej transmisji sygnałów, na przykład w telefonii. W tym systemie podczas inicjacji połączenia z telefonu nadawcy zestawiany jest obwód od telefonu nadawcy przez lączne międzymiastowe do odległej centrali i w koncu do telefonu odbiorcy. Taki sam mechanizm stosowany jest w niektórych sieciach komputerowych, które w roli medium transmisji wykorzystują lączna telefoniczne.

Komutacja komunikatów

Komutacja komunikatów nie wymaga tworzenia dedykowanego połączenia pomiędzy punktami końcowymi łączności. Komunikat dzielony jest na małe części, z których każda stanowi niezależna całość i zawiera dane adresu docelowego. Komunikaty są gromadzone w każdym przełączniku, zanim zostaną przesłane do kolejnego przełącznika na trasie. *Przelacznik* (ang. *switch*) to wyspecjalizowane urządzenie, służące do łączenia dwóch lub kilku linii transmisyjnych. W każdym przełączniku komunikaty odbierane są do buforów, kontrolowane na obecność błędów i wysypane ponownie. Aby zmagać się z komunikatami w buforach zanim będzie można je przesłać dalej, przełącznik potrzebuje wystarczającej pamięci. Sieci stosujące technikę komutacji komunikatów nazywane są inaczej sieciami typu *pamiętajaco-wysyłajacego* (*store-and-forward*).

W komutacji komunikatów rozmiar bloku komunikatu nie jest ograniczony, wobec tego dostarczenie komunikatu może się opóźnić, jeśli pojedynczy blok komunikatu „zatrzyma” linię komunikacyjną. Z tego powodu mechanizm komutacji komunikatów nie nadaje się do zastosowań czasu rzeczywistego, obejmujących np. łączność audio i wideo. Można go jednak stosować tam, gdzie pewien poziom opóźnień jest dopuszczalny — na przykład w systemach pracy grupowej, planowaniu i *workflow*.

Komutacja pakietów

W mechanizmie *komutacji pakietów* komunikaty dzielone są na segmenty zwane pakietami, które następnie przesyłane są przez sieć indywidualnie. Kazdy pakiet oprócz danych zawiera również informacje adresowe nadawcy i odbiorcy.

Chociaż komutacja pakietów wydaje się być podobna do komutacji komunikatów, między tymi dwoma mechanizmami jest jedna różnica. W komutacji komunikatów nie istnieje górna granica rozmiarów bloku komunikatu, zas w komutacji pakietów rozmiar pakietu jest ograniczony do ustalonej wartości, dzięki czemu metoda pozwala na przesył szybszy i wydajniejszy od komutacji komunikatów. Mechanizm komutacji pakietów pozwala ponadto urządzeniom przelaczającym zarządzac danymi pakietu w samej tylko pamięci, co eliminuje konieczność tymczasowego składowania przez te urządzenia danych na dysku.

Komutacja pakietów jest przydatna, gdy istnieje potrzeba przesyłania danych pomiędzy dwoma komputerami wiecej niż jednym kanałem. W komutacji pakietów można to osiągnąć bez stosowania odrebnego linii dla poszczególnych kanałów, ponieważ jeden kanał łączności może służyć do nadawania pakietów z kilku komunikatów — technika ta nosi nazwę *multipleksowania*. Istnieją dwa typy multipleksowania: *przez podział częstotliwości* (FDM — Frequency Division Multiplexing) oraz *przez podział czasu* (TDM — Time Division Multiplexing). W multipleksowaniu przez podział częstotliwości pasmo częstotliwościowe dzielone jest na kanaly logiczne. W multipleksowaniu przez podział czasu każdy użytkownik otrzymuje okresowo całe pasmo na własny użytek.

Dwie inne odmiany komutacji pakietów to *przekazywanie ramki* (ang. *frame relay*) oraz *komutacja komórek* (ang. *cell switching*).

- ♦ Mechanizm *przekazywania ramki* jest szybka wersja komutacji pakietów i nadaje się do dużych szybkości transmisji. W tym mechanizmie dane dzielone są na ramki, których długość może ulegać zmianie, w zależności od typu sieci.
- ♦ Mechanizm *komutacji komórek* działa na tej samej zasadzie co mechanizm komutacji pakietów, lecz omija ograniczenia multipleksowania przez podział czasu (TDM). W TDM urządzenia nadające i odbierające są synchronizowane, aby rozpoznawać te same szczeliny czasowe, przez co niektóre ze szczelin czasowych mogą zostać nie wykorzystane. W komunikacji komórek szczeliny czasowe są przydzielane w miarę potrzeb, dzięki czemu ich wykorzystanie jest zoptymalizowane.

Porównanie transmisji analogowej i cyfrowej

Transmisja analogowa jest stosowana w łączności od 100 lat. Jednakże transmisja cyfrowa, od chwili pojawienia się w 1962 roku, staje się coraz popularniejsza. Zalety transmisji cyfrowej w zestawieniu z analogową jest wiele; część z nich została omówiona poniżej.

- ♦ *Stopa błędów* — sygnał przesyłany medium analogowym stopniowo traci moc i może ulec zakłóceniom. Ta utrata mocy nosi nazwę *tlumienia*. Transmisja sygnałów cyfrowych ma bardzo niska stopę błędów. W prawdziwe obwody analogowe mogą zawierać wzmacniacze kompensujące tlumienie, lecz sygnałów nie można nigdy odtworzyć w pełni. Jeśli na długim odcinku zastosujemy wiele wzmacniaczy,

bledy beda sie kumulowac, zas sygnaly ulegna wyraznemu znieksztalceniu.

W przeciwnieństwie do nich, sygnaly cyfrowe przenosza tylko dwie wartosci: 0 i 1, dzieki czemu slabe sygnaly przesylane na daleki dystans mozemy odtworzyc do wartosci poczatkowej, unikajac kumulacji znieksztalcen.

- ♦ *Multipleksowanie* — mozna razem przesylac rózne typy informacji: glos, dane, muzyke i obrazy (np. telewizyjne, faksymile lub obrazy z videotelefonu).
- ♦ *Szybkosc transmisji* — objetosc danych, jaka mozna przeslac w ciagu sekundy. Cyfrowa transmisja sygnalow pozwala osiagnac wyzsze szybkosci transmisji.

Publiczne systemy lacznosci, na przyklad systemy telefonii, od poczatku korzystaly z analogowej transmisji sygnalow. Gdy jednak zalety transmisji cyfrowej staly sie oczywiste, zaczely takze z niej korzystac. Spadek cen komputerow i cyfrowych ukladow scalonych rowniez ma dodatni wpływ na wykorzystanie transmisji cyfrowej. Aby wiec zaspokoic rosnace zapotrzebowanie na transmisione danych, obrazow i sygnalow video, duza czesc ogólnoświatowych systemow telefonicznych zastapiono nowoczesnym systemem cyfrowym *ISDN* (ang. *Integrated Services Digital Network* — siec cyfrowa ze zintegrowanymi uslugami).

Nosniki fizyczne

Warstwa fizyczna moze korzystac z dowolnych mediów fizycznych:

- ♦ elektrycznych,
- ♦ mechanicznych,
- ♦ optycznych.

W tym rozdziale omówimy rózne nosniki fizyczne, sluzace do komunikacji.

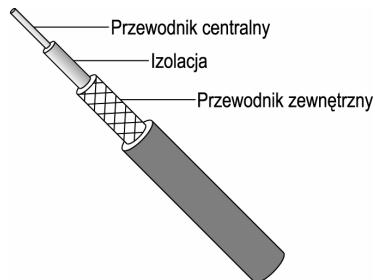
Kabel koncentryczny

Kabel koncentryczny (lub: wspolosiowy) jest najpowszechniej stosowanym typem kabla sieciowego, przede wszystkim dlatego, ze jest tani, lekki, elastyczny i latwy w eksploatacji. Kabel koncentryczny zawiera dwa przewodniki o wspolnej osi (patrz rysunek 3.6). Przewodnik centralny (drut lub linka miedziana) otoczony jest przez przewodnik zewnetrzny (*ekran*), sluzacy jako uziemienie i chroniacy przewodnik centralny przed zaklonciami elektromagnetycznymi (EMI — *Electromagnetic Interference*). Warstwa izolacji pomiedzy przewodnikami wewnetrnym i zewnetrnym utrzymuje staly dystans jednego od drugiego. Ostatni element — zewnetrzna koszulka z tworzywa sztucznego — ochrania kabel. Kable koncentryczne umozliwiaja predkosci transmisji od 10 Mb/s do 100 Mb/s. Stosowane sa dwa typy kabla koncentrycznego: gruby i cienki.

Cienki kabel koncentryczny

Cienki kabel koncentryczny (tzw. *thinnet* lub *thin Ethernet*) jest elastycznym kablem o srednicy okolo 6 mm. Moze sluzyc do przesyłania sygnalow na odleglosc ok. 185 metrow bez znaczacej utraty mocy. Kabel ten posiada impedancje falowa 50Ω , co oznacza, ze dla pradu zmiennego ma opornosc 50Ω . Cienki kabel koncentryczny nalezy

Rysunek 3.6.
Kabel koncentryczny



do rodziny kabli RG-58, które posiadają wewnętrzny przewodnik w postaci pojedynczego drutu lub linki miedzianej. Tabela 3.1 wymienia typy kabli koncentrycznych stosowanych powszechnie w sieciach.

Tabela 3.1. Typy kabli koncentrycznych

Kabel	Opis
RG-58 /U	Przewodnik centralny w postaci pojedynczego drutu miedzianego.
RG-58 A/U	Przewodnik centralny w postaci linki miedzianej.
RG 58 C/U	Wersja RG-58 A/U do zastosowań wojskowych.
RG-59	Stosowany w telewizji kablowej; impedancja falowa 75Ω .
RG-62	Stosowany w sieciach ARCnet.

Gruby kabel koncentryczny

Gruby kabel koncentryczny (inaczej *thicknet* lub *standard Ethernet*) jest sztywnym kablem o średnicy ok. 12 mm. Kabel ten pozwala na przesyłanie danych na duże odległości (około 500 metrów), lecz jest droższy od cienkiego.



Cienki kabel koncentryczny można podłączyć do grubego za pomocą nadajnika-odbiornika lub urządzenia o nazwie *jednostka sprzeżenia z nosnikiem* (MAU — *Media Attachment Unit*).

Osprzęt do połączeń kablem koncentrycznym

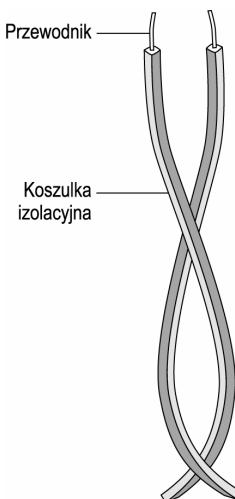
Cienki kabel koncentryczny wymaga stosowania odpowiednich elementów do łączenia kabla z komputerem. Noszą one nazwy *złącza BNC* (*British Naval Connector*) — złącze marynarki brytyjskiej). W standardzie BNC dostępnych jest kilka typów skladników:

- ◆ Złącze BNC montowane na kablu — przymocowane lub zaciszczone na koncu kabla.
- ◆ Trójkątnik BNC, który służy do łączenia karty sieciowej z kablem sieciowym.
- ◆ Złącze beczulkowe, które łączy ze sobą dwa oddzielne kabły koncentryczne w jeden długi.
- ◆ Terminator BNC, służący do zakończenia magistrali rezystancja równa impedancji faliowej kabla.

Skretka

Kabel typu skretka składa się z pary izolowanych przewodów miedzianych, skreconych ze sobą, jak na rysunku 3.7. Skrecenie przewodów zmniejsza:

Rysunek 3.7.
Skretka



- ◆ Tendencje do generowania przez kabel szumu o częstotliwościach radiowych, który mógłby zakłócać znajdujące się w pobliżu urządzenia elektroniczne i inne przewody. Redukcja zakłóceń powodowana jest tym, że emisje z obu przewodów znoszą się nawzajem.
- ◆ Podatność kabla na zakłócenia elektromagnetyczne.
- ◆ Przesłuch (czyli zjawisko mieszania sygnałów z jednego kabla z sygnałami pochodzącymi z kabla sąsiedniego).

Istnieją dwa typy skretki:

- ◆ *Skretka nieekranowana (UTP — Unshielded Twisted Pair)* — kabel UTP zawiera pary przewodów miedzianych, z których każda skrecona jest razem, aby zredukować zakłócenia elektromagnetyczne. Kabel UTP jest popularny w sieciach LAN. Jedna z przyczyn jego popularności jest fakt, iż stosowany jest powszechnie w istniejących systemach telefonicznych i wiele budynków biurowych jest już wyposażonych w odpowiednie instalacje. Skretka nieekranowana jest tania i pozwala na połączenia o długości około 100 metrów, z prędkościami transmisji od 10 Mb/s do 100 Mb/s. Jedna z głównych wad UTP jest podatność na przesłuch; możemy ją znacznie zmniejszyć, ekranując skretkę.
- ◆ *Skretka ekranowana (STP — Shielded Twisted Pair)* — kable STP posiadają metalowy ekran otaczający skrecone pary przewodów, który chroni przed zewnętrznymi zakłóceniami elektromagnetycznymi. Dzięki temu kable STP są mniej podatne na przesłuch niż UTP i pozwalają na większe prędkości transmisji. Kable STP są droższe od UTP i cienkiego koncentrycznego, lecz tanie od grubego koncentrycznego lub światłowodów. Długość połączeń dla kabli STP wynosi około 100 metrów, zas prędkości transmisji od 10 Mb/s do 100 Mb/s.

Elementy konstrukcyjne stosowane wraz ze skretka

Do laczenia komputera ze skretka stosowane jest zlaczce RJ. Sa one produkowane w różnych wersjach dla różnych typów skretki. Najlepiej znane zlaczce — RJ-11 — uzywane jest w telefonach. Zlaczce RJ wystarczy wetknac w odpowiednie gniazdo karty sieciowej.

Swiatlowody

Kable swiatlowodowe przesyłaja dane za pomocą impulsów światła zamiast sygnałów elektrycznych, wobec czego dane trzeba przetworzyć na impulsy świetlne. Do konwersji służy źródło światła, emitujące impulsy świetlne po przepuszczeniu prądu elektrycznego. Źródłem światła może być *dioda świecaca LED (Light Emitting Diode)* lub *dioda laserowa*. Impuls świetlny reprezentuje bit „1”, a jego brak — „0”. Detektor na drugim koncu światłowodu (*fotodioda*) odbiera sygnały świetlne i przekształca je z powrotem na sygnały elektryczne.

Jak widać na rysunku 3.8, kabel światłowodowy posiada rdzeń, z bardzo czystego szkła lub stopionego kwarcu, który może przepuszczać sygnały świetlne. Szklany płaszcz otaczający rdzeń ma niższą gestość od centralnego włókna, przez co sygnały świetlne pozostają w włókinie centralnym dzięki zjawisku całkowitego wewnętrznego odbicia. Płaszcz szklany otacza wzmacniające druty i koszulkę zewnętrzną z tworzywa sztucznego. Kable światłowodowe są droższe od elektrycznych, lecz pozwalają na większe przepustowości łączy i połączenia na dłuższe odległości.

Rysunek 3.8.
Kabel światłowodowy



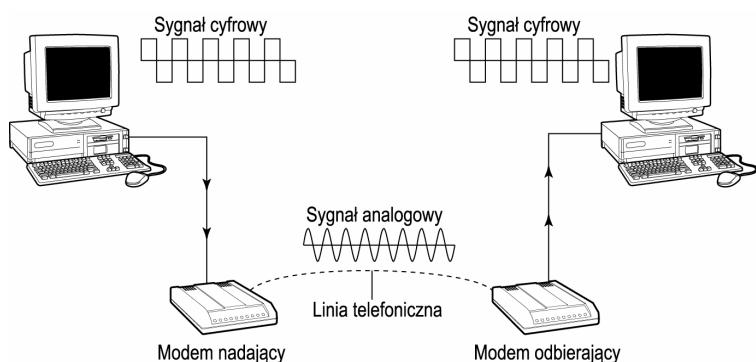
Trzy cechy decydują o przewadze kabli światłowodowych nad kablami elektrycznymi:

- ◆ *Przepustowość* — kable światłowodowe mają wyjątkowo wysoką przepustowość. Ponieważ jest w nich używane światło zamiast sygnałów elektrycznych (a światło przemieszcza się przedżej od prądu elektrycznego), objętość danych wysyłanych w jednostce czasu jest o wiele większa niż w przypadku kablów elektrycznych. Obecnie dostępne technologie pozwalają na prędkość transmisji od 100 Mb/s do 2 Gb/s.
- ◆ *Tłumienie* — kable światłowodowe mają niższe tłumienie od miedzianych. Segmente kabla światłowodowego mogą przesyłać sygnały na odległość mierzoną w kilometrach.
- ◆ *Zakłócenia elektromagnetyczne (EMI)* — kable światłowodowe są na EMI całkowicie niewrażliwe. Ponieważ kable te nie emisują sygnałów na zewnątrz, nie występuje zjawisko przesłuchu. Ponadto światłowodami trudno manipulować, więc są bardzo bezpieczne.

Modemy

Nazwa *modem* jest skrótem od *MODulator/DEModulator*. Urządzenia te służą do komunikacji pomiędzy różnymi sieciami za pomocą analogowego medium transmisji sygnałów, na przykład linii telefonicznej. Warstwa fizyczna wysyła dane binarne — lecz noszony analogowy może przesyłać jedynie sygnały analogowe. Aby w takiej sytuacji umożliwić komunikację, nadawane sygnały cyfrowe trzeba przekształcić na analogowe; podobnie po stronie odbiorcy — sygnały analogowe trzeba z powrotem przekształcić na cyfrowe. Po stronie nadawcy modem konwertuje sygnały cyfrowe na analogowe, aby przesłać je przez noszony analogowy; ten proces nosi nazwę *modulacji*. Po stronie odbiorcy modem przekształca sygnały analogowe z powrotem na dane cyfrowe; ten proces nazywany jest *demodulacją*. Rysunek 3.9 przedstawia sposób działania modemu.

Rysunek 3.9.
Sposób działania modemu



Modemy można instalować wewnątrz komputera lub na zewnątrz. Instalacja wewnątrz polega na włożeniu modemu do jednego z gniazd rozszerzających na płycie głównej komputera. Modem zewnętrzny jest małym urządzeniem we własnej obudowie, połączonym z komputerem. Wszystkie modemy wymagają do funkcjonowania następującego sprzętu:

- ◆ szeregowego interfejsu RS-322,
- ◆ interfejsu linii telefonicznej RJ-11.

W zależności od stosowanej metody transmisji, modemy dzielą się na dwie odrebnne kategorie:

- ◆ *Asynchroniczne* — stosują w komunikacji transmisje asynchroniczne. Inaczej mówiąc, dane dzielone są na szeregowy ciąg bajtów, z których każdy oddzielony jest od innych bitem startu i bitem stopu. Pomiedzy komputerami wysyłającym i odbierającym dane nie ma zadnej koordynacji, wobec czego komunikacja jest asynchroniczna.
- ◆ *Synchroniczne* — te modemy koordynują transmisje pomiędzy komputerami nadającym i odbierającym. W metodzie synchronicznej dane przesyłane są w formie ramek, nie zawierających bitów startu ani stopu. Do osiągnięcia synchronizacji i zapewnienia dokładności transmisji stosowane są specjalne znaki.

Korekcja błędów to mechanizm stosowany przez modemy w celu zapewnienia poprawności transmitowanych danych. Modemy zdolne do jego obsługi dzielą dane na małe

porcje zwane *ramkami*. Modem wysylajacy dane ustala podsumowanie wartosci kazdej ramki danych. To podsumowanie wartosci nosi nazwe *sumy kontrolnej* i jest dolaczane do kazdej wyslanej ramki. Modem odbierajacy również oblicza sume kontrolna dla kazdej ramki i porównuje ja z wartoscia sumy kontrolnej dolaczonej do ramki danych. Jesli obie wartosci nie sa identyczne, cala ramka zostaje przeslana ponownie.

Nosniki bezprzewodowe

Jak sama nazwa wskazuje, nosniki bezprzewodowe *nie* stosuja do przesyłania danych zadnych przewodów czy kabli. W wiekszosci przypadków nosnikiem transmisijs jest dla nich atmosfera ziemska. Jednakze siec uzywajaca nosnikow bezprzewodowych nie jest calkowicie uniezalezniona od okablowania. W sieciach zlozonych z różnych elementów urzadzenia bezprzewodowe komunikuja sie z siecia oparta na kablach.

Z uwagi na niezaleznosc od nosnikow fizycznych, technologie nosnikow bezprzewodowych rozwijaja sie bardzo szybko. Sa one szczególnie przydatne w sytuacjach, gdy nie opłaca sie laczyć elementów sieci kablami oraz gdy nie jest to mozliwe. Zazwyczaj siec komputerowa w pojedynczym budynku skonstruowana jest z nosnikow fizycznych — kabli elektrycznych lub światłowodowych. Lecz polaczenie dwóch sieci komputerowych w odrebnich budynkach w obrebie fabryki moze wymagac przeciagania kabli pod ulica, co jest przedsiemie kosztownym. W takiej sytuacji nosnik bezprzewodowy moze byc najlepszym rozwiazaniem.

Sieci bezprzewodowe, w zaleznosci od stosowanych technik transmisijs, dziela sie na trzy kategorie:

- ♦ *Sieci lokalne (LAN)* — bezprzewodowe sieci LAN korzystaja z czterech technik transmisijs:
 - ♦ podczerwien,
 - ♦ laser,
 - ♦ radio waskopasmowe (o pojedynczej częstotliwosci),
 - ♦ radio o pasmie rozproszonym.
- ♦ *Rozszerzone sieci lokalne* — zasieg sieci bezprzewodowych mozna zwięksyc przez zastosowanie wyspecjalizowanych urzadzen, na przykład mostów bezprzewodowych. *Most* stosuje technologie bezprzewodowa, na przykład radio o widmie rozproszonym, aby stworzyc sciezke transferu danych pomiedzy dwiema sieciami. Za pomoca tej metody mozna przesyłac zarówno głos, jak i dane.
- ♦ *Przenosny sprzęt komputerowy* — ta technologia pozwala osobom podróżujacym utrzymywac polaczenie z siecią komputerowa. Przenosny sprzęt komputerowy do wymiany sygnalów stosuje fale radiowe z zakresu publicznego i nalezace do sieci telefonicznych. Technologia ta korzysta z jednej z nastepujacych uslug:
 - ♦ *Packet radio* — dane sa dzielone na jednostki zwane pakietami, które nastepnie przesyłane sa do satelity i rozglaszane na określonym terenie. Komputery odbieraja te pakiety danych, które sa do nich zaadresowane.

- ◆ *Sieci komórkowe* — siec komórkowa jest rozszerzona bezprzewodowa siecia LAN, która korzysta z uslug firm telefonicznych. Siec ta nosi również nazwe *komórkowych pakietów danych cyfrowych* (CDPD — *Cellular Digital Packet Data*) i jest wystarczajaco szybka, aby umozliwiac transmisię w czasie rzeczywistym.
- ◆ *Systemy mikrofalowe* — system mikrofalowy sklada sie z dwóch radiowych nadajników-odbiorników, niezbednych do odbierania i wysyłania rozwiazan, oraz z dwóch anten kierunkowych, skierowanych na siebie wzajemnie. Anteny te nawiazuja lacznosc na podstawie sygnalów rozwiazanych przez nadajniki-odbiorniki.

Najczesciej stosowane sa techniki radiowe i podczerwien. Zostaly one szczegółowo omówione w ponizszych punktach.

Podczerwien

Technologia komunikacji w podczerwieni jest najpowszechniej stosowana w pilotach do telewizorów. Po kazdym nacisnieciu przycisku pilot wysyla serie podczerwonych impulsów, które niosą zakodowane informacje dla odbiornika w telewizorze.

Zasieg transmisiji w podczerwieni jest ograniczony do okolo 30 metrów. Szerokie pasmo fal podczerwonych pozwala na transfer danych z predkoscia do 10 Mb/s. Stosowane sa cztery typy technologii komunikacji w podczerwieni:

- ◆ *Szerokopasmowa z optycznym skupieniem wiazki* — stosuje technologie szerokopasmowa i spełnia wymagania wysokiej jakosci zastosowan multimedialnych.
- ◆ *W linii widzenia* — wymaga nie zastawionej niczym linii widzenia pomiedzy nadajnikiem i odbiornikiem.
- ◆ *Odbita* — komputery kieruja wszystkie transmisię w jeden punkt, z którego sa dalej kierowane do odpowiednich komputerów.
- ◆ *Rozproszona* — nadajniki rozwylaja sygnały, które w koncu trafiają do odbiornika po odbiciach od podlogi, scian i sufitu. Z uwagi na rozproszenie trasy, predkosci przesyłu danych sa niskie.

Radio

W transmisji radiowej nadajnik nie musi byc umieszczony w bezposrednim widoku odbiornika. I poniewaz fale radioowe odbijaja sie od ziemskej jonosfery, ich zasieg moze byc duzy. Stosowane sa dwa typy transmisiji radiowej:

- ◆ *Transmisja waskopasmowa* — ten typ nazywany jest inaczej *transmisja radiowa na jednej częstotliwości*, poniewaz nadajniki uzywaja pojedynczej częstotliwosci. Ten typ transmisiji nie wymaga ustawienia nadajnika i odbiornika na linii widzenia, a ponadto jej zasieg jest wiekszy niz w przypadku podczerwieni.
- ◆ *Transmisja w pasmie rozproszonym* — w tej technice transmisię odbywaja sie na wielu częstotliwosciami. Radio z pasmem rozproszonym jest powszechnie stosowane w sieciach rozleglych i moze stosowac jedna z ponizszych metod:

- ♦ *Przeskoki częstotliwości* — transmisja odbywa się przez przelaczanie pomiędzy kilkoma dostępnymi częstotliwościami i może działać poprawnie tylko wtedy, gdy nadajnik i odbiornik są synchronizowane. Ta metoda zapewnia przepustowość od 250 kb/s do 2 Mb/s.
- ♦ *Bezpośrednia modulacja sekwencji* — oryginalny komunikat jest dzielony na części zwane *chipami*, które są następnie przesyłane na odrębnych częstotliwościach. Ta metoda zapewnia przepustowość od 2 Mb/s do 6 Mb/s.

Najczęściej stosowane topologie

Topologia sieci definiuje struktury sieci. Topologie możemy podzielić na kategorie w dwóch dziedzinach:

- ♦ *Fizyczne* — schemat połączeń sieci i faktyczny rozkład kabli lub innych nosników.
- ♦ *Logiczne* — sposób, w jaki hosty (komputery, drukarki, skanery) uzyskują dostęp do nosnika (kabla) i jak przezeń się porozumiewają.

Jak widać, topologia sieci nie tylko określa, jaki typ sprzętu powinien zostać użyty, lecz również dostarcza wskazówek do implementacji sieci. Topologia określa także, jak komputery komunikują się w sieci. Komputery uzyskują dostęp do nosnika stosując metody dostępu — zbiór regul, decydujących o sposobie współużytkowania nosnika transmisji.

Wybór określonego typu topologii może mieć wpływ na wymogi dotyczące sprzętu i oprogramowania, zarządzanie i rozwój sieci. Przed podjęciem decyzji dotyczącej wyboru topologii dla naszej sieci musimy rozważyć następujące czynniki:

- ♦ budżet sieci,
- ♦ rozmiary sieci,
- ♦ wymagany poziom bezpieczeństwa,
- ♦ fizyczny rozkład sieci,
- ♦ typ działalności użytkowników,
- ♦ natężenie ruchu w sieci.

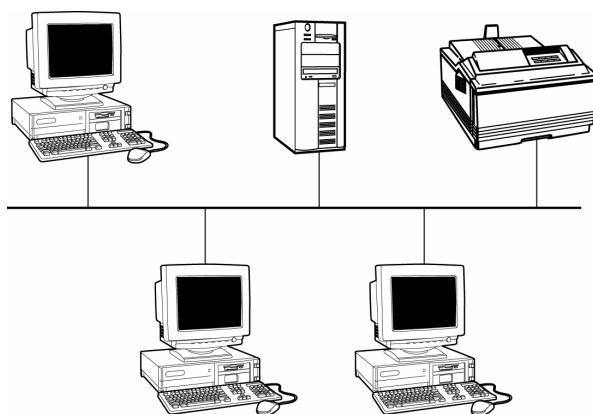
Do najczęściej stosowanych topologii należą magistrala z rozgłoszeniami (ang. *broadcast bus*), magistrala z przekazywaniem zetona (*token bus*), pierścień z przekazywaniem zetona (ang. *token ring*), FDDI (*fiber distributed data interface* — złącze danych w rozproszonych sieciach światłowodowych) oraz ATM (*asynchronous transfer mode* — tryb transferu asynchronicznego). Poniżej punkty omawiają te topologie dokładniej.

Magistrala

W topologii magistrali wszystkie komputery w sieci korzystają z jednego wspólnego kanału komunikacyjnego, nazywanego *szkieletem (backbone)* lub *magistralą (trunk line lub bus)*. Szkielet może być liniowy lub mieć postać drzewa. Rysunek 3.10 przedstawia topologię magistrali.

Rysunek 3.10.

Siec o topologii magistrali



Sieci posiadajace topologie magistrali moga rozglaszac komunikaty w obu kierunkach lub w jednym określonym. Siec magistralowa musi po obu koncach magistrali posiadac specjalne złącze, zwane terminatorem, które zapobiega odbiciom sygnalu od konców kabla, powodujacym zakłócenia.

Topologia magistrali ma nastepujace zalety:

- ◆ prosta, w bardzo malych sieciach niezawodna, łatwa w użyciu i łatwa do zrozumienia,
- ◆ zużywa się najmniej kabla, by połączyc ze sobą komputery,
- ◆ łatwa w rozbudowie,
- ◆ najtansza (w porównaniu z innymi topologiami).

Topologia ta ma również wady:

- ◆ Nie sprawuje się dobrze przy dużym obciążeniu. W przypadku intensywnego ruchu losowe transmisje danych z komputerów w sieci magistralowej mogą prowadzić do przerw, powodowanych równoczesnymi transmisjami.
- ◆ Zbyt wiele odprowadzeń z magistrali może osłabić sygnał elektryczny.
- ◆ Znajdowanie problemów może być trudne.

W zależności od mechanizmu komunikacji w sieci, topologie magistrali można podzielić na rozgłoszeniowe i stosujące przekazywanie zetona.

Magistrala rozgłoszeniowa

W topologii magistrali rozgłoszeniowej (ang. *broadcast bus*) nie tylko wszystkie komputery korzystają z jednego wspólnego kanalu komunikacyjnego, lecz również wszystkie nadajniki-odbiorniki odbierają wszystkie transmisje w sieci. *Nadajnik-odbiornik* (ang. *transceiver*) jest urządzeniem, które odbiera i wysyła sygnały przez nosnik. Do sterowania działaniem nadajnika-odbiornika wymagany jest adapter (interfejs) komputera macierzystego. *Interfejs komputera macierzystego* (inaczej *interfejs hosta*) jest połączony do magistrali komputera (na płytce głównej) oraz do nadajnika-odbiornika.

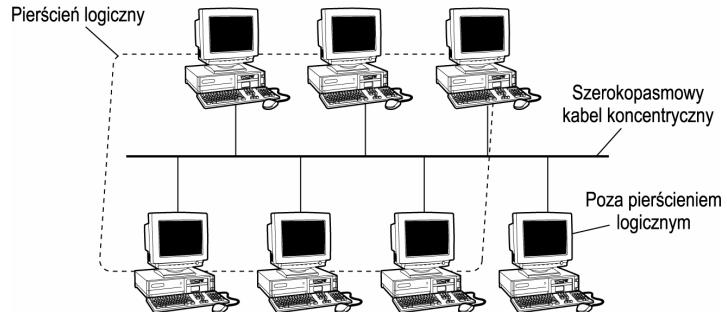
Gdy komputer nadaje dane, pakiety są rozglaszane do wszystkich nadajników-odbiorników. Kazdy z nich z kolei przesyła pakiety do interfejsu hosta, który wybiera wszystkie pakiety zaadresowane do swojego komputera i odrzuca pozostałe. W tym mechanizmie urządzenie nie przesyła do nadawcy żadnych informacji o odbiorze pakietu. Pakiety wysyłane do wyłączonego komputera są tracone, a nadawca nie jest o tym powiadamiany.

Topologia magistrali rozgłoszeniowej stosuje w komunikacji metodę dostępu *CSMA/CD* (*Carrier Sense Multiple Access/Collision Detection* — wykrywanie wielokrotnego dostępu do nosnika i wykrywanie kolizji). W tej metodzie, gdy wystąpi kolizja podczas nadawania danych przez komputer, dane są wysyłane ponownie po upływie losowego odcinka czasu.

Magistrala z przekazywaniem zetonu (token bus)

Ta topologia w łączności stosuje metodę *dostępu z przekazywaniem zetonu*. W tej metodzie każdy komputer w sieci zna adres komputerów po swojej lewej i prawej stronie. Pojedyncza ramka, nazywana *zetonem (token)*, kąry po sieci po trasie logicznego pierścienia, jak na rysunku 3.11. Tylko komputer posiadający właśnie zeton ma prawo nadawac dane. Po zakończeniu transmisji komputer przekazuje zeton do następnego komputera w sieci.

Rysunek 3.11.
*Pierścień logiczny
w magistrali
z przekazywaniem
zetonu*

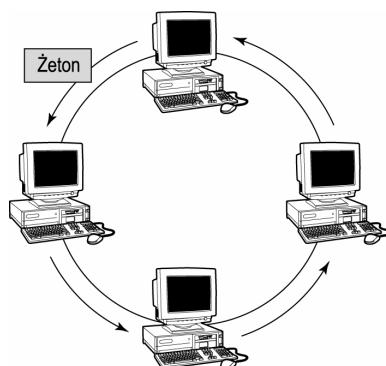


Token Ring

Topologia Token Ring stosuje w komunikacji metodę dostępu z przekazywaniem zetonu. W topologii pierścienia komputery przyłączone są do pojedynczej petli kabla. W przeciwieństwie do topologii magistrali, w topologii pierścienia z przekazywaniem zetonu nie występuje konieczność stosowania terminatorów na końcach kabla. Każdy komputer jest połączony z sąsiadami po dwóch stronach, jak na rysunku 3.12. Sygnały podróżują po petli tylko w jednym kierunku, przechodząc kolejno przez każdy komputer. Każdy z komputerów posiada odbiornik i nadajnik, i każdy pełni funkcje regeneratora wzmacniającego sygnał przekazywany do następnego komputera. Ponieważ sygnał jest regenerowany w każdym komputerze, jego stopień zniekształcenia jest niski. Ponieważ jednak awaria jednego komputera w topologii pierścienia może spowodować awarie całej sieci, fizyczna topologia pierścienia jest używana bardzo rzadko. Najczęstszym zastosowaniem pierścienia są topologie logiczne.

Rysunek 3.12.

Topologia pierscienia z przekazywaniem zetonu (token ring)



Siec o topologii pierscienia z przekazywaniem zetonu ma nastepujace zalety:

- ◆ Nawet przy duzym obciążeniu sieci, jej wydajność może być zbliżona do 100%.
- ◆ Wszystkie komputery mają równie szanse dostępu do sieci.

Ta siec ma tez wady:

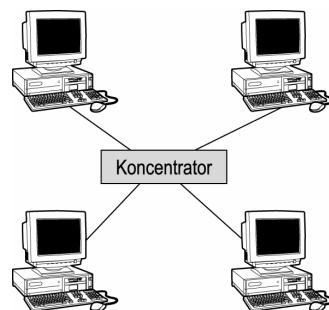
- ◆ Awaria jednego komputera w pierscieniu wpływa na całą sieć.
- ◆ Znajdowanie problemów w sieci token ring jest trudne.
- ◆ Operacja dodawania lub usuwania komputera powoduje przerwę w działaniu sieci.

Gwiazda

W topologii gwiazdy, przedstawionej na rysunku 3.13, poszczególne komputery są przyłączone do centralnego urządzenia zwanego koncentratorem. Jego funkcje mogą pełnić hub, przełącznik lub komputer. Kazdy kabel, łączący komputer z koncentratorem, jest identyfikowany przez unikatowy numer. Gdy dane przesyłane są z określonego komputera do komputera docelowego, dane przechodzą od nadawcy przez koncentrator do odbiorcy.

Rysunek 3.13.

Topologia gwiazdy



Topologia gwiazdy ma jedną podstawową przewagę nad topologiami magistrali i pierscienia. Po odczepieniu jednego komputera od koncentratora, reszta sieci działa nadal bez żadnych przeszkód. Ponieważ jednak każdy komputer musi być indywidualnie połączony do koncentratora, na sieć o topologii gwiazdy zużywa się więcej kabla.

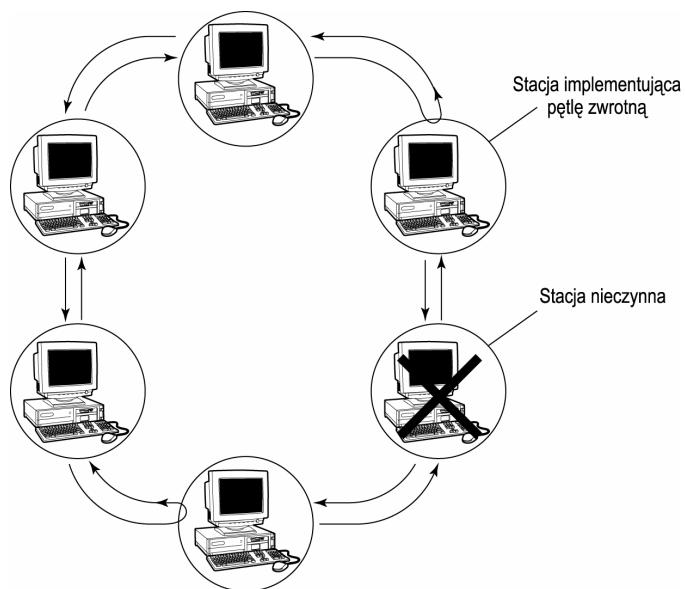
FDDI

Topologia sieci światłowodowych FDDI (ang. *Fiber Distributed Data Interface*) jest podobna do topologii pierscienia z przekazywaniem zetona. Podobnie jak sieci token ring, FDDI tworzy obieg danych, który zaczyna się w jednym komputerze, przechodzi przez wszystkie pozostałe i kończy się z powrotem w źródle. I podobnie jak w sieciach token ring, FDDI do komunikacji wykorzystuje metodę dostępu z przekazywaniem zetona. Taka metoda dostępu daje wszystkim komputerom w sieci równe szanse dostępu.

Miedzy sieciami FDDI a token ring istnieja dwie zasadnicze różnice. Po pierwsze, w przeciwienstwie do sieci token ring stosujacych kable elektryczne, FDDI wykorzystuja swiatlowody szklane i przesyłaja dane zakodowane w postaci impulsow swiatla. Po drugie, siec FDDI posiada zdolnosc samoczynnej naprawy, poniewaz potrafi wykrywac i naprawiac bledy. Dzieki temu urzadzenia FDDI moga automatycznie reagowac na awarie.

Siec FDDI sklada sie z dwóch niezależnych pierscieni; każdy komputer podłączony jest do obu pierscieni, jak na rysunku 3.14. Zastosowanie dwóch niezależnych pierscieni pozwala na automatyczne przywracanie funkcjonowania po awarii. Ruch sieciowy w obu pierscieniach odbywa się w przeciwnych kierunkach.

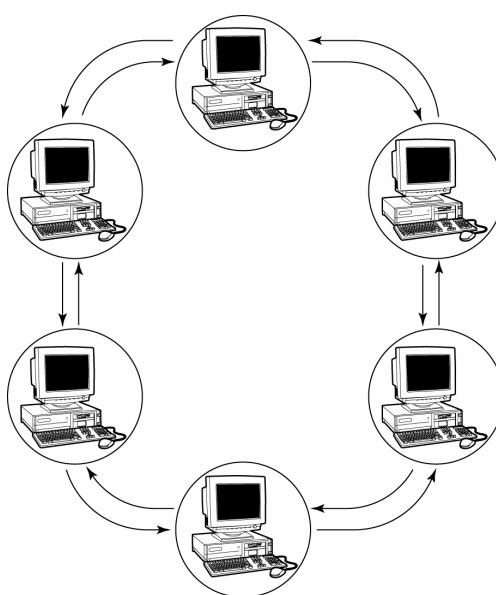
Rysunek 3.14. Sieć FDDI



Gdy w sieci nie ma zadnych uszkodzen, FDDI funkcjonuje dokladnie tak samo, jak siec token ring, wykorzystujac tylko jeden pierscien. Jednakze w razie awarii, na przyklad uszkodzenia interfejsu sieciowego hosta, drugi pierscien (zapasowy) uzywany jest do ominiecia punktu uszkodzenia. W sytuacji awarii sprzutowej, FDDI, aby umozliwic komunikacje pomiedzy pozostalymi komputerami, automatycznie „zawija” trase przesypania danych do pierscienia zapasowego, w którym ruch odbywa sie w przeciwnym kierunku. Rysunek 3.15 przedstawia siec FDDI, w ktorej jeden interfejs hosta jest niewykonny, co spowodowalo zastosowanie petli zwrotnej.

Rysunek 3.15.

*Siec FDDI
z uszkodzeniem*

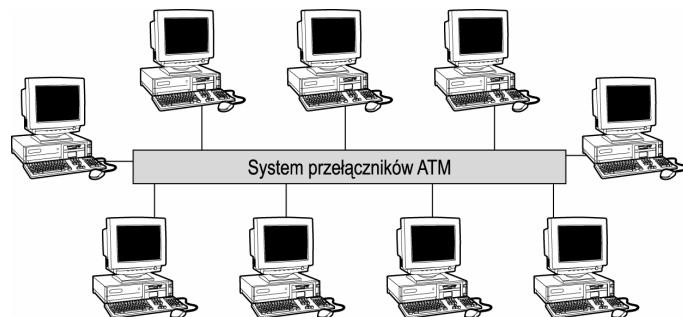


Sieci ATM

ATM (Asynchronous Transfer Mode — tryb przesyłania asynchronicznego) jest technologia sieciowa o dużych predkosciach transmisji, zorientowana na polaczenie. Sieci ATM, jak ta na rysunku 3.16, posiadaja topologie oczkowa, w której kazdy komputer jest polaczony z wszystkimi pozostalymi lub ich wiekszoscia. Szybkie sieci pozwalaja na przesyłanie danych z predkoscia 100 Mb/s i wiecej. Sieci ATM moga komutowac dane z predkosciami rzedu gigabitow na sekunde. Aby osiagnac takie parametry, sieci te wymagaja stosowania zlozonego sprzetu, przez co sa drozsze od innych technologii sieciowych.

Rysunek 3.16.

Siec ATM



Siec ATM uzyskuje duze predkosci transmisji dzieki zastosowaniu specjalizowanego sprzetu i technik programistycznych, do których zaliczaja sie:

- ◆ szybkie przełączniki, laczace komputery i inne przełączniki ATM,
- ◆ kable swiatlowodowe nie tylko pomiedzy przełącznikami ATM, lecz również laczace komputery z przełącznikami,

- ◆ ramki o stałych rozmiarach, tzw. *komórki* (ang. *cell*) o długości 53 bajtów: 5 bajtów nagłówka i 48 bajtów danych, które mogą być szybko przetwarzane przez przełączniki ATM.

ATM stosuje technologie połączeniowa zamiast komutacji pakietów. Komputer, który chce wysłać komórkę do innego komputera zdalnego, musi współpracować z przełącznikiem ATM, aby podać adres docelowy. Ta interakcja przypomina nawijazywanie połączenia telefonicznego. Komputer macierzysty czeka, aby przełącznik ATM skontaktował się z systemem zdalnym i ustalił trasę. Jeśli z jakiegoś powodu połączenia nie da się nawiazać (na przykład, jeśli zdalny komputer odrzuca zadanie lub nie odpowiada, albo też przełącznik ATM nie ma w danej chwili dostępu do zdalnego komputera), zgłoszane przez komputer macierzysty zadanie połączenia nie może zostać zrealizowane.

Z drugiej strony, gdy połączenie jest nawiazane, przełącznik ATM wysyła do komputera macierzystego identyfikator tego połączenia oraz komunikat o pomyslnym nawijazniu połączenia. Komputer macierzysty następnie wykorzystuje ten identyfikator połączenia do wysyłania i odbierania komórek. Taki typ ścieżki połączenia nosi nazwę *obwodu wirtualnego*.

Gdy komputer macierzysty przestaje potrzebować połączenia, wówczas zada od przełącznika ATM jego zerwanie, po czym przełącznik rozłącza komputery.

Rozdział 4.

Warstwa interfejsu sieciowego

W tym rozdziale:

- ◆ Warstwa interfejsu sieciowego — omówienie
- ◆ Standardy sterowania dostęmem do nosnika
- ◆ Odwzorowanie adresów fizycznych na adresy IP

TCP/IP można odwzorować na czterowarstwowy model DARPA. Model ten został opracowany przez biuro rządu USA DARPA (*Defense Advanced Research Project Agency* — biuro zaawansowanych obronnych projektów badawczych). Warstwa interfejsu sieciowego (*Network Interface Layer*), zwana również warstwą dostępu do sieci (*Network Access Layer*), jest druga warstwa tego modelu. Odpowiada ona części warstwy fizycznej lub kompletnej warstwie łączącej danych modelu odniesienia OSI.

Bieżący rozdział przedstawia warstwę interfejsu sieciowego i role, jaka odgrywa w transmisji danych. Dane przesyłane są w postaci małych porcji, które w każdej warstwie noszą określona nazwę. W warstwie interfejsu sieciowego porcje noszą nazwę ramek (*frame*), lecz ogólna nazwa dla tych porcji we wszystkich warstwach to *pakiety*. Niniejszy rozdział omawia takie zagadnienia, jak zawartość pakietów i najpopularniejszy typ ramki (Ethernet), różnorodne standardy sterowania dostęmem do nosnika, odpytywanie stosowane w ATM i ARCnet oraz przekazywanie zetona w Token Ring. Scharakteryzowane zostały także sposoby odwzorowania podczas transmisji fizycznych adresów komputerów na odpowiadające im adresy IP za pomocą ARP. W szybko zdobywających popularność sieciach ATM adresy IP hostów sieciowych są mapowane na adresy fizyczne za pomocą ATMAP.

Warstwa interfejsu sieciowego — omówienie

Adresy MAC, sterowniki kart sieciowych i określone interfejsy kart sieciowych funkcjonują w warstwie interfejsu sieciowego. Choć ta warstwa zajmuje się przede wszystkim komunikacją z kartami sieciowymi i innym sprzętem sieciowym, nie są obecne w tej warstwie funkcje protokołu IP. Oznacza to, że warstwa internetowa nie

moze korzystac z zadnych uslug potwierdzen lub sekwencjonowania, ktore moglyby istniec w warstwie interfejsu sieciowego. Lacznosc moglaby nie byc wiarygodna. Wobec tego wiarygodnosc komunikacji zapewniaja warstwy powyzej warstwy interfejsu sieciowego.



Chociaz warstwa interfejsu sieciowego odpowiada czesci warstwy fizycznej w modelu odniesienia OSI, nie bierze udzialu w samej transmisji danych.

Urzadzenia sieciowe skojarzone z warstwa interfejsu sieciowego to:

- ♦ NIC (karty interfejsu sieciowego),
- ♦ mosty,
- ♦ koncentratory inteligentne.

Do podstawowych obowiazkow tej warstwy naleza:

- ♦ identyfikacja wezelow (komputerow) w sieci,
- ♦ organizacja odebranych z nosnika sieci bitow w logiczne grupy, zwane ramkami, oraz nadzór nad rozmiarami ramek,
- ♦ przekształcanie adresów IP na adresy LAN,
- ♦ sterowanie przeplywem danych,
- ♦ pakowanie i nadawanie danych wychodzacych,
- ♦ wykrywanie bledow (lecz bez korygowania),
- ♦ udostepnianie uslug (na przyklad jakosci uslug — *Quality of Service*) i zdolnosci do adresowania (*unicast*, adresowanie grupowe i rozgloszenia) warstwie internetowej (zwanej czasem warstwa miedzysieciowa).

TCP/IP zostal zaprojektowany tak, by byl niezalezny od różnic w metodach dostepu do sieci, formatach ramek, czy tez nosnikach transmisji. Dzieki temu TCP/IP moze sluzyc do laczenia mieszanych typow sieci — technologii obejmujacych sieci lokalne (jak np. Ethernet czy tez Token Ring) oraz rozlegle (na przyklad X.25 i Frame Relay). Brak zaleznosci od okreslonej technologii sieciowej pozwala na latwa adaptacje TCP/IP dla nowych technologii, takich jak ATM (*Asynchronous Transfer Mode* — asynchroniczny tryb transmisji).

Z technologii wymienionych powyzej najpowszechniej uzywany jest Ethernet. Komputery podlaczone do sieci Ethernet do przesyłania miedzy soba danych uzywaja protokołów wysokiego poziomu, takich jak TCP/IP. Pakiety takich protokołów sa przesylane pomiedzy komputerami w postaci ramek Ethernet.

Zawartosc ramki Ethernet

Jak juz wspomniano, pakiety danych w warstwie interfejsu sieciowego nosza nazwe ramek. W systemie Ethernet urzadzenia komunikuja sie ze sobą za pomoca *ramek Ethernet*. Ramka taka sklada sie z ciagu bitow zorganizowanych w pola, do których naleza różne pola adresów, pole danych i pole kontroli bledów, które nadzoruje poprawnosc

danych zamkniętych w ramce. Pole danych może mieć rozmiary od 46 do 1500 bajtów. Rysunek 4.1 przedstawia jeden z bardziej popularnych formatów ramek Ethernet — IEEE 802.3.

Rysunek 4.1.

Format ramki Ethernet
IEEE 802.3

	6	2	1	1	1	Zmienna długość	4	
	Adres docelowy	Adres źródłowy	Długość	DSAP	SSAP	Bajt kontrolny	Dane	FCS



Obecnie dostępnych jest wiele wersji ramek Ethernet, miedzy innymi: IEEE 802.3 (tzw. standardowy Ethernet), DIX Ethernet (DEC/Intel/Xerox) — inaczej nazywany Version II Ethernet lub Ethernet_II, IEEE 802.3 SNAP (inaczej Ethernet_SNAP) i tak dalej. Należy zwrócić uwagę, iż wersje te niekoniecznie są ze sobą nawzajem zgodne.

Specyfikacja IEEE 802.3 obejmuje 14-bajtowy nagłówek lacza danych (*Data Link*), po którym następuje 3-bajtowy nagłówek sterowania laczem logicznym (LLC — *Logical Link Control*). Nagłówek lacza danych podaje adres wezła docelowego i nadawcy oraz długość danych w ramce. Nagłówek sterowania laczem logicznym wskazuje na bufor pamięci wezła odbierającego, w którym dane będą składowane — co pozwala wyższym warstwom łatwo lokalizować dane. Następnie są pola zawierające dane użytkownika i ciąg kontrolnego ramki (FCS — *Frame Check Sequence*).



Ethernet_II i Ethernet_802.3 (inaczej Novell Proprietary) nie posiadają nagłówka LLC. Ethernet_SNAP, podobnie jak implementacja IEEE 802.3, zawiera nagłówek LLC, a oprócz tego 5-bajtowy nagłówek protokołu dostępu do podsieci (SNAP — Sub-Network Access Protocol), zajmujący bajty od 18 do 22.

Do pół ramki Ethernet IEEE 802.3 należą:

- ◆ *Ciąg wstępny* — niezależnie od używanego typu ramki, sygnały we wszystkich sieciach Ethernet są kodowane na taki sam sposób — za pomocą kodu Manchester. Skuteczność tej metody wymaga spełnienia dwóch warunków: synchronizacji wewnętrznych zegarów wszystkich komputerów w sieci Ethernet oraz wstępniego ustalenia czasu trwania każdego bitu podczas transmisji. Oba te warunki pozwala spełnić ciąg wstępny (*preamble*), czyli sekwencja jedynek i zer poprzedzającą faktyczną ramkę Ethernet. Ciąg wstępny składa się z osmiu bajtów, zawierających na przemian zera i jedynki, i kończy się dwiema jedynkami. Gdy komputer w sieci Ethernet nadaje ramkę, pozostałe komputery używają ciągu wstępnego do synchronizacji z wewnętrznym zegarem nadawcy. Ponieważ synchronizacja zajmuje jakiś czas, pierwsze bity ciągu wstępnego są zasadniczo tracone. Po osiągnięciu synchronizacji między nadawcą i odbiorcą, odbiorca czeka na sekwencję „11”, która oznacza, iż nastepna będzie ramka Ethernet. Ponieważ ciąg wstępny służy jedynie do synchronizacji komputerów, jego bity nie są wprowadzane do bufora w pamięci karty sieciowej.
- ◆ *Adres docelowy* — pierwsze szesc bajtów ramki, o numerach od 0 do 5, składa się na adres wezła (komputera) docelowego. Niezależnie od typu ramki Ethernet, format adresu docelowego pozostaje taki sam we wszystkich implementacjach. Gdy wszystkie bity adresu docelowego mają wartość 1, wówczas komunikat jest typu rozgłoszeniowego i zostaje odebrany przez wszystkie wezły w segmencie.



- ◆ *Adres zródłowy* — następne szesc bajtów ramki, o numerach od 6 do 11, podaje adres wezla nadawcy. Podobnie jak dla adresu docelowego, format adresu zródłowego pozostaje taki sam we wszystkich implementacjach Ethernetu.
- ◆ *Dlugosc* — następne dwa bajty o numerach 12 i 13 określają długosć ramki danych, z pominieciem ciągu wstępnego, 32-bitowej wartości CRC, adresu DLC i samego pola Dlugosc.

Minimalna długosć ramki Ethernet wynosi 64 bajty, zaś maksymalna 1518 bajtów.



- ◆ *Punkt dostępu do usługi docelowej (DSAP — Destination Service Access Point)* — następny bajt, o numerze 15, wskazuje adres bufora pamięci, w którym wezel odbiorcy powinien składować odebrane dane. Pole to gra ważną rolę w wezłach posiadających wiele stosów protokołów.
- ◆ *Punkt dostępu do usługi zródłowej (SSAP — Source Service Access Point)* — następny bajt, o numerze 16, wskazuje adres procesu wysyłającego ramkę.
- ◆ *Bajt kontrolny* — następny bajt, o numerze 17, podaje typ ramki LLC.
- ◆ *Dane* — od 43 do 1497 bajtów to dane użytkownika. Długość tego pola jest zmieniona.
- ◆ *Ciąg kontrolny ramki (FCS — Frame Check Sequence)* — ostatnie cztery bajty nazywane są też bajtami cyklicznej kontroli nadmiarowej (CRC — Cyclic Redundancy Check). Gdy odbiorca otrzymuje pakiet, wówczas oblicza jego sumę kontrolną za pomocą złożonego wielomianu, a następnie porównuje otrzymany wynik z czterema ostatnimi bajtami ramki. Jeśli sumy kontrole nie zgadzają się ze sobą, ramka zostaje uznana za uszkodzoną i odrzucona. Zapewnia to wykrywanie błędów transmisji i zachowanie poprawności odebranych ramek.

Typowe składniki pakietu sieciowego

Pakiet jest blokiem danych, wyslanym przez sieć. W różnych warstwach nosi on różne nazwy:

- ◆ W warstwie fizycznej pakiet nosi nazwę *bitów*.
- ◆ W warstwie interfejsu sieciowego pakiet nazywany jest *ramką*.
- ◆ W warstwie internetowej pakiet nazywany jest *datagramem*.
- ◆ W warstwie transportowej pakiet nazywany jest *segmentem*.
- ◆ W warstwie aplikacji pakiet nosi nazwę *komunikatu*.

Podczas transmisji każda warstwa dodaje własne dane do pakietu, który pochodzi z warstwy aplikacji. Lecz niezależnie od warstwy, każdy pakiet zawiera podobne składniki:

- ◆ *Nagłówek* — zawiera adres zródłowy, adres docelowy i typ ramki. *Adres zródłowy* oznacza adres wezła, z którego pochodzi pakiet. *Adres docelowy* zawiera adres wezła, który będzie przetwarzac informacje zawarte w pakiecie. Nagłówek zawiera ponadto dwubajtowe pole *typ ramki*, dzięki któremu pakiet sam sie

identyfikuje. Oznacza to, iż po dotarciu pakietu do wezła przeznaczenia, system operacyjny wezła używa pola typu ramki do identyfikacji oprogramowania protokołu, które będzie przetwarzac pakiet. Pole typu ramki pozwala na jednoczesna obsługę i używanie przez wezel wielu protokołów.

- ◆ *Dane (inaczej informacje)* — to pole zawiera zdefiniowane przez użytkownika dane, które należy przesłać siecią z jednego wezła do drugiego. Długość tego pola jest zmieniącą się, w granicach określonych przez używany protokół.
- ◆ *CRC (lub FCS)* — to pole ma długość czterech bajtów i pomaga wykrywać ewentualne błędy transmisji. Nadawca wysyła wynik cyklicznej kontroli nadmiarowej danych zawartych w pakiecie. Gdy odbiorca otrzymuje pakiet, wówczas przelicza ponownie CRC. Jeśli wyniki są zgodne, dane są wolne od błędów. W przeciwnym razie pakiet zostaje odrzucony.

Standary sterowania dostępu do nosnika

Jesli sieć zawiera dużą liczbę urządzeń, zdolnych do wysyłania danych gdy sieć jest na to gotowa, istnieje wysokie prawdopodobieństwo równoczesnego nadawania przez dwa lub więcej urządzeń. W takich sytuacjach w nosniku transmisji (na przykład w kablu) obecnych jest więcej sygnałów niż jeden, co powoduje uszkodzenie sygnałów i utratę przenoszonych przez nie danych. Zdarzenie takie nosi nazwę kolizji i niszczy łączność.



Potocznie nosnik transmisji jest nazywany *kanałem*.

Aby sieć działała wydajnie, kolizje powinny zostać ograniczone lub wyeliminowane. Sieci stosują określone reguły, zwane standardami sterowania dostępu do kanału, decydujące, kiedy urządzenie ma prawo wysłać pakiet danych.

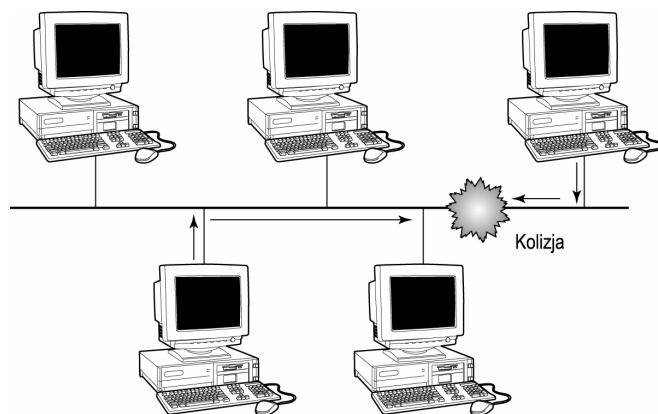
Różne topologie używają różnych standardów sterowania dostępu do nosnika. Na przykład, Ethernet obsługuje *rywalizację*, ARCnet *odpytywanie*, zaś Token Ring *przekazywanie zetonu*.

Ethernet

Ethernet do kontroli nad występowaniem kolizji stosuje metodę rywalizacji o dostęp (ang. *contention*) (przedstawiona na rysunku 4.2). W tej metodzie dostęp do nosnika jest przyznawany na zasadzie „którego pierwszy, ten lepszy”, co oznacza konieczność rywalizacji o dostęp do nosnika ze strony każdego urządzenia sieciowego. Gdy urządzenie chce wysłać dane, wysyła do nosnika transmisji własny sygnał. W tej metodzie istnieje duże prawdopodobieństwo równoczesnego umieszczenia sygnałów w nosniku przez dwa lub więcej urządzeń, co prowadzi do kolizji.

Rysunek 4.2.

Metoda rywalizacji o dostęp



Im więcej urządzeń w sieci, tym większe prawdopodobienstwo kolizji.

Aby ograniczyć liczbę kolizji w sieci Ethernet, opracowano protokoły rywalizacji noszące nazwę *wielodostęp z badaniem stanu kanału* (CSMA — *Carrier Sense Multiple Access*). Do CSMA należą CSMA/CD (*Carrier Sense Multiple Access with Collision Detection* — wielodostęp z badaniem stanu kanału i wykrywaniem kolizji) oraz CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance* — wielodostęp z badaniem stanu kanału i unikaniem kolizji). Protokoły te nakazują urządzeniom nasłuchiwać w nosziku transmisji przed nadawaniem. Jeśli stwierdzony zostanie brak wszelkich sygnałów w nosziku, wówczas w celu transmisji umieszcza tam własny sygnał. W przeciwnym razie CSMA czeka na zwolnienie nosziku. Choć protokoły CSMA zmniejszają prawdopodobienstwo wystąpienia kolizji, nie eliminują go całkowicie. Kolizje wciąż mogą zajść, gdy dwa urządzenia nie wykryją sygnału w nosziku i równoczesnie wysiączą swoje sygnały.

CSMA/CD

Protokół CSMA/CD nakazuje urządzeniom sieciowym nasłuchiwać w nosziku transmisji przed nadaniem sygnału, a ponadto pomaga urządzeniom wykrywać kolizje. Gdy kolizja zostanie wykryta, wszystkie urządzenia w sieci powstrzymują się od nadawania danych przez określony czas. Po jego upływie urządzenie zaczyna rywalizować o noszик. Do przykładów protokołów CSMA/CD należą Ethernet-II oraz IEEE 802.3.

CSMA/CA

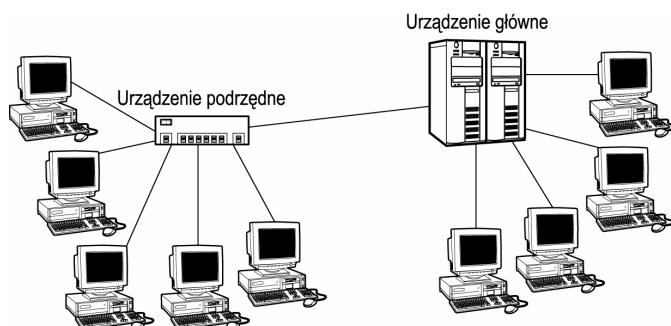
Protokół CSMA/CA w bardzo dużym stopniu redukuje prawdopodobienstwo kolizji, stosując jedną z dwóch technik: albo każde urządzenie posiada ustalony przedział czasowy na transmisję, albo wysyła do nosziku transmisji zadanie dostępu przed wysłaniem danych. W pierwszej metodzie każdemu urządzeniu przydzielany jest określony przedział czasowy, na który musi czekać, aby móc nadawać. Żadne inne urządzenie nie może przeprowadzić transmisji w tym przedziale czasowym. W ten sposób protokół CSMA/CA pomaga uniknąć kolizji. Jednym z przykładów takich protokołów jest LocalTalk firmy Apple.

ARCnet

ARCnet używa topologii okablowania typu „polaczone gwiazdy”, w której sterowanie dostępu urządzenia do nosnika transmisji odbywa się za pomocą metody odpytywania (*polling*). W tej metodzie oprogramowanie protokołu wyznacza jedno urządzenie do roli głównego (*master*), inaczej podstawowego lub kontrolera, zas pozostałe są urządzeniami wtórnymi (*podrzednymi*). Urządzenie główne odpytuje po kolei wszystkie urządzenia podrzędne w ustalony z góry sposób, aby sprawdzić, czy chce wysyłać informacje. Gdy któreś chce wysłać dane, wówczas urządzenie główne wysyła do niego pakiet żądania. W odpowiedzi na ten pakiet urządzenie podrzędne wysyła dane do urządzenia głównego, które z kolei pakiet danych przesyła dalej, do segmentu, w którym znajduje się zamierzony odbiorca. Objętość danych, jaka urządzenie podrzędne może nadać po odpytaniu, jest ograniczona przez protokół. Rysunek 4.3 przedstawia metodę odpytywania używaną przez ARCnet.

Rysunek 4.3.

Metoda dostępu z odpytywaniem



Urządzenie główne nosi inaczej nazwę *administratora dostępu do kanalu*.



W przypadku niektórych aplikacji mogą powstawać opóźnienia spowodowane odpytywaniem w tym czasie innych; sprawę kolejności odpytywania możemy rozwiązać przez nadanie priorytetów. Ponadto metoda odpytywania umożliwia całkowite wykorzystanie przepustowości nosnika transmisji, ponieważ całkowicie eliminuje prawdopodobienstwo kolizji.

Sieci stosujące systemy odpytywania najlepiej nadają się dla urządzeń sieciowych, dla których ważny jest czas reakcji, na przykład dla urządzeń automatyki.



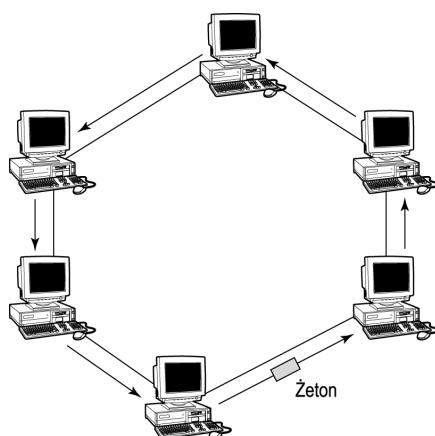
Token Ring

Sieci oparte na topologii Token Ring do sterowania dostępu urządzeniem do nosnika transmisji używają metody *przekazywania zetonu* (inaczej sztafetowej). W tej metodzie specjalna ramka danych, zwana *zetonem* (token), kraczy w całej sieci. Kazde urządzenie „wie”, od którego odebrało zeton, i do którego powinno go przesłać. Urządzenie, które chce nadawać dane, przechwytuje zeton, co daje mu chwilową kontrolę nad nosnikiem transmisji. Urządzenie nadające używa ramki zetonu do opakowania danych, które chce wysłać, a następnie umieszcza zeton w nosniku. Kazde urządzenie odbierające po dro-

dzie ramke sprawdza, czy jest jej zamierzonym adresatem. Jesli tak, wówczas przyjmuje dane i odsyła zeton do nadawcy. W przeciwnym razie zeton zostaje przekazany do nastepnego urzadzenia. Zeton pozostaje przechwycony, dopóki urzadzenie nadajace nie zakonczy transmisiJI, po czym zeton jest „uwalniany”. Rysunek 4.4 przedstawia metode dostepu z przekazywaniem zetona, uzywana w sieciach Token Ring.

Rysunek 4.4.

Metoda dostepu
z przekazywaniem
zetonu



Przekazywanie zetona rozkłada sterowanie dostepem na wszystkie urzadzenia sieciowe. Okres, przez który urzadzenie ma prawo posiadac zeton, jest ograniczony przez odpowiednie protokoly. Kazde urzadzenie po kolei otrzymuje kontrole nad ramka zetona, nadaje dane i zwalnia zeton na uzytek nastepnego urzadzenia.



Metoda dostepu z przekazywaniem zetona jest idealna dla sieci, gdzie sygnaly — na przykład, audio i wideo — musza byc przesypane na czas i z ustalonymi priorytetami.

ATM

Sieci ATM (*Asynchronous Transfer Mode* — tryb przesyłania asynchronicznego) zostały obwieszczone wschodzaca gwiazda technologii sieciowych, poniewaz zapewniaja wyjątkowo szybkie i niezawodne przesyłanie danych na male i duze odleglosci. ATM obsługuje szeroki zakres aplikacji, w tym tradycyjne przesyłanie danych oraz lacznosc audio i wideo w czasie rzeczywistym.

ATM posiada wiele zalet w porównaniu z konwencjonalnymi sieciami lokalnymi, przede wszystkim elastyczosc w korzystaniu z różnorodnych nosników z różnymi predkosciami transmisiJI. ATM moze stosowac w roli nosnika kable koncentryczne, skretke i swiatlowody, z przepustowoscia w zakresie od 25 Mb/s do 2,5 Gb/s. Standardowa predkosc sieci to 155 Mb/s w nosniku swiatlowodowym. Sumaryczna przepustowosc wzrasta wraz z dodawaniem nowych hostów w sieci.

ATM opiera sie na sieci polaczonych ze soba przelacznikow i hostow, które tworza razem polaczona gwiazde. Poniewaz sieci ATM opieraja sie na topologii gwiazdy, korzystaja z metody dostepu z odpytywaniem. Przelacznik ATM pelni funkcje koncentratora w centrum gwiazdy oraz administratora dostepu do kanalu.

Kanaly wirtualne

ATM opiera się na modelu zorientowanym na połaczenia, który używa kanałów wirtualnych do szybkiego transferu danych, synchronicznego lub asynchronicznego. Inaczej mówiąc, ATM wymaga utworzenia wirtualnego połaczenia dwupunktowego, zanim będzie można przesłać dane. Połaczenia te noszą nazwę kanałów wirtualnych (VC — *Virtual Channel*).

Kanal wirtualny przesyła pomiędzy punktami koncowymi *pakiety* (*cell*) o stalej, 53-bajtowej długosci. Każdy kanał wirtualny składa się z identyfikatora kanału wirtualnego (VCI — *Virtual Channel Identifier*) oraz identyfikatora trasy wirtualnej (VPI — *Virtual Path Identifier*), które zawarte są w 5-bajtowym nagłówku pakietu, co umożliwia identyfikację dla właściwej komutacji w przełączniku ATM z portu wejściowego do portu wyjściowego. Ponieważ kanał wirtualny obejmuje wiele łączy, VCI może potencjalnie zmieniać się dla każdego łącza. Gdy kanał wirtualny zostaje utworzony, przełącznik ATM tworzy i utrzymuje wpis w tablicy, przypisujący nadchodzące VCI na porcie wejściowym do wychodzących VCI na porcie wyjściowym. Algorytm ATM z uwagi na prostotę może być łatwo implementowany sprzętowo. Oprogramowanie wymagane jest jedynie do zarządzania połaczeniami i utrzymywania tablicy komutacji.

Aby pomyslnie komunikować się z innymi komputerami, każde urządzenie musi być właściwie identyfikowane w unikalowy sposób. Nie stanowi to problemu w sieci LAN, ponieważ w niej liczba urządzeń nie jest wyjątkowo wysoka i można do nich identyfikacji użyć adresów fizycznych. Adres fizyczny, czyli adres sterowania dostępnego do nosnika (MAC — *Media Access Control*), jest zakodowany w sposób trwałym w karcie interfejsu sieciowego urządzenia. Jednakże stosowanie adresów MAC do identyfikacji urządzeń sieciowych nie jest możliwe w dużych, mieszanych i globalnych systemach sieciowych, takich jak Internet. Aby możliwa była w nich łączność, każdy komputer z dowolnej sieci musi być zdolny do komunikacji z wszystkimi innymi komputerami, niezależnie od ich nazw.

Aby umożliwić komunikację globalną, opracowano system nazewniczy, w którym każdemu komputerowi przydzielony jest unikalny, 32-bitowy *adres internetowy*, czyli *adres IP*. Przykładowy adres IP może wyglądać tak:

10000000 000010000 00000111 00011111

Chociaż powyższy ciąg jest unikalny, dla użytkowników jest trudny do zapamiętania. Wobec tego adresy IP są zapisywane w postaci czterech dziesiętnych liczb naturalnych, oddzielonych od siebie kropkami. Każda liczba reprezentuje bajt (oktet). Taka forma zapisu nosi nazwę *notacji dziesiętnej rozzielonej kropkami* (*dotted decimal notation*). Jeśli przekształcimy powyższy adres na te notacje, otrzymamy coś takiego:

128.8.7.31



Dodatkowe informacje o adresach IP znajdują się w rozdziale 5.

Adres IP podzielony jest na dwie części: adres sieci i adres hosta. Adres sieci identyfikuje sieć, do której host jest połączony. Adres hosta, jak wskazuje nazwa, reprezentuje adres urządzenia (hosta) w sieci. Jak za pomocą adresu IP nadawca znajduje właściwego hosta docelowego, zwłaszcza położonego w innej sieci? Inaczej mówiąc, w jaki sposób komputer-nadawca kojarzy adres IP hosta docelowego z jego adresem fizycznym?

Odwzorowanie adresów fizycznych na adresy IP

Gdy komputer komunikuje sie z innym, polozyonym w innej sieci, odwzorowanie adresów musi odbyc sie dwukrotnie:

1. Nadawca musi odwzorowac adres posredniczacych ruterów, które znajdują sie po drodze do sieci docelowej, tak by dane mogły dotrzeć do pozadanej sieci.
2. Nadawca musi również odwzorowac adres IP komputera docelowego na jego adres fizyczny, aby dane dotarły do właściwego hosta.

Odwzorowanie adresów wysokiego poziomu (adresów IP) na adresy sprzętowe (niskiego poziomu) nosi nazwę *rozwiązywania adresu* (*address resolution*). Adresy mogą być rozwiązywane w różny sposób. Na przykład, każdy komputer w sieci może utrzymywać tablice, która przypisuje adresy wysokiego poziomu do odpowiadających im adresów fizycznych; adresy fizyczne mogą też być zakodowane w adresach wysokiego poziomu. Jednakże obie te metody zużywają duzo zasobów i wymagają pracy administratora.

Aby poradzić sobie z problemem rozwiązywania nazw bez dodatkowego obciążenia i recznej obsługi w każdym komputerze, opracowane zostały protokoly ARP i RARP. Obecnie należą one do najczęściej stosowanych technik rozwiązywania adresów. Razem z nową technologią ATM pojawił się nowy, lecz szybko zdobywający popularność, protokół ATMARP (*Asynchronous Transfer Mode Address Resolution Protocol*).

ARP i RARP

Gdy karta sieciowa ulega awarii i zostaje wymieniona, adres fizyczny komputera również ulega zmianie. Powoduje to problemy, gdy każdy komputer utrzymuje tablice odwzorowujące adresy IP na odpowiadające im adresy fizyczne. W tym przypadku trzeba recznie zmodyfikować tablice we wszystkich komputerach. ARP (*Address Resolution Protocol* — protokół rozwiązywania nazw) umożliwia zastępowanie urządzeń w istniejącej sieci i dodawanie nowych bez konieczności recznych zmian w tablicach utrzymywanych przez każdy komputer. ARP jest protokołem niskiego poziomu, który dynamicznie wiąże adresy IP urządzeń sieciowych z ich adresami fizycznymi, korzystając z funkcjonalności rozgłoszeń dostępnej w sieci.



Gdy protokół współpracuje bezpośrednio ze sprzętrem, wówczas jest *protokołem niskiego poziomu* (low-level).

W ARP komputer, który chce znaleźć fizyczny adres innego komputera, rozglasza specjalny pakiet. Pakiet ten zawiera zadanie skierowane do hosta o danym adresie IP, aby odpowiedział podając swój adres fizyczny. Ponieważ pakiet jest komunikatem rozgłoszeniowym, odbierają go wszystkie urządzenia. Jednakże tylko host, którego zadanie dotyczy, rozpoznaje w nim swój adres IP i odpowiada podając własne adresy IP i fizyczny.

Rozgłoszenia zużywają spora części przepustowości nosnika, dlatego komputery używające ARP utrzymują pamięć podręczną rozwiązań adresów, która zawiera ostatnio odebrane powiązania IP z adresami fizycznymi. Gdy komputer chce wysłać dane, w pierwszej kolejności usiłuje pobrać potrzebny adres z pamięci podręcznej. W razie

powodzenia nie musi rozglaszac zadania rozwiazania adresu, co zmniejsza znaczaco liczbe rozgloszen, a co za tym idzie, koszt polaczenia i ruch sieciowy.

Funkcjonalnie mozna podzielic ARP na dwie czesci. Jedna odwzorowuje adresy IP na odpowiadajace im adresy fizyczne, zas druga ma za zadanie odpowiadac na zadania rozwiazania adresów odbierane z innych komputerów. Odwzorowanie adresów IP na adresy fizyczne i odpowiadanie na zadania rozwiazania adresów moze wydawac sie proste, lecz ponizsze sytuacje moga powodowac problemy:

- ◆ *Komputer docelowy jest wylaczony lub zbyt zajety, by przyjac zadanie ARP.*
W tym przypadku nadawca moze nie otrzymac odpowiedzi, lub otrzymac z opoznieniem. To moze powodowac zawieszenie niektórych aplikacji w komputerze zrodlowym do czasu, gdy zadanie ARP zostanie przetworzone.
- ◆ *Nadawca niedawno otrzymał odwzorowanie adresu z innego komputera, w którym karta sieciowa w tym czasie została wymieniona.* Chociaz adres fizyczny również ulegl zmianie, wpis w pamieci podrecznej nadawcy pozostal niezmieniony. Kolejnym razem, gdy nadawca wysle dane do komputera docelowego, transmisja pomiedzy nimi bedzie niemozliwa.

ARP utrzymuje pamiec podreczna odwzorowan adresów IP na MAC do wykorzystania w przyszlosci, co rowniez pozwala znaczaco zmniejszyc liczbe rozgloszen w sieci. Pamiec podreczna ARP moze zawierac dwa typy wpisow:

- ◆ *Wpisy dynamiczne* — dodawane i usuwane automatycznie z pamieci podrecznej ARP w ustalonych odstepach czasu.
- ◆ *Wpisy statyczne* — pozostaja w pamieci ARP az do restartu komputera.

Wpisy dynamiczne moga pozostac w pamieci podrecznej ARP przez najwyzej dziesiec minut. Nowe wpisy do pamieci podrecznej otrzymuja znaczniki czasowe. Jesli wpis nie zostanie ponownie uzyty w przeciagu dwóch minut od dodania do pamieci podrecznej, wówczas jest z niej usuwany. Jesli zostanie uzyty, otrzymuje kolejne dwie minuty czasu waznosci. Jesli wpis uzywany jest regularnie, za kazdym razem otrzymuje dodatkowe dwie minuty az do maksymalnej wartosci dziesieciu minut, po których zostaje usuniety.

Pamiec podreczna ARP posiada określona wielosc, która nie moze przekroczyć ustalonego limitu. Gdyby pozwolono jej rozrastac sie bez ograniczen, pamiec podreczna mogłaby zostac zapelniona wpisami niekompletnymi lub przestarzalymi. Aby temu zapobiec, pamiec podreczna ARP jest okresowo oczyszczana z wszystkich wpisow. W ten sposob nie uzywane wpisy sa usuwane, co zwalnia miejsce dla wpisow nowszych i bardziej przydatnych, a ponadto zmniejsza sie prawdopodobienstwo prób kontaktowania sie z komputerami, które sa aktualnie niedostepne.

RARP

Adres IP komputera jest zazwyczaj zapisany na dysku twardym i pobierany podczas uruchomienia komputera. Jednakze ustalenie adresu IP staje sie krytycznym problemem dla komputerow, które nie posiadaja dysku twardego, poniewaz potrzebny jest im adres IP, aby pomyslnie pobrac plik inicjujacy.

Komputery bez dysków twardych do pobrania swojego adresu IP z serwera uzywaja protokolu RARP (*Reverse Address Resolution Protocol* — protokół wstecznego rozwiazywania adresów). RARP nalezy do pakietu protokolów TCP/IP. W przeciwnieństwie do ARP, RARP pozwala również urzadzeniom wysylac zapytania o adresy IP innych komputerów niz docelowy, oraz o wiele fizycznych typów sieci.



Podobnie jak zadania ARP, zadania RARP moga w czasie transmisiji zostac utracone lub ulec uszkodzeniu.

Format ramki protokolów ARP i RARP

Pakiety zadan ARP i RARP maja wspólny format ramki (komunikatu), przedstawiony na rysunku 4.5.

Rysunek 4.5.

Format ramki
w protokolach
ARP i RARP

0	8	16	24	31
Typ sprzetu		Typ protokolu		
HLEN		Dzialanie		
Adres sprzetowy nadawcy (oktety 0-3)				
Adres sprzetowy nadawcy (oktety 4-5)		IP nadawcy (oktety 0-1)		
IP nadawcy (oktety 2-3)		Adres sprzetowy nadawcy (oktety 0-1)		
Adres sprzetowy odbiorcy (oktety 2-5)				
IP odbiorcy (oktety 0-3)				

Pola w ramce ARP (RARP) reprezentuja:

- ◆ *Typ sprzetu* — okresla typ interfejsu sprzetowego, uzywanego przez nadawce. Interfejs Ethernet jest reprezentowany przez wartosc 1.
- ◆ *Typ protokolu* — podaje adres protokolu wysokiego poziomu, który wyslal zadanie. Adresy IP reprezentuje wartosc 0800h.
- ◆ *HLEN* — podaje dlugosc adresu sprzetowego, zawartego w polu *Typ sprzetu*.
- ◆ *PLEN* — podaje dlugosc adresu protokolu wysokiego poziomu, zawartego w polu *Typ protokolu*
- ◆ *Dzialanie* — okresla, czy ramka stanowi zadanie (odpowiedz) protokolu ARP, czy RARP. Jesli wartosc zawarta w tym polu wynosi 1, wówczas jest ona zadaniem ARP. Wartosc 2 oznacza odpowiedz ARP. Wartosc 3 to zadanie RARP, a wartosc 4 — odpowiedz RARP.
- ◆ *Adres sprzetowy nadawcy (Sender HA)* — podaje adres sprzetowy urzadzenia nadajacego.
- ◆ *IP nadawcy* — podaje adres IP urzadzenia nadajacego.
- ◆ *Adres sprzetowy docelowy* — podaje adres sprzetowy urzadzenia docelowego, jesli jest znany przez nadawce.
- ◆ *IP docelowy* — podaje adres IP urzadzenia docelowego.

Po otrzymaniu ramki urządzenie docelowe wypełnia w razie potrzeby brakujący adres, zmienia wartość w polu Działanie na odpowiedź i zamienia miejscami pola nadawcy i odbiorcy. W rezultacie ramka odpowiedzi zawiera adresy IP i fizyczny zarówno oryginalnego nadawcy, jak i urządzenia docelowego, którego adres miał zostać rozwiązyany.

ATMARP

W sieci ATM host, który chce wysłać dane do innego komputera, musi podać adres sprzętowy hosta docelowego. Odwzorowanie adresu hosta na odpowiadający adres sprzętowy ATM w sieciach ATM stanowi problem, ponieważ sieci te, w przeciwieństwie do technologii rozgłoszeniowych, takich jak Ethernet czy Token Ring, nie obsługują rozgłoszeń sprzętowych. Aby rozwiązać adres IP na odpowiadający mu adres sprzętowy, dany host ATM musi skontaktować się z serwerem, który zawiera odwzorowania. Ramki protokołu ATMARP (*Asynchronous Transfer Mode Address Resolution Protocol* — protokół rozwiązywania adresów ATM) wykorzystują te komunikacje pomiędzy hostem i serwerem.

ATMARP przypomina pod wieloma względami protokół ARP używany w sieciach Ethernet i Token Ring. Gdy host ATM chce poznać adres fizyczny innego komputera na podstawie jego adresu IP, wówczas generuje zadanie, zawierające ten adres IP, które zostaje następnie wysłane do serwera ATMARP. Jeśli serwer posiada zadany adres sprzętowy w swojej pamięci podręcznej, wówczas odsyła odpowiedź ATMARP. W przeciwnym wypadku zwraca negatywną odpowiedź ATMARP.

Format pakietu ATMARP różni się nieco od tradycyjnego formatu ramki ATM. Zawiera on dodatkowe pola długości adresu, aby pomieścić dodatkowy format adresów, który wprowadza w ATM dwupoziomowa hierarchię adresów.



ATM obsługuje większą liczbę formatów adresów, ponieważ przedsiębiorstwa telefoniczne oferujące usługi i sieci ATM używają 8-bajtowego formatu adresu. Z drugiej strony, według ATM Forum, każdy komputer przyłączony do sieci ATM może otrzymać 20-bajtowy adres o nazwie NSAP (*Network Service Access Point* — punkt dostępu do usługi sieciowej). W ten sposób tworzy się dwupoziomowa hierarchia adresów, w której adresy 8-bajtowe używane są do dostępu zdalnego, zaś NSAP — do dostępu lokalnego.

Rysunek 4.6 przedstawia format pakietu ATMARP. Pola typu sprzętu, typu protokołu i działania są takie same, jak w ramce ARP.

Pola w pakiecie ATMARP to:

- ♦ *Typ sprzętu* — określa typ interfejsu sprzętowego, używanego przez nadawcę. Interfejs ATM jest reprezentowany przez wartość 0x0013.
- ♦ *Typ protokołu* — podaje adres protokołu wysokiego poziomu, który wysłał zadanie. Adresy IP są reprezentowane przez wartość 0x0800.
- ♦ *HLEN nadawcy* — podaje długość adresu ATM nadawcy.
- ♦ *HLEN 2 nadawcy* — podaje długość adresu podadresu ATM nadawcy.

Rysunek 4.6.

*Format pakietu
ATMARP*

0	8	16	24	31
Typ sprzętu		Typ protokołu		
HLEN nadawcy		Działanie		
PLEN nadawcy	HLEN odbiorcy	HLEN2 odbiorcy	PLEN odbiorcy	
Adres ATM nadawcy (oktety 0-3)				
Adres ATM nadawcy (oktety 4-7)				
Adres ATM nadawcy (oktety 8-11)				
Adres ATM nadawcy (oktety 12-15)				
Adres ATM nadawcy (oktety 16-19)				
Adres protokołu nadawcy				
Adres ATM odbiorcy (oktety 0-3)				
Adres ATM odbiorcy (oktety 4-7)				
Adres ATM odbiorcy (oktety 8-11)				
Adres ATM odbiorcy (oktety 12-15)				
Adres ATM odbiorcy (oktety 16-19)				
Adres protokołu odbiorcy				

- ◆ *Działanie* — określa, czy ramka stanowi zadanie, czy odpowiedź. Jeśli wartość zawarta w tym polu wynosi 1, ówczas jest ona zadaniem ATMARP. Wartość 2 oznacza odpowiedź ATMARP. Wartość 8 oznacza odwrotne zadanie ATMARP, 9 oznacza odwrotną odpowiedź ATMARP, a wartość 10 negatywne potwierdzenie ATMARP.
- ◆ *PLEN nadawcy* — podaje długość adresu protokołu nadawcy.
- ◆ *HLEN docelowy* — podaje długość adresu ATM komputera docelowego.
- ◆ *HLEN 2 docelowy* — podaje długość adresu ATM komputera docelowego.
- ◆ *PLEN docelowy* — podaje długość adresu protokołu komputera docelowego.
- ◆ *Adres ATM nadawcy* — podaje adres ATM urządzenia nadającego. To pole może mieć długość 20 bajtów.
- ◆ *Adres protokołu nadawcy* — podaje adres protokołu, który zainicjował zadanie u nadawcy.
- ◆ *Docelowy adres ATM* — podaje adres ATM komputera docelowego.
- ◆ *Docelowy adres protokołu* — podaje adres protokołu w hostie docelowym, który powinien przetworzyć zadanie.

Rozdział 5.

Warstwa internetowa

W tym rozdziale:

- ◆ Adresowanie IP
- ◆ Podstawy trasowania
- ◆ Wykorzystanie protokolu ICMP
- ◆ Wykorzystanie protokolu IGMP

Duża część działań, które zachodzą w warstwie internetowej TCP/IP, jest ukryta przed użytkownikiem, podobnie jak silnik samochodu jest ukryty pod maską. Istotnie, możemy uznac warstwę internetową za silnik TCP/IP. Bieżący rozdział, który koncentruje się na adresowaniu i dostarczaniu pakietów, pozwoli nam „zajrzeć pod maskę”. Po dotarciu do końca rozdziału, Czytelnik lepiej zrozumie, jak pakiety są adresowane i trasowane. Przedstawimy tutaj również podstawowa wiedzę o wielu zagadnieniach omówionych w dalszych rozdziałach, jak np. planowanie schematu adresowania (patrz rozdział 18.) i podsieci (patrz rozdział 19.).

Przeznaczenie warstwy internetowej

Gdy dane muszą zostać przesłane przez sieć ruterów IP, które stosują różne rozmiary ramek, wówczas jest to zadanie dla warstwy internetowej. Jeśli komputer próbuje połączyć się przez sieć z nieistniejącym hostem, to za komunikat, powiadamiający, że po drugiej stronie „nikogo nie ma w domu” odpowiada protokół ICMP warstwy internetowej. Gdy używamy narzędzi, które pozwalają urządzając przez Internet wirtualne zebrania z wybraną grupą uczestników, warstwa internetowa służy do przesyłania treści spotkania jedynie do tych odbiorców. Wszystkie te funkcje udostępnia stosunkowo niewielka grupa protokołów działających w warstwie internetowej. Mówiąc krótko, warstwa internetowa TCP/IP odpowiada za adresowanie pakietów i przesyłanie datagramów przez sieć ustaloną trasę.

Uslugi warstwy internetowej dostarczane są przez trzy współpracujące ze sobą protokoły:

- ◆ IP (*Internet Protocol*) udostępnia usługi pakowania i adresowania. IP identyfikuje hosty lokalne i zdalne. Gdy trasa do sieci docelowej wymaga innych rozmiarów pakietu, IP dzieli pakiet na fragmenty, co pozwala na ich transmisję bez błędów, a następnie składa razem fragmenty w pakiet w celu docelowym. IP odrzuca

tez pakiety przeterminowane oraz przekazuje wyznaczone pakiety do protokolów w wyzszych warstwach. Adresowanie IP jest opisane w RFC 791.

Dokumenty RFC mozna znalezc wedlug numerow pod adresem www.ietf.org.



- ◆ ICMP (*Internet Control Messaging Protocol*) — protokół komunikacyjny sterowania siecia Internet) sluzy do raportowania i diagnozowania problemów wystepujacych podczas transmisji. Czytelnik zapewne zetknal sie z niektórymi komunikatami ICMP, np. „Host docelowy jest niedostepny”. Protokół ICMP opisany jest w RFC 792.
- ◆ IGMP (*Internet Group Management Protocol*) — protokół zarzadzania grupami internetowymi odpowiada za zarzadzanie przesyaniem grupowym i dostarczaniem selektywnym bez rozgloszen. Protokół IGMP jest opisany w RFC 1112 i 2236.



Pojecia *adres Ethernet*, *adres MAC* i *adres fizyczny* moga byc uzywane zamiennie, podobnie jak pojecia *podsieci* i *segment*.

Ustalenie, czy adres docelowy jest lokalny czy odległy

Kazdy pakiet w sieci jest adresowany w warstwie internetowej za pomoca dwóch adresów IP: zródłowego i docelowego dla danego pakietu. Pola te sa widoczne i moga byc uzywane przez inne hosty, które przetwarzaja pakiety IP. Protokół IP ustala, czy miejsce przeznaczenia kazdego pakietu jest lokalne czy zdalne, porównujac pole adresu docelowego w pakiecie z własnym adresem IP. Różnica pomiedzy sieciowym ruchem lokalnym i zdalnym jest dosc duza, poniewaz host lokalny potrafi sam dostarczyc pakiety lokalnie, zas w ruchu zdalnym dostarczenie pakietu do odleglych sieci dostarczenie pakietu wymaga trasowania przez ruter.

Protokół ARP

Protokół rozwiazywania adresów (ARP — *Address Resolution Protocol*) jest przedstawiany w warstwie internetowej. W rzeczywistosci ARP dziala po obu stronach granicy miedzy warstwa lacza danych i warstwa internetowa. Jak wspomniano w poprzednim rozdziale, zadaniem ARP jest rozwiazywanie adresów IP na adresy fizyczne (MAC), aby pakiety mogly zostac dostarczone do odpowiedniego adaptera sieciowego w lokalnym segmencie. Protokół ARP do znajdowania hostów docelowych uzywa rozgloszen, wiec jego funkcjonalnosć jest ograniczona do lokalnego segmentu — poniewaz wiekszosc ruterów nie przepuszcza rozgloszen. Protokół ARP, opisany w RFC 826, zazwyczaj stosowany jest w nastepujacy sposob:

- ◆ Host uzywa ARP do wymiany adresów IP i MAC z lokalnym hostem docelowym, aby mōc przeslac do niego datagramy IP.
- ◆ Host uzywa ARP do wymiany adresów IP i MAC ze swoja brama domyslna (*default gateway*), aby mōc dostarczyc datagramy IP przez rutery do sieci docelowej.

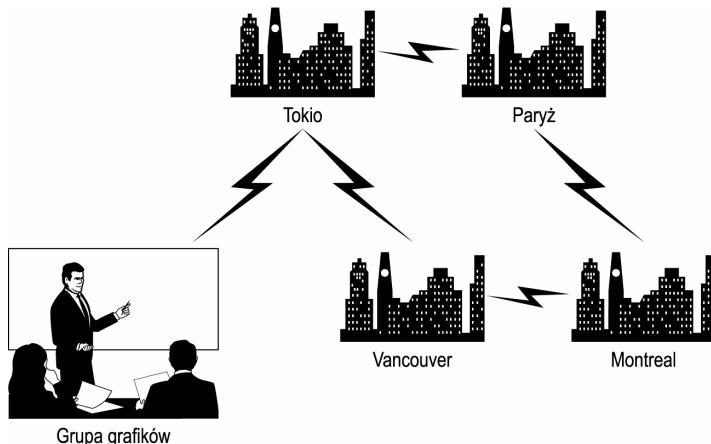
Wprowadzenie do trasowania

Czy znacie organizacje prowadzące interesy w wielu lokalizacjach? Jeśli tak, wówczas prawdopodobnie połączone są one laczami sieci rozległych (WAN), na przykład ISDN lub T1. Sieci WAN są drogie w eksploatacji i mają ograniczoną przepustowość. Czy dane tejże organizacji są szybko przesyłane z jednej siedziby firmy do drugiej za pomocą poczty elektronicznej, transferu plików lub wewnętrznej sieci? Czy określone grupy w organizacji regularnie pogarszają sprawność sieci, przesyłając duże zasoby danych? Z takimi problemami pozwala uporać się trasowanie, selektywnie zezwalając na przesyłanie danych pomiędzy sieciami.

Srodowisko z sieciami trasowanymi pozwala na tworzenie nadmiarowych łącz komunikacyjnych pomiędzy lokalizacjami, na przykład między Montrealem a Tokio, jak na rysunku 5.1. Ruch sieciowy może być kierowany przez najszybsze dostępne łącza, lecz w przypadku, gdy będzie ono niedostępne, dane można przesłać wolniejszym łączem. Co więcej, tylko kierowany ruch będzie obciążać sieci rozległe. Ruting może również zapobiec spowalnianiu przez grupę grafików działania sieci, jeśli stworzymy dla tej grupy dedykowane łącza.

Rysunek 5.1.

Sieć trasowana



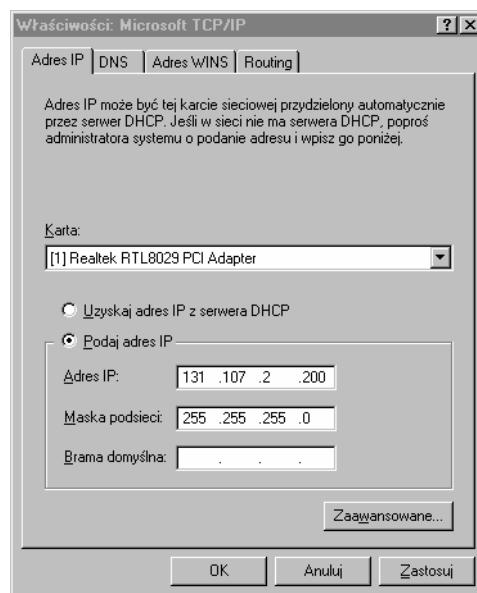
Adresy IP

Dla niektórych użytkowników adres protokołu internetowego (*Internet Protocol*) jest jedynie dwunastocyfrowa liczba, podzielona na cztery części za pomocą kropki. Adresy IP to jednak coś wiele niż zwykłe liczby — służą bowiem do unikatowego opisywania wszystkich urządzeń w sieci. Komputery, routery, drukarki sieciowe, a nawet witryny WWW posiadają własne unikatowe adresy IP.

Aby urządzenie poprawnie funkcjonowało w międzysieci IP, musi zostać odpowiednio skonfigurowane. Hosty IP można konfigurować automatycznie lub ręcznie. Kazde urządzenie potrzebuje danych adresu IP, maski podsieci i bramy domyślnej. Rysunek 5.2 przedstawia okno konfiguracji IP dla komputera Windows NT. Inne systemy operacyjne mogą do ręcznej konfiguracji IP udostępniać inne metody, lecz wynik pozostaje taki sam.

Rysunek 5.2.

Konfiguracja IP
w Windows NT



Konfiguracja reczna wymaga poprawnego wprowadzenia parametrów adresu IP, maski podsieci i bramy domyślnej. W sieciach IP to zadanie stanowi najczęstsze źródło problemów.

Automatyczna konfiguracja parametrów IP wymaga obecności działającego serwera DHCP (*Dynamic Host Configuration Protocol* — protokół dynamicznej konfiguracji hostów) w sieci. Gdy usługa DHCP jest dostępna, wystarczy zaznaczyć pole wyboru *Uzyskaj adres IP z serwera DHCP*, aby dokonać automatycznej konfiguracji IP.



Protokół DHCP opisany jest bardziej szczegółowo w rozdziale 9.

Adresy IP zostały zoptymalizowane na użytek komputerów. Gdyby Czytelnik rozumiał kod dwójkowy równie dobrze jak język ojczysty, mógłby czytać adresy IP jak numery telefonów, a tablice tras równie łatwo, jak mapy samochodowa. Jest to jednak mało prawdopodobne. Aby zrozumieć, jak funkcjonuje protokół IP, trzeba wiedzieć, jak przekształcać adresy IP z formy dwójkowej na dziesiętną i na odwrót. Zanim więc zaglebimy się w szczegóły IP, proponujemy krótka lekcja algebrai dwójkowej. Czytelnicy czujacy się swobodnie w kodzie dwójkowym mogą pominiac ten punkt i przejść do punktu „*Identyfikatory sieci i hostów*”.

Notacja dwójkowa i dziesiętna

Weźmy pod uwagę bardzo pospolity adres IP: 192.68.0.1. Adres IP składa się z czterech liczb rozdzielonych kropkami. Ten typ systemu notacji nosi nazwę *notacji dziesiętnej rozdzielonej kropkami* (*dotted decimal notation*). Kazda z liczb nosi nazwę oktetu, ponieważ w rzeczywistości reprezentuje 8-bitową liczbę dwójkową. Oznacza to, iż w adresie IP największa wartość, jaka może przyjąć każda z czterech liczb dziesiętnych, to 255, a nie 999.

Konwersja z systemu dwójkowego na dziesiętny

Tabela 5.1 przedstawia tablice konwersji z systemu dwójkowego na dziesiętny. *Wartości dwójkowe bitów* (WDB) w okciecie zostały wypisane w górnym wierszu tablicy. Wystarczy porównać oktet z ta tablica, bit po bicie, aby przekształcić liczbę dwójkową na czytelna wartość dziesiętną bez konieczności zapamiętywania wartości bitów. Pierwszy wiersz przedstawia wartość dziesiętną jedynki w systemie dwójkowym, gdy pojawi się na danej pozycji oktetu. Aby dokonać konwersji na system dziesiętny, wystarczy dodać do siebie wszystkie WDB, pod którymi znajduje się jedynka — jeśli więc wartość na danej pozycji wynosi zero, WDB nie dodajemy do wyniku.

Tabela 5.1. Tablica konwersji z systemu dwójkowego na dziesiętny

WDB	128	64	32	16	8	4	2	1	Wynik
Wiersz A	1	1	1	1	1	1	1	1	255
Wiersz B	1	1	0	0	0	1	0	0	196
Wiersz C									

W wierszu A wszystkie bity mają wartość 1. Aby przekształcić liczbę dwójkową z wiersza A na dziesiętną, dodajemy WDB dla wszystkich jedynek i ignorujemy zera. Jak widać, maksymalna wartość 8-bitowej liczby dwójkowej wynosi 255. Inaczej mówiąc, $128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$.

W wierszu B tylko trzy bity mają wartość 1. Aby dokonać konwersji, ponownie wystarczy dodać do siebie WDB nad bitami o wartości 1. W tym przypadku 128, 64 i 4 dają w sumie 196.

Ten sam proces można zastosować do każdego oktetu adresu IP, aby przekształcić go z notacji dwójkowej na dziesiętną rozzieloną kropkami. 11000000 10101000 00000000 00000001 to adres IP 192.168.0.1 wyrażony w postaci dwójkowej. Jeśli porównamy każdy z oktetów z wartościami WDB, wykorzystując wiersz C, otrzymamy w wyniku 192.168.0.1

Konwersja z systemu dziesiętnego na dwójkowy

Jak przekształcić adres IP z postaci dziesiętnej rozzielonej kropkami na notację dwójkową? Oktet po okciecie. Wystarczy odejmować najwyższą możliwą wartość WDB od wartości dziesiętej i notować, które WDB były używane, aż do wyczerpania reszty. Wiersz C z tablicy 5.1 może posłużyć do konwersji wartości dziesiętej 168 na dwójkową według poniższej procedury:

1. Zaczniemy od odejęcia 128 od 168, ponieważ 128 jest najwyższą wartością WDB, która zmieści się w 168.
2. $168 - 128 = 40$ i zanotuj dwójkowa wartość 1 w kolumnie WDB = 128.
3. $64 > 40$. Wobec tego można pominać WDB = 54 i przejść do 32.
4. $40 - 32 = 8$, więc zanotuj „1” w kolumnie WDB = 32.
5. Można pominać 16, ponieważ nie da się odjąć 16 od 8.

6. $8 - 8 = 0$, co daje „1” w kolumnie WDB = 8.
7. Tam, gdzie WDB byly pominiete, wypelnij pola zerami, co da w wyniku 10101000.
8. Jeden oktet zostal przekształcony. Ten sam scenariusz zastosuj do pozostalych oktetów.



Do tego samego celu mozna wykorzystac kalkulator systemu Windows w widoku naukowym. Klawisze F6 i F8 sluzą do konwersji notacji dwójkowej na dziesietna i odwrotnie. Aby uzupełnić liczbę do osmio bitów, moze byc potrzebne uzupełnienie liczby zerami od lewej.

Kazda liczba dziesietna w adresie IP reprezentuje liczbę dwójkową. Zrozumienie notacji dwójkowej adresów IP pomoże nam objasnic pojecia omówione w dalszych rozdziałach, takie jak ruting i podsieci.

Identyfikatory sieci i hostów

Adres IP składa się z dwóch identyfikatorów: ID hosta oraz ID sieci. Odróżnienie ich od siebie jest kluczem do zrozumienia adresowania IP.

ID sieci

Kazda sieć IP musi posiadać unikatowy identyfikator, wspólny dla wszystkich hostów w danym segmencie. Ponieważ sieci widoczne w Internecie identyfikowane są przez swoje ID sieci, identyfikatory te muszą być unikalne na skali światowej. Dwie sieci nie mogą posiadać tego samego ID. Identyfikator sieci możemy zarezerwować, kontaktując się z dostawcą usług internetowych (ISP) lub organizacją IANA (*Internet Assigned Numbers Authority*) — www.iana.org. Standardowe ID sieci mają długość 8, 16 lub 24 bitów.



Rozdział 18. zawiera wiele informacji o tym, jak uzyskać adres IP oraz połączenia.

ID hosta

ID hosta służy do opisania każdego urządzenia w sieci i musi być unikatowy w obrębie sieci. Dwa hosty w jednej sieci nie mogą posiadać tego samego identyfikatora hosta. W każdej sieci zarezerwowane są dwa specjalne adresy; jednym z nich jest adres rozgłoszeniowy podsieci, którego ID hosta w zapisie dwójkowym składa się z samych jedynek. Adres ten może służyć do równoczesnego wysłania informacji do wszystkich hostów w sieci. Drugim jest adres lokalny, który nie podlega trasowaniu, lecz identyfikuje go ID hosta składający się z samych zer, również w zapisie dwójkowym. Wobec tego ID hosta mogą stanowić dowolne kombinacje wartości dwójkowych, z wyjątkiem samych jedynek i samych zer. Długość standardowego ID hosta wynosi od 8 do 24 bitów.



Gdyby dwa hosty posiadały taki sam ID w jednej podsieci, jeden z nich nie byłby zdolny do komunikacji w sieci, zas oba prawdopodobnie odbieraliby powtarzające się komunikaty o błędzie.

Porady dotyczące ID sieci i hostów

Poniższe wytyczne są powszechnie stosowane w stosunku do ID sieci i hostów:

- ◆ ID sieci musi być unikatowy na skali ogólnoswiatowej i zarejestrowany, jeśli sieć ma być połączona z Internetem.
- ◆ ID sieci nie może wynosić 127 — jest to wartość zarezerwowana dla lokalnego adresu zwrotnego (*loopback*).
- ◆ Ani ID sieci, ani ID hosta nie może zawierać w zapisie dwójkowym samych jedynek. Taka wartość jest zarezerwowana dla rozgłoszeń.
- ◆ Ani ID hosta, ani ID sieci nie może zawierać samych zer w zapisie dwójkowym. Ten specjalny adres jest zarezerwowany dla pakietów „tylko lokalnych”, które nie są przekazywane przez routery.
- ◆ ID hostów nie mogą się duplikować w jednej sieci.



Organizacja IANA zarezerwowała na potrzeby sieci prywatnych poniższe trzy bloki z przestrzeni adresów IP:

IP 10.0.0.0 — 10.255.255.255 z maską podsieci 255.0.0.0
 IP 172.16.0.0 — 172.31.255.255 z maską podsieci 255.255.0.0
 IP 192.168.0.0 — 192.168.255.255 z maską podsieci 255.255.255.0

Więcej informacji o prywatnych adresach IP można znaleźć w RFC 1918.

Klasy adresów IPv4

Adresowanie IP pozwala w TCP/IP tworzyć sieci od bardzo małych aż do ogromnych, wielomilionowych organizacji, z wykorzystaniem pojedynczego schematu adresowania. W chwili obecnej powszechnie stosowany jest Internet Protocol w wersji 4 (IPv4). Organizacje, które chcą mieć połączenie z Internetem, zasadniczo rezerwują na swój użytek zakresy adresów, kontaktując się ze swoim dostawcą usług internetowych lub organizacją wydającą adresy IP w danym kraju.

IPv4 stosuje pięć klas adresów, oznaczonych literami od A do E. Adresy klas A, B i C są dostępne do rezerwacji. Adresy klasy D są zarezerwowane dla specjalnych aplikacji, które używają adresowania grupowego (pojęcie to zostanie omówione w dalszej części rozdziału), zaś adresy klasy E są eksperymentalne. Na razie skoncentrujemy się na klasach od A do C.

Tabela 5.2 pokazuje, iż klasy IP można rozróżnić na podstawie wartości dziesiątej pierwszego oktetu adresu IP. Klasy IP są podzielone tak, by rozróżnić sieci małe, średnie i duże. Bardzo duże organizacje, posiadające miliony hostów, mogą potrzebować adresów klasy A, jednakże takich adresów jest dostępnych bardzo niewiele. Organizacje średnich rozmiarów mogą używać adresów klasy B, które wciąż mogą pomieścić ponad 65 000 urządzeń sieciowych; adresów tych jest dostępnych znacznie więcej niż w klasie A. Najczęściej będziemy spotykać się z adresami klasy C. Chociaż nie mogą one adresować więcej niż 254 hostów w pojedynczej sieci, dostępnych jest ponad 2 miliony zakresów klasy C.

Tabela 5.2. Zakresy i pojemnosci adresów IP

Klasa adresu	Zakres pierwszego oktetu	Liczba sieci	Liczba hostów w sieci
A	1 – 126	126	16 777 214
B	128 – 191	16 384	65 534
C	192 – 223	2 097 152	254
D	224 – 239	Nie dotyczy	Nie dotyczy
E	240 – 254	Nie dotyczy	Nie dotyczy



Adresy klas D i E nie obsługują adresowania hostów w typowym znaczeniu tego słowa. Klasa D służy do adresowania grupowego, zas adresy klasy E są zarezerwowane do celów eksperymentalnych.

Tabela 5.3 przedstawia dwójkowa postać adresów klas A, B i C, w której S oznacza bity identyfikatora sieci, zas H bity ID hosta. Adresy klasy A używają tylko pierwszego oktetu na ID sieci, pozostawiając 24 bity na ID hosta. Adresy klasy B używają dwóch pierwszych oktetów na ID sieci, a dwóch pozostałych na ID hosta, czyli 16 bitów na każdy identyfikator. Adresy klasy C używają pierwszych trzech oktetów (24 bity) na ID sieci oraz ostatniego oktetu (8 bitów) na ID hosta.

Tabela 5.3. Standardowe długości ID sieci i hosta

Klasa adresów	Oktet 1	Oktet 2	Oktet 3	Oktet 4
A	SSSSSSS	HHHHHHH	HHHHHHH	HHHHHHH
B	SSSSSSS	SSSSSSS	HHHHHHH	HHHHHHH
C	SSSSSSS	SSSSSSS	SSSSSSS	HHHHHHH

Jesli wezmieśmy pod uwagę zarezerwowane adresy: „tylko lokalny” i „rozgłoszenia w podsieci”, których nie można używać na ID hosta, możemy wyrazić maksymalną liczbę ID hostów w sieci przez $2^x - 2$, gdzie x oznacza liczbę bitów w ID hosta.

Wobec tego, maksymalna liczba ID hostów w poszczególnych klasach adresów IP wynosi:

- ◆ Sieć klasy A: ID hosta ma 24 bity, więc $2^{24} = 16 777 216 - 2 = 16 777 214$
- ◆ Sieć klasy B: ID hosta ma 16 bitów, więc $2^{16} = 65 536 - 2 = 65 534$
- ◆ Sieć klasy C: ID hosta ma 8 bitów, więc $2^8 = 256 - 2 = 254$

O czym informuje adres IP

Adres IP jest podobny do adresu, którego używamy wysyłając przesyłkę pocztą. Adres musi zawierać pełny zbiór informacji, niezbędnych do dostarczenia przesyłki; w przeciwnym razie lista nie dotarłby do miejsca przeznaczenia. Niepełny lub nieprawidłowy adres IP ma na adresowanie IP taki sam wpływ, jak adres pocztowy na dostarczenie przesyłki. Na przykład, sam numer domu i mieszkania to za mało, aby przesyłka dotarła

do adresata, poniewaz kazde miasto moze posiadac szereg domów o takim samym numerze. Podobnie sama nazwa ulicy, na przyklad *ul. Wierzbowa*, nie wystarczy jako adres. Wedle wszelkiego prawdopodobienstwa przy ulicy Wierzbowej jest wiecej niz jeden dom.

Adres IP to polaczone ID hosta z ID sieci. Położenie hosta o adresie 47.0.0.18 nie budzi watpliwości — jest to 18. host w sieci 47.0.0. Identyfikator sieci razem z ID hosta dają pełną i jednoznaczna informację, jak dotrzeć do hosta przeznaczenia, podobnie jak adres *ul. Wierzbowa 10* stanowi jednoznaczne instrukcje co do dostarczenia poczty w obrębie miasta.

Jak stosuje się maski podsieci

Maska podsieci jest ciągły lancuchem jedynek w systemie dwójkowym, który identyfikuje, inaczej mówiąc *demaskuje* ID sieci w adresie IP. Zadaniem maski podsieci jest identyfikacja długości i wartości ID sieci. IP używa maski lokalnej podsieci w połączeniu z lokalnym adresem IP do identyfikacji lokalnej sieci. Tabela 5.4 przedstawia standardowe maski podsieci o długości 8, 16 i 24 bitów. Jak widać, pierwszy oktet w adresie klasy A przedstawionym w tabeli jest odsłoniety przez maskę podsieci zapisaną ponizej adresu. ID sieci w adresie klasy A z przykładu wynosi 11. Widac też, że dwa pierwsze oktety w adresie IP klasy B są odsłonięte przez maskę podsieci. ID sieci w adresie klasy B wynosi 131.107. Adres IP klasy C ma wszystkie oktety z wyjątkiem ostatniego odsłonięte przez maskę podsieci. ID sieci klasy C to 192.168.0.

Tabela 5.4. Standardowe maski podsieci

Klasa adresu	Adres IP Maska podsieci w zapisie dwójkowym	Adres IP Maska podsieci w zapisie dziesiętnym z kropkami
A	00001011.00000000.00000001.00010010	10.0.1.18
	11111111.00000000.00000000.00000000	255.0.0.0
B	10000011.01101011.00000010.11001000	131.107.2.200
	11111111.11111111.00000000.00000000	255.255.0.0
C	11000000.10101000.00000000.00001111	192.168.0.15
	11111111.11111111.11111111.00000000	255.255.255.0



Zadaniem maski podsieci jest identyfikacja wartości ID sieci.

Brama domyslna

W środowisku z trasowaniem pakietów każdy host standardowo posiada skonfigurowany adres routera dla danego segmentu. Adres IP routera w każdej podsieci nosi nazwę *bramy domyslnej*. Hosty w sieci X posiadają bramę domyslną 192.168.1.1, zaś hosty w sieci Y używają jako bramy domyslnej adresu 10.0.0.1. Z punktu widzenia klienta

brama domyslna sluzy do dostarczania wszelkich transmisji zdalnych. Datagramy do odleglych adresow w sieci X beda dostarczane pod adres bramy domyslnej 192.168.1.1, zas w sieci Y pod adres bramy domyslnej 10.0.0.1.

Ustalenie czy adres docelowy jest lokalny, czy zdalny

Protokol IP do ustalenia, w jakiej sieci znajduje sie lokalny host, uzywa lokalnej maski podsieci razem z adresami IP hostow lokalnych i odleglych. Datagramy przeznaczone dla sieci innych niz lokalna uznawane sa za odlegle i odpowiednio traktowane. IP uzywa funkcji boolowskiej I (AND) do porownania adresow hosta lokalnego i docelowego z lokalna maska podsieci, co daje w wyniku ID sieci dla obu hostow — lokalnego i docelowego. Jesli oba identyfikatory maja te sama wartosc, to host docelowy jest lokalny; w przeciwnym razie — zdalny. Omowlismy juz wszystkie elementy skladanki — sam proces jest bardzo prosty.

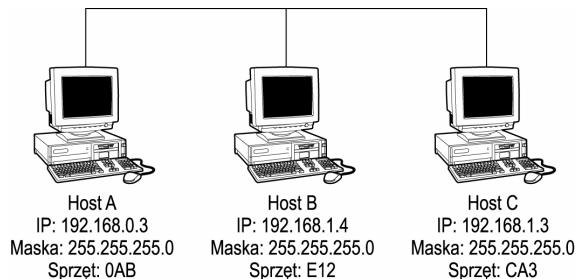


Funkcja boolowska AND porownuje liczby dwójkowe bit po bicie, dajac w wyniku „1” („prawda”) tylko wtedy, gdy wartosci obu bitow wynosza 1. Inaczej mowiac, 1 AND 1 = 1.

Na rysunku 5.3 tylko dwa z trzech hostow sa w stanie komunikowac sie ze soba lokalnie, poniewaz tylko dla nich dwóch identyfikatory sieci sa takie same. ID sieci hosta A to 192.168.0, zas hosty B i C posiadaja ID sieci równe 192.168.1. Hosty B i C sa wzgledem siebie lokalne, lecz host A nie jest. Poniewaz jego ID sieci jest inny niz dla pozostalych hostow, A uznawany jest za zdalny. Czy host jest lokalny, czy zdalny, mozna ustalic za pomoca polaczenia adresu IP i maski podsieci.

Rysunek 5.3.

Identyfikatory sieci lokalnych i zdalnych



IP zachowuje sie w sposob nastepujacy:

- ◆ Gdy host docelowy jest lokalny, IP dostarcza pakiet za pomoca protokolu ARP.
- ◆ Gdy host docelowy jest zdalny, pakiet dostarczany jest za pomoca ARP do bramy domyslnej.
- ◆ Gdy pakiet przeznaczony jest dla zdalnego hosta, a brama domyslna nie jest skonfigurowana, pakiet jest odrzucany.

Aby dostarczyc pakiet do lokalnego hosta docelowego, host IP wykonuje nastepujace operacje:

1. Pakiet zostaje przekazany do IP przez protokół z wyższej warstwy. W pakiecie określony jest docelowy adres IP. Protokół IP porównuje ID sieci hosta docelowego z ID sieci lokalnego hosta i rozpoznaje, że oba identyfikatory mają te same wartości.

Trasowanie klasowe i bezklasowe

Omówiliśmy korzystanie z masek podsieci w sieciach klas A, B i C, aby podkreślić potrzebę porównania adresu IP i maski podsieci w celu poprawnego dostarczenia datagramu w sieci z trasowaniem pakietów. Korzystanie z tych standardowych masek podsieci nosi nazwę *trasowania klasowego (classful routing)*. Choć zrozumienie tej techniki jest niezbedne, by pojąć mechanizm dostarczania pakietów, trzeba zaznaczyć, że bardzo niewiele sieci w rzeczywistości z niej korzysta, z uwagi na marnowanie adresów IP. Ruterów internetowych jako metody odzyskiwania marnowanej przestrzeni adresów IP powszechnie używa *bezklasowego trasowania międzydomenowego (CIDR — Classless Inter-Domain Routing)*. Zamiast podziału maski podsieci na oktety, CIDR dzieli ją na 32 sekcje, co pozwala na dokładniejszy dobór wielkości sieci i mniejszych niewykorzystanych adresów IP. Wyobraźmy sobie sieć zawierającą 2000 hostów i używającą zakresu adresów klasy B, co spowodowałoby zmarnowanie ponad 60 000 adresów. CIDR pozwala na wydzielenie z tejże sieci klasy B podsieci na 2046 hostów, dzięki czemu zmarnuje się jedynie około 50 adresów zamiast 60 000. Pozostałe adresy można przydzielić do innych sieci. W notacji CIDR po adresie IP zapisywany jest numer określający liczbę użytych bitów maski podsieci. Sieć 10.0.0.0 z 12-bitową maską podsieci w notacji CIDR byłaby zapisana jako 10.0.0.0/12. Trasowanie klasowe i bezklasowe opisano dokładniej w rozdziale 19.

2. Ponieważ ID sieci docelowej i lokalnej są takie same, pakiet jest przeznaczony dla hosta w lokalnej sieci.
3. Adresy IP i Ethernet hostów: źródłowego i docelowego zostają wymienione pomiędzy hostami za pomocą rozgłoszeń ARP.
4. Pakiet zostaje przesłany do warstwy łącza danych w celu przesłania do hosta docelowego.

Aby dostarczyć pakiet do zdalnego hosta, host IP przeprowadza następujące działania:

1. Pakiet zostaje przekazany do IP przez protokół z wyższej warstwy. W pakiecie określony jest docelowy adres IP. Protokół IP porównuje ID sieci hosta docelowego z ID sieci lokalnego hosta i rozpoznaje, że identyfikatory różnią się.
2. Ponieważ ID sieci docelowej i lokalnej są różne, pakiet jest przeznaczony dla zdalnego hosta.
3. Protokół IP szuka w tablicy tras trasy do sieci docelowej. Ponieważ nie znajduje jej, użycia zostaje trasa domyslna.
4. Za pomocą rozgłoszeń ARP host wymienia informacje o adresach IP i Ethernet z bramą domyslną hosta, która podana jest w trasie domyslnej.
5. Zostają użyte protokoły warstwy łącza danych, aby dostarczyć pakiet do bramy domyslnej, gdzie wchodzi on do sieci ruterów.
6. Sieć ruterów zajmuje się resztą procesu dostarczenia pakietu. Ostatni ruter posiada lokalne łącze do hosta docelowego i może dostarczyć do niego pakiet za pomocą protokołu ARP.

Podstawy trasowania

Trasowanie jest funkcja protokolu IP, która pozwala na przesyłanie pakietów pomiędzy sieciami IP. Kazdy datagram IP zawiera wewnatrz dane adresowe, które moga posluzyc ruterom do przeslania pakietu do miejsca przeznaczenia. W zaleznosci od rozmiarów posredniczacej sieci, moze byc potrzebna spora liczba ruterow, aby zapewnic dostawe pakietu. Trasowanie IP to pewnego rodzaju proces kolejnych przyblizen, w którym kazdy ruter zajmujacy sie pakietem przesyła go nieco blizej miejsca przeznaczenia. Moze my wyobrazic sobie ruter jako „straznika” mapy sieci.

Rutery sprzętowe i programowe

Rutery bazujace na komputerach (osobistych lub serwerach) sa powszechnie nazywane *ruterami programowymi* (*software router*). Ruterem nie musi byc komputer stacjonarny. W rzeczywistosci uzywanie serwera w roli ruter dla duzego srodowiska moze sprawic powazne problemy z wydajnoscia. Niektórzy producenci ruterow sprzętowych, na przyklad Cisco i Nortel Networks, posiadaja w swojej ofercie bardzo szybkie urzadzenia komunikacyjne, niezbedne do sumowania ruchu sieciowego z wielu obciazonych podsieci i posiadajace bogate zestawy funkcji, dostosowanych do określonych srodowisk.

Typy tras

Internet jest przykładem bardzo zlozonej sieci trasowanej, w której pakiety sa wysylane i odbierane przez olbrzymia sieci ruterów w lokalizacjach na całym świecie. Kazdy z tych ruterów wymaga częstych aktualizacji informacji o stanie sieci. Stan sieci zaowany jest w tablicy tras, przechowywanej w każdym ruterze. Tablica tras składa się z listy tras, opisujących najlepsze marszruty do miejsc przeznaczenia. Typy tras definiowane są przez metody, która jest stosowana do aktualizacji informacji o trasie we wszystkich ruterach. Istnieją trzy typy tras: domyslnie, statyczne i dynamiczne.



Trasy i protokoly dynamiczne omówione sa szczegółowo w rozdziale 19.

Trasy domyslne

Trasowanie stanowi funkcje IP. Kazdy host posiada tablice znanych przez siebie tras. Trasy domyslne tworzone sa w tablicy kazdego hosta IP w wyniku konfiguracji ustawien IP i sluzą do dostarczania pakietów do szeregu lokalizacji. Ponizej przedstawiony zostal przyklad tablicy tras domyslnych dla komputera Windows 95 z jednym interfejsem sieciowym, posiadajacego adres IP 131.107.2.252 i brame domyslna 131.107.2.169. Listing powstal w wyniku użycia polecenia route print, wpisanego w wierszu poleceń:

Adres sieciowy	Maska sieci	Adres bramy	Interfejs Metryka
0.0.0.0	0.0.0.0	131.107.2.169	131.107.2.252
1	127.0.0.0	255.0.0.0	127.0.0.1

	131.107.2.0	255.255.255.0	131.107.2.252	131.107.2.252
1	131.107.2.252	255.255.255.255	127.0.0.1	127.0.0.1
1	131.107.255.255	255.255.255.255	131.107.2.252	131.107.2.252
1	224.0.0.0	224.0.0.0	131.107.2.252	131.107.2.252
1	255.255.255.255	255.255.255.255	131.107.2.252	131.107.2.252
1				

Poniższa lista objasnia siedem tras domyślnych, wyświetlonych w wyniku wydania polecenia `route print`:

1. 0.0.0.0 jest wpisem domyślnym, uzywanym, gdy zaden inny nie pasuje.
2. 127.0.0.0 jest lokalnym adresem zwrotnym (*loopback*), sluzacym do wysylania pakietów do lokalnego hosta.
3. 131.107.2.0 jest trasa lokalnej podsieci.
4. 137.107.2.252 jest trasa lokalnego hosta.
5. 131.107.255.255 jest trasa rozgłoszen w podsieci.
6. 224.0.0.0 jest trasa dla adresowania grupowego (*multicast*), uzywana przez hosta do rejestracji w grupach.
7. 255.255.255.255 jest ograniczonym adresem rozgłoszeniowym.

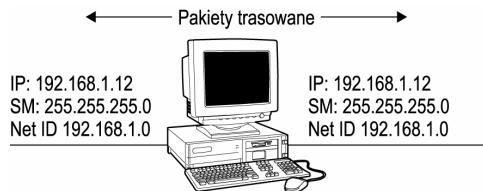
Poniższa lista zawiera opis pieciu kolumn tablicy tras:

- ♦ *Adres sieciowy* — w tablicy tras oznacza adres docelowy: hosta, podsieci lub trasy domyślnej.
- ♦ *Maska sieci* — definiuje kryteria uzycia trasy. Gdy identyfikator sieci dla docelowego adresu IP jest zgodny z czescia adresu sieciowego przykryta przez maske sieci, mozna uzyc tej trasy.
- ♦ *Adres bramy* — adres, pod który nalezy wysylac pakiet.
- ♦ *Interfejs* — adres interfejsu sieciowego, przez który nalezy wyslac pakiet.
- ♦ *Metryka* — koszt trasy. Nizsza metryka oznacza trasę preferowaną.

Trasowanie nie musi byc skomplikowane — moze to byc cos tak prostego, jak komputer z dwoma interfejsami sieciowymi (wieloadresowy), laczacy dwie sieci. Komputer ten mozemy skonfigurowac do przesyłania pakietów pomiedzy sieciami, zalaczajac przekazywanie IP. Rysunek 5.4 przedstawia komputer wieloadresowy grajacy role routera. Pakiety moga byc przesylane z interfejsu 10.0.0.2 do interfejsu 192.168.1.12 lub w przeciwnym kierunku. Do przesyłania datagramów z jednej sieci do drugiej wystarcza tutaj trasy domyslne.

Rysunek 5.4.

*Komputer
wieloadresowy*



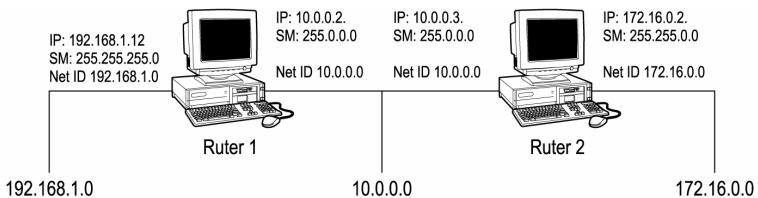
Komputer wieloadresowy posiada dwa lub wiecej interfejsów sieciowych, które zwykle podlaczone sa do różnych podsieci.

Do przesyłania pakietów pomiędzy sieciami polaczonymi przez wieloadresowy komputer nie jest wymagana zadna dodatkowa konfiguracja. Jednakze pakiety przeznaczone dla wszelkich innych sieci niz dwie wymienione w tablicy tras domyslnych beda odrzucone, poniewaz protokol IP „wie” tylko, jak znalezc dwie podsieci.

Jesli rozbudujemy odrobine siec, tak by skladala sie z trzech podsieci polaczonych dwoma ruterami, zacznie robic sie troche ciekawiej. Jak juz zostało powiedziane w tym rozdziale, IP uzywa identyfikatorów sieci do opisania lokalnych i odleglych miejsc przeznaczenia. Rutery uzywaja ID sieci do identyfikacji docelowych podsieci. Rysunek 5.5 pokazuje konfiguracje trzech sieci, w ktorej hosty z sieci 192.168.1.0 i 172.16.0.0 moga wprawdzie przesyłac pakiety do sieci 10.0.0.0, lecz nie moga komunikowac sie ze soba wzajemnie. Hosty w sieci 192.168.1.0 nie moga przesyłac pakietów do sieci 172.16.0.0 i vice versa.

Rysunek 5.5.

*Dwa routery
w trzech sieciach*

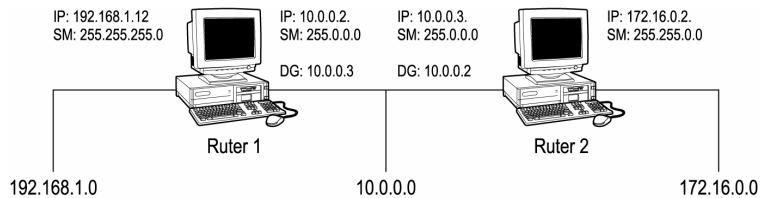


Komputery wieloadresowe moga przekazywac pakiety IP jedynie do segmentow, do których posiadaja lokalny interfejs. Gdy potrzebuja lacznosci z odleglymi podsieciами, do znalezienia kazdej sieci odleglej wymagana jest brama domyslna lub wpis w tablicy tras.

Problem trasowania do odleglych podsieci mozemy rozwiązać używając bramy domyslnej, jak na rysunku 5.6. Jesli skonfigurujemy karte sieciowa 10.0.0.3 do korzystania z 10.0.0.2 w roli bramy domyslnej oraz 10.0.0.2 do korzystania z 10.0.0.3, w rezultacie skryzyżujemy bramy domyslne. Problem z trasowaniem zostanie rozwiązany, poniewaz ruch sieciowy uznany za odległy dla danego routera bedzie przesyłany do drugiego. Ponieważ mamy do czynienia tylko z trzema sieciami, użycie bram domyslnych nie musi być złym pomysłem. Proszę jednak pamiętać, że w sieci LAN rozbudowanej do pięciu – szesciu podsieci rozwiązanie oparte na bramach domyslnych przestaje być skuteczne.

Rysunek 5.6.

*Trasowanie
za pomocą
bram domyślnych*



Ruterzy zwykłe nie posiadają skonfigurowanych bram domyślnych. Tutaj posłużyły one do prezentacji skutków działania trasy domyślnej (0.0.0.0).



Załóżmy, że host 192.168.1.14 w sieci 192.168.1.0 chce skomunikować się z hostem 172.16.0.72. Proces będzie przebiegać następująco:

1. W hostie 192.168.1.14 pakiety o adresie przeznaczenia 172.16.0.72 przekazywane są w dół z protokołu wyższej warstwy.
2. IP w hostie porównuje ID sieci docelowej z ID sieci lokalnej i ustala, iż ruch przeznaczony jest dla hosta zdalnego.
3. Zostaje użyty ARP do znalezienia bramy domyślnej (w tym przypadku 192.168.0.12) i pakiet zostaje wysłany do Ruteru 1.
4. IP w Ruterze 1 sprawdza ID sieci docelowej w odebranym pakiecie. Adres przeznaczenia (172.16.0.72) nie pasuje do ID zadnej z sieci lokalnych, więc zostaje uznany za odległy dla Ruteru 1.
5. Ruter 1 ma skonfigurowaną bramę domyślną, więc do rozwiązania jej adresu zostaje zastosowany ARP. IP w Ruterze 1 zmniejsza TTL pakietu i przekazuje ten pakiet do 10.0.0.3.
6. IP w Ruterze 2 sprawdza ID sieci w docelowym adresie IP pakietu i rozpoznaje, iż pakiet zaadresowany jest do jednej z sieci lokalnych — 172.16.0.0.
7. Zostaje użyty protokołu ARP do znalezienia adresu sprzętowego hosta docelowego.
8. Pakiet zostaje dostarczony do 172.16.0.72.

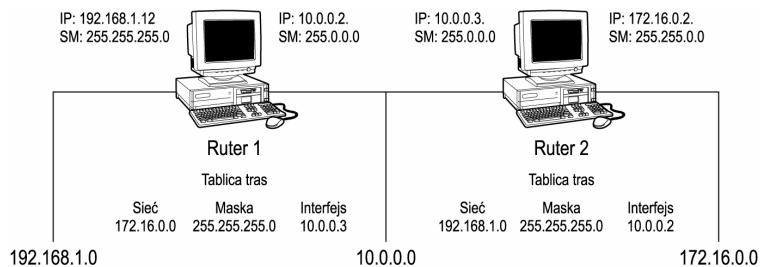
Jak widać, każdy ruter przesyła pakiety zaadresowane do zdalnych hostów do swojej bramy domyślnej. Choć prostota tego rozwiązania jest cenna, gorzej jest ze skalowalnością. Konfiguracja taka nadaje się jedynie dla małych sieci, zawierających niewiele segmentów.

Trasy statyczne

Innym sposobem znajdowania tras do wszystkich trzech sieci jest użycie tablic tras we wszystkich serwerach. Każdy wpis w tablicy tras służy do identyfikacji odlegiej sieci i wskazania adresu IP następnego routera po drodze do tej sieci. Rysunek 5.7 przedstawia wpisy w tablicach tras, wymagane dla naszej trójsegmentowej sieci.

Rysunek 5.7.

*Trasowanie
za pomocą
tras statycznych*



Ponownie załóżmy, że host 192.168.1.14 chce skomunikować się z hostem 172.16.0.72. Gdy użyjemy tablic tras zamiast bram domyślnych, proces będzie przebiegać następująco:

1. W hostie 192.168.1.14 pakiety o adresie przeznaczenia 172.16.0.72 przekazywane są w dół z protokołu wyższej warstwy.
2. Protokół IP w hostie porównuje ID sieci docelowej z ID sieci lokalnej i ustala, iż ruch przeznaczony jest dla hosta zdalnego.
3. Zostaje użyty protokół ARP do znalezienia bramy domyślnej (w tym przypadku 192.168.0.12) i pakiet zostaje wysłany do Rutera 1.
4. IP w Ruterze 1 sprawdza ID sieci docelowej w odebranym pakiecie. Adres przeznaczenia (172.16.0.72) nie pasuje do ID zadnej z sieci lokalnych, więc zostaje uznany za zdalny względem Rutera 1.
5. IP w Ruterze 1 szuka w tablicy tras trasy do sieci 172.16.0.0. Interfejsem dla tej trasy jest 10.0.0.3.
6. Zostaje użyty protokół ARP do rozwiązania adresu 10.0.0.3 i wymiany adresów sprzętowych pomiędzy obydwoma routery. Pakiet zostaje przekazany do 10.0.0.3.
7. IP w routerze 2 sprawdza ID sieci w docelowym adresie IP pakietu i ustala, iż pakiet zaadresowany jest do jednej z sieci lokalnych — 172.16.0.0. Zostaje zmniejszony TTL pakietu i przeliczona na nowo suma kontrolna.
8. Zostaje użyty ARP do znalezienia adresu sprzętowego adresata.
9. Pakiet zostaje dostarczony do 172.16.0.72.

Trasy dynamiczne

Trasy dynamiczne nie stanowią funkcji protokołu IP, lecz specjalnych protokołów trasowania, takich jak RIP (*Routing Information Protocol* — protokół informacyjny trasowania) lub OSPF (*Open Shortest Path First* — protokół wykorzystania najkrótszej ścieżki). Wszystkie protokoły trasowania dynamicznego posiadają metody udostępniania w sieci informacji o trasach, poprzez rozgłoszanie do pobliskich routrów całych tablic tras lub ich fragmentów. Ponieważ routery mogą gromadzić informacje o otaczającej sieci, odbierając rozgłoszenia swoich sąsiadów, implementacja routrów korzystających z tras dynamicznych jest ogromnie uproszczona.

W prawdziwej realności routery dynamiczne mogą korzystać z tras wprowadzanych ręcznie (na przykład statycznych), lecz główna korzyść ze stosowania routingu dynamicznego jest

zdolność sieci ruterów do reagowania na zmieniające się warunki, poprzez modyfikowanie zawartości tablic tras i metryk.



Rozdział 19. omawia protokoly trasowania dynamicznego OSPF (*Open Shortest Path First*) oraz RIP (*Routing Information Protocol*).

Fragmentacja i składanie

Niezależnie od tego, czy używamy trasowania statycznego, dynamicznego, czy tras domyślnych, musi być dostępny mechanizm regulujący rozmiary pakietów, aby pasować do sieci docelowej bez powodowania problemów lub utraty danych. Mechanizm ten nosi nazwę funkcji *fragmentacji* i *składania* protokołu IP.

Gdy ruter sprawdza nadchodzący pakiet, porównuje jego całkowitą długość z maksymalną jednostką transmisji (MTU — *Maximum Transmission Unit*) następnej sieci. Jeśli rozmiary pakietu przekraczają MTU następnej sieci, pakiet jest dzielony na fragmenty wystarczające małe, by zmieszczyć się w pakietach w następnym segmencie. Wszystkie fragmenty pakietu posiadają te same informacje w polu tożsamości, łącznie z informacją, iż stanowią część jednego pakietu, oraz unikatowa wartość przesunięcia, która służy do złożenia fragmentów pakietu we właściwej kolejności u celu jego podróży.

Zawartość datagramu IP

Diagram IP składa się z dwóch odrebnego części: nagłówka i ładunku. Nagłówek służy do sterowania zachowaniem w warstwie IP: trasowaniem, fragmentacja i tak dalej. Nagłówki i dane protokołów z wyższych warstw są zawarte w ładunku IP, czyli w obszarze danych. Niniejszy podrozdział zajmuje się strukturą pakietu IP.

Nagłówek IP

Około 20 początkowych bajtów pakietu IP zawiera ważne informacje o tym, jak należy traktować pakiet. Te informacje noszą nazwę nagłówka IP. Pola składające się na nagłówek zostały przedstawione w tabeli 5.5.

Tabela 5.5. Pola nagłówka IP

4 bity	4 bity	4 bity	4 bity	4 bity	4 bity	4 bity	4 bity
Ver.	IHL	Typ usługi		Całkowita długość			
		Identyfikator		Flagi	Przesunięcie fragmentu		
Czas życia		Protokół		Suma kontrolna nagłówka			
			Adres źródłowy				
			Adres docelowy				
			Opcje i wypełnienie				

Poniższa lista zawiera opis wszystkich tych pól z ewentualnymi objasnieniami:

- ◆ *Wersja (Ver.)* — stosowana wersja protokolu IP (obecnie 4.).
- ◆ *Dlugosc nagłówka IP (IHL — IP Header Length)* — długosc nagłówka IP mierzona w 32-bitowych słowach.
- ◆ *Typ usługi* — w razie konieczności pole to pozwala na ustalenie pierwszeństwa ruchu i wprowadzanie opóźnień.
- ◆ *Calkowita długosc* — całkowita długosc pakietu mierzona w oktetach — razem nagłówka i danych. Wartosc minimalna wynosi 576 oktetów, maksymalna 65 535 oktetów (64 kilobajty).
- ◆ *Identyfikacja* — pole zawierajace unikatowy, 8-bitowy identyfikator dla kazdego pakietu.
- ◆ *Flagi* — to 3-bitowe pole pozwala sterowac fragmentacją; decyduje, czy pakiet wolno fragmentowac (lub bardziej fragmentowac).
- ◆ *Przesuniecie fragmentu* — w przypadku pakietu pofragmentowanego, pole to sluzi do określania, gdzie dany fragment mieści sie w pakiecie, licząc od poczatku datagramu. Przesuniecie mierzone jest w jednostkach 64-bitowych.
- ◆ *Czas zycia (TTL)* — czas zycia pakietu, mierzony w hopach (przejsciuach przez kolejne routery) lub sekundach. Pakiety o TTL równym zero sa odrzucane.
- ◆ *Protokół* — to pole wskazuje nagłówek protokolu nastepnego po IP. Jesli wartosc w tym polu wynosi 6, pakiet zostanie przekazany do protokolu TCP. Typowe przykłady wartosci pola *Protokół* wymienione sa w tabeli 5.6.

Tabela 5.6. Wartosci pola protokolu w nagłówku IP

Wartosc	Protokół	Opis
1	ICMP (Internet Control Message Protocol)	Protokół komunikacyjny sterowania siecią Internet
2	IGMP (Internet Group Management Protocol)	Protokół zarządzania grupami internetowymi
6	TCP (Transmission Control Protocol)	Protokół sterujący transmisją
8	EGP (Exterior Gateway Protocol)	Zewnętrzny protokół bramowy
17	UDP (User Datagram Protocol)	Protokół datagramów użytkownika

- ◆ *Suma kontrolna nagłówka* — matematyczna suma kontrolna, przeliczana w kazdym ruterze z uwagi na zmiany informacji nagłówka.
- ◆ *Adres źródłowy* — adres IP hosta źródłowego w 32-bitowej notacji dwójkowej.
- ◆ *Adres docelowy* — adres IP hosta docelowego w 32-bitowej notacji dwójkowej.
- ◆ *Opcje i wypełnienie* — to pole moze posluzyć do zawarcia opcji wybranych przez nadawce, na przykład trasy, która pakiet powinien podać do miejsca przeznaczenia. Wypełnienie zapewnia, iż długosc nagłówka to wielokrotnosc 32 bitów.

Ladunek IP

Ladunek IP to pole o zmiennej długosci — od 8 bitów do 64 kilobajtów, łącznie z nąglówkiem IP oraz danymi wyższych warstw. Ladunek w warstwie IP składa się z nąglówek protokołów wyższych warstw, na przykład TCP lub UDP, oraz z danych aplikacji, które używają tych protokołów.

Protokół ICMP

Sieci powinny działać poprawnie przez cały czas, lecz tak nie jest. Gdy coś dzieje się nie tak w warstwie internetowej, role narzędzia do rozwiązywania problemów odgrywa protokół komunikacyjny zarządzania siecią Internet (ICMP — *Internet Control Message Protocol*). ICMP jest protokołem serwisowym, który zgłasza błędy łączności między hostami. Protokół ten został udokumentowany w RFC 792.

Przeznaczenie ICMP

W warstwie internetowej datagramy dostarczane są w sposób bezpolaczeniowy, na zasadzie „najlepiej, jak się da”. Protokół ICMP jest zestawem komunikatów, przesyłanych w datagramach IP i zdolnych do zgłoszenia błędów w dostarczaniu innych datagramów IP. Poniższa lista zawiera kilka sytuacji, z powodu których bramy lub hosty mogą wysyłać komunikaty ICMP:

- ◆ Gdy ruter lub host jest zbyt obciążony, aby móc przyjmować do buforów kolejne datagramy, komunikaty ICMP służą do zwolnienia szybkości napływanego datagramów do danego routera.
- ◆ Gdy ruter lub host znajduje lepszą trasę do miejsca przeznaczenia, może wysłać do hosta źródłowego komunikat ICMP, powiadamiający o krótszej trasie.
- ◆ Gdy host docelowy jest nieosiągalny, ostatnia brama wysyła komunikat ICMP z powrotem do hosta źródłowego, informując o niedostępności adresata.
- ◆ Gdy host lub brama przetwarza pakiet o TTL równym 0 hopów, wówczas odrzuca ten pakiet i ewentualnie wysyła komunikat ICMP do hosta źródłowego.

Komunikaty ICMP są narzędziem diagnostycznym „wbudowanym” w warstwę internetową. Jeśli dwa hosty nie są w stanie komunikować się ze sobą, komunikaty ICMP mogą pomóc w zdiagnozowaniu problemu.

Ponieważ w szybko ewoluującym środowisku może wystąpić zalew komunikatów, niedostarczenie komunikatu ICMP nie powoduje wysłania komunikatu ICMP o błędzie. Szczególnie, gdy komunikat ICMP o niedostępności hosta docelowego nie dotrze do hosta źródłowego, ten nie wysyła kolejnego komunikatu ICMP.

Pakiety ICMP

Pakiet ICMP jest zawarty w samym datagramie IP i identyfikowany przez wartość w polu *Protokół* równą 1. Pakiet ICMP zawiera 8-bitowe pola *Typ* i *Kod* oraz 16-bitowe pole sumy kontrolnej, jak pokazano w tabeli 5.7.

Tabela 5.7. Pola pakietu ICMP

8 bitów	8 bitów	16 bitów
Typ	Kod	Suma kontrolna

Pole *Typ* służy do identyfikacji typu komunikatu ICMP (patrz tabela 5.8).

Tabela 5.8. Najczęściej spotykane komunikaty ICMP

Typ	Komunikat
0	<i>Echo Reply</i> (Odpowiedź echa)
3	<i>Destination Unreachable</i> (Cel nieosiągalny)
4	<i>Source Quench</i> (Wstrzymaj przesyłanie danych, dosł. tlumienie źródła)
5	<i>Redirect</i> (Przekierowanie)
8	<i>Echo Request</i> (Zadanie echa)
11	<i>Time Exceeded</i> (Przekroczyony czas)

Typy i kody ICMP

Pole kodu zawiera w razie potrzeby dodatkowe informacje o typie komunikatu:

- ◆ *Zadanie echa* — służy do sprawdzenia laczności pomiędzy dwoma hostami. Narzedzie ping wysyla zadania echa ICMP.
- ◆ *Odpowiedź echa* — odpowiedź na komunikat *Zadanie echa*.
- ◆ *Wstrzymaj przesyłanie danych* — gdy ruter jest przeciążony ruchem z danego hosta, moze do niego wysłać komunikat *Wstrzymaj przesyłanie danych*. Komunikat ten wskazuje na zagrożenie utraty danych z uwagi na zator w ruterze.
- ◆ *Przekierowanie* — gdy ruter zna lepszą trasę do miejsca przeznaczenia, wówczas moze za pomoca komunikatu *Przekierowanie* poinformowac hosty o tej trasie.
- ◆ *Przekroczyony czas* — gdy ruter otrzymuje pakiet o TTL równym 0, moze wysłać ten komunikat do hosta źródłowego.
- ◆ *Cel nieosiągalny* — z różnych powodów moze zaistnieć niemożliwość dostarczenia pakietu. Komunikat *Cel nieosiągalny* zawiera kody wskazujące na niektóre z powodów niemożności osiągnięcia przez pakiet miejsca przeznaczenia. Kody te wymienione zostały w tabeli 5.9.

Tabela 5.9. Kody komunikatu Cel nieosiagalny

Kod	Opis
0	Sieć nieosiagalna
1	Host nieosiagalny
2	Protokół nieosiagalny
3	Nieosiagalny port u adresata
4	Wymagana fragmentacja, lecz ustawiony zakaz fragmentowania
5	Informacje źródłowe o wyborze trasy dostarczone, lecz niepoprawne

Powszechnie stosowane narzędzia ICMP

Do najczesciej uzywanych, wygodnych narzędzi ICMP naleza PING i Tracert.

PING

Narzędzie PING (*Packet InterNet Groper Utility*) jest najlepiej znany i powszechnie uzywanym narzędziem rozwiązywania problemów warstwy internetowej. PING uzywa komunikatów ICMP Echo i Odpowiedz echa do weryfikacji polaczenia pomiędzy dwoma hostami IP. W rzeczywistosci PING w powtarzajacy sie sposób wysyla z hosta źródłowego do docelowego czesc alfabetu. W hostie docelowym ICMP odpowiada na zadanie echa pakietem Odpowiedz echa. Po dotarciu danych z powrotem do hosta źródłowego, wyswietlane sa w nim różne informacje, na przykład o czasie, jaki zajela podróż pakietu tam i z powrotem. Ponizszy listing przedstawia wynik polecenia ping adresu IP 131.107.2.169, obejmujacy czas odpowiedzi i TTL pakietu.

```
C:\>ping 131.107.2.169
```

```
Badanie 131.107.2.169 z użyciem 32 bajtów danych:
```

```
Odpowiedz z 131.107.2.169: bajtów=32 czas=18ms TTL=128
Odpowiedz z 131.107.2.169: bajtów=32 czas=18ms TTL=128
Odpowiedz z 131.107.2.169: bajtów=32 czas=15ms TTL=128
Odpowiedz z 131.107.2.169: bajtów=32 czas=18ms TTL=128
```

```
Statystyka badania dla 131.107.2.169:
```

```
Pakiety: Wyslane = 4, Odebrane = 4, Utracone = 0 (0% utraconych),
Szacunkowy czas bladzenia pakietów w milisekundach:
Minimum = 15ms, Maksimum = 18ms, Srednia = 17ms
```

Tracert

Narzędzie tracert (traceroute w systemach Unix i Linux) wykorzystuje komunikat ICMP 11 (Przekroczony czas) do rozpoznania trasy do sieci docelowej. Gdy polecenie tracert podaje adres hosta docelowego, do tego hosta wyslany zostaje ciąg komunikatów zadania echa o wartosciami TTL przystepujacych o 1 i zaczynajacych sie od 1 dla pierwszego pakietu. Poniewaz kazda brama przetwarzajaca pakiet musi zmniejszyc TTL, wartosc TTL dla kolejnego pakietu osiaga 0 w kolejnej Bramie po drodze do miejsca przeznaczenia. Trasa jest ustalana poprzez badanie komunikatów o przekroczeniu czasu, nadchdzacych z kazdej Bramy po drodze pomiedzy dwoma hostami. Po-

nizszy przyklad przedstawia wynik polecenia tracert sprawdzajacego trase do popularnej witryny WWW. Prosze zwrotic uwage, jak kazda brama identyfikuje sie poprzez komunikat *Time Exceeded*.

```
Trasa sledzenia do www.yahoo.akadns.net [64.58.76.179]
przewyzsza maksymalna liczbe przeskokow 30

      1   35 ms    7 ms    6 ms  213.224.213.156
      2   10 ms    8 ms   12 ms  z.kat-ar2.do.kat-r1.tpnet.pl
[195.205.0.153]
      3   11 ms   12 ms   11 ms  z.lodz-r1.do.kat-r1.tpnet.pl
[194.204.175.159]
      4   19 ms   26 ms   14 ms  z.war-r2.do.lodz-r1.tpnet.pl
[194.204.175.119]
      5   136 ms  137 ms  129 ms  kbn-b2-pos1-o.telia.net [213.248.67.45]
      6   130 ms  128 ms  156 ms  kbn-bb2-pos1-3-0.telia.net
[213.248.64.57]
      7   222 ms  220 ms  255 ms  nyk-bb2-pos0-2.telia.net [213.248.64.46]
      8   283 ms  220 ms  228 ms  exodus.k.telia.net [213.248.82.74]
      9   220 ms  229 ms   *     64.15.224.17

     10  225 ms  226 ms  226 ms  bbr01-p0-0.stng01.exodus.net
[206.79.9.102]
     11  231 ms  226 ms  231 ms  dcr03-g6-0.stng01.exodus.net
[216.33.99.83]
     12  224 ms  221 ms  228 ms  csr22-ve242.stng01.exodus.net
[216.33.98.10]
     13  224 ms  222 ms  224 ms  216.35.210.126
     14  225 ms  226 ms  231 ms  www.yahoo.akadns.net [64.58.76.179]

Sledzenie zakonczone.
```

Protokol IGMP

Gdyby przymocowac karte sieciowa do telewizora, do którego programy bylyby przesylane przez Internet za pomoca transmisji grupowych, wówczas protokół zarzadzania grupami internetowymi (IGMP — *Internet Group Management Protocol*) mozna by uznac za tuner telewizyjny, który daje selektywny dostep do wlosciwego kanalu.

IGMP jest protokołem warstwy internetowej, który umozliwia hostom (aplikacjom) przylaczac sie do grupy multicast (adresowania grupowego) lub opuscic ja, zas w niektórych przypadkach podaje zródlo informacji grupowych. W ruterze IGMP pozwala sledzic, do których sieci trzeba wysylac transmisyje grupowe, na podstawie przynaleznosci hostów do grup. Protokół IGMP jest opisany w RFC 1112. W kontekscie lacznosci IGMP pojecia *host i aplikacja hosta* sa uzywane zamiennie.

Wprowadzenie do transmisji grupowych

Wiekszosc ruchu w sieci LAN naprawdopodobnie jest typu kierowanego (*unicast*), co oznacza przesyl danych pomiedzy dwoma urzadzeniami. Inaczej mówiac, kazdy pakiet jest nadawany raz i adresowany do określonego hosta. Czesc ruchu w sieci jest typu rozgloszeniowego. Rozgloszenie oznacza jednokrotne wyslanie pakietu adresowanego do wszystkich hostów. Kazdy host przetwarza pakiet rozgloszeniowy na wypadek, gdyby dane byly adresowane wlasnie do niego. Projektanci sieci staraja sie, w miare možliwosci, ograniczajac ruch sieciowy rozgloszen, poniewaz zwalnia on dzialanie sieci.

Liczba powszechnie używanych aplikacji multimedialnych wciąż rośnie, podczas gdy w tradycyjnych sieciach udostępniających pliki i drukarki dostępna przepustowość zmniejsza się. W przypadku aplikacji multimedialnych część hostów w sieci (lecz nie wszystkie) może wymagać dostępu do informacji ze wspólnego źródła. Gdyby przesyłać te informacje za pomocą rozgłoszeń, wówczas wszystkie routery i hosty musiałyby wszędzie przetwarzac te pakiety, nawet gdyby nie należały do grupy zamierzonych adresatów transmisji. Gdyby zastosować transmisje kierowane, wówczas informacje trzeba było przesyłać do każdego hosta z osobna. Jak widać, ani rozgłoszenia, ani transmisje kierowane nie spełniają wymagań komunikacji z wybrana grupą odbiorców. Obie metody zużywają zbyt dużo przepustowości sieci — albo w związku z koniecznością wielokrotnej transmisji danych, albo przez zalanie sieci danymi, których potrzebują jedynie pewne aplikacje hostów.

Transmisje grupowe (*multicasting*) nadają się do zastosowania w powyższym przykładzie, ponieważ pozwala na zarejestrowanie się w grupie wszystkim klientom potrzebującym informacji. Grupa taka jest znana lokalnym ruterom jako odbiorcy określonego ruchu dla danej aplikacji. Dopóki ruter posiada w swoich sieciach członków danej grupy, dopóty będzie przekazywał do nich transmisje grupowe. Serwer źródłowy musi wysłać dane tylko raz i tylko hosty należące do grupy odbiora te dane. Transmisje grupowe stanowią wiele wydajniejsze rozwiązanie dla scenariuszy, w których informacje przekazywane są z jednego hosta do wybranej grupy, niż rozgłoszenia i transmisje kierowane.

Jak IGMP współpracuje z klientami

Pakiety IGMP są zawarte w datagramach IP w prawie taki sam sposób, jak w przypadku pakietów ICMP. Transmisje grupowe używają adresów klasy D o zakresie od 224.0.0.0 do 239.255.255.255, więc pakiety IGMP mogą być przez warstwę internetową rozpoznane po adresie IP. Używane adresy grupowe są również bezpośrednio odwzorowane na udostępnione adresy Ethernet, aby wykorzystać dostarczanie danych przez warstwę laczącą danych.

Hosty, które chcą przylać się do danej grupy, powinny o tym powiadomić najbliższe routery. Również chęć opuszczenia grupy musi być sygnalizowana.



Komunikaty IGMP *Explicit Leave* (jasne opuszczenie grupy) to funkcjonalność nowszych wersji protokołu IGMP. Oryginalny IGMP w wersji 1 po prostu pozwalał na „przeterminowanie” grupy, po którym ruter przestał wysyłać dane.

Przylaczenie hosta do grupy obejmuje dwa procesy u klienta:

1. Host powiadamia ruter, że chce przylać się do odpowiedniej grupy.
2. Host dynamicznie wiazze IP z adresem grupowym, zarezerwowanym dla danej aplikacji, oraz z zarezerwowanym adresem Ethernet.

Przylaczenie do grupy odbywa się przez transmisję pakietu IGMP *Host Membership Report* (Raport członkostwa hosta). Pakiet ten zawiera adres IP pozadanej grupy.

Jak IGMP współpracuje z ruterami

Kazdy ruter okresowo odpytuje swoje sieci, aby sprawdzic, czy dostarczanie danych grupowych nadal jest wymagane. Kontrola ta odbywa sie za pomoca zapytan o czlonkostwo hosta, ktore kierowane sa pod zarezerwowany adres IP „wszystkie hosty” — 224.0.0.1 i posiadaja TTL rowny 1. Hosty przynalezace do grup odpowiadaja na ten komunikat raportem, ktorego adres docelowy odpowiada wymaganemu adresowi grupowemu.



Zapytania IGMP *Host Membership* (czlonkostwo hosta) maja TTL rowny 1, co zapobiega przekazywaniu ich do innych sieci.

Poniewaz na podstawie wynikow tych okresowych raportow rutery moga ustalic, ktore grupy sa potrzebne, zostaja odrzucone wszelkie niepotrzebne pakiety adresowane grupowo.

IGMP jest ostatnim krokiem w dostarczaniu grupowym pakietow. W duzych srodowiskach (na przyklad w Internecie) ruch grupowy moze wymagac przesypania przez wiele bram (ktorych lokalne hosty nie naleza do grupy), aby dotrcie do wlasciwej bramy, ktorej hosty naleza do grupy — patrz rysunek 5.8. Ta komunikacja i dostarczanie pakietow miedzy ruterami to zadanie grupowych protokolow trasowania (*Multicast Router Protocol*), a nie protokolu IGMP.

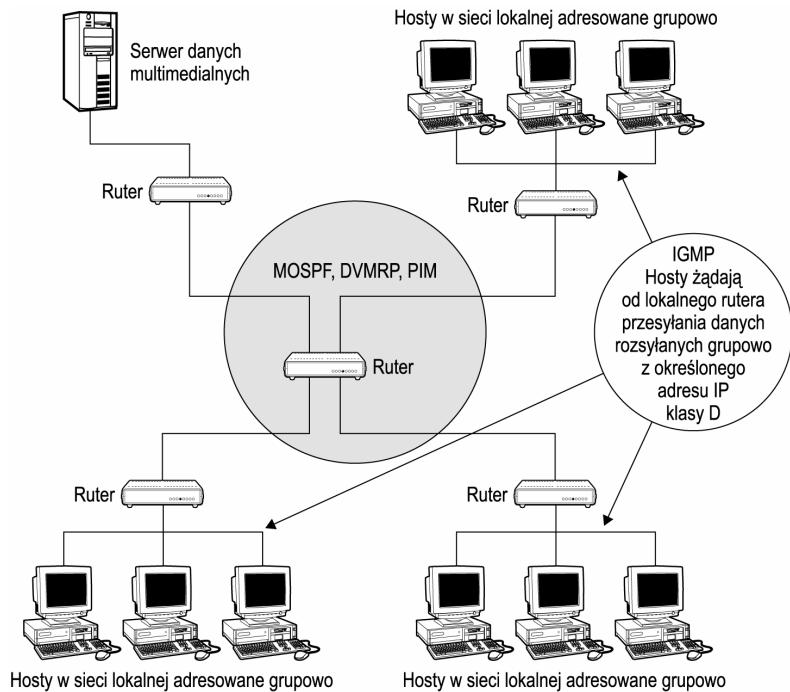
Przesywanie danych grupowych miedzy ruterami jest wynikiem dzialania protokolow typu MOSPF (*Multicast Extensions to OSPF* — rozszerzenie protokolu OSPF o adresowanie grupowe) lub DVMRP (*Distance Vector Multicast Routing Protocol* — protokol trasowania grupowego na podstawie wektorow odleglosci), a nie IGMP. Powyzsze protokoly miedzyruterowego adresowania grupowego zostaly opracowane na podstawie różnych algorytmów sterujacych tworzeniem, utrzymaniem i redukcja drzew ruterow, przez ktore przesylna jest lacznosc grupowa.

Ponizsze dokumenty RFC opisuja trzy powszechnie stosowane protokoly miedzyruterowego adresowania grupowego:

- ♦ *RFC 2117 — PIM (Protocol Independent Multicast Protocol* — protokol adresowania grupowego niezaleznego od protokolu)

Rysunek 5.8.

Adresowanie grupowe i IGMP



- ◆ *RFC 1584 — MOSPF (Multicast Extensions to OSPF — rozszerzenie protokołu OSPF o adresowanie grupowe)*
- ◆ *RFC 1075 — DVMRP (Distance Vector Multicast Routing Protocol — protokół trasowania grupowego na podstawie wektorów odległości)*

Do czego służy adresowanie grupowe

Adresowania grupowego używa się do selektywnego dostarczania danych tylko do tych hostów, które zgłosili zapotrzebowanie na te dane. Hosty mogą w każdej chwili rejestrować się w grupach multicast, aby otrzymać dane adresowane do danej grupy. Hosty mogą też w każdej chwili opuścić grupę, co jest równoznaczne z zakończeniem przyjmowania danych grupowych. Witryny WWW zwykle ogłaszają datę i godzinę specjalnych zdarzeń, do których klienci mogą „dostroić się” za pomocą adresowania grupowego.

Pakiety IGMP

Tylko dwa typy pakietów są interesujące dla klientów IGMP. Tabela 5.10 przedstawia strukturę pakietu IGMP, zawierającego 32-bitowe pole adresu grupowego, 16-bitowa suma kontrolna i dwa czerobitowe pola wersji i typu.

Tabela 5.10. Struktura pakietu IGMP

Wersja	Typ	Nie używane	Suma kontrolna
Adres grupy			

Pola pakietu IGMP maja nastepujace znaczenie i moga przyjmowac nastepujace wartosci:

- ◆ *Wersja* — okresla uzywana wersje protokolu IGMP. Mozliwe wartosci to 1, 2 i 3.
- ◆ *Typ* — oznacza jeden z dwóch mozliwych komunikatów, interesujacych dla hosta:
 - ◆ 1 — zapytanie o przynaleznosc hosta,
 - ◆ 2 — raport o przynaleznosci hosta.
- ◆ *Suma kontrolna* — wartosc obliczana matematycznie, sluzaca do zapewnienia integralnosci pakietu.
- ◆ *Adres grupy* — w zapytaniu o przynaleznosc hosta pole to jest pozostawione puste. W komunikacie raportu pole adresu zawiera adres IP zgloszonej grupy.

Rozdział 6.

Warstwa transportowa

W tym rozdziale:

- ◆ Typy przesyłu danych
- ◆ Bezpolaczeniowe przesyłanie danych
- ◆ Polaczeniowe przesyłanie danych

Warstwa transportowa, która mieści się pomiędzy warstwą aplikacji i internetową, jest sercem warstwowej architektury sieci. Warstwa ta dzieli na segmenty dane nadchodzące z warstwy aplikacji i przesyła je razem z adresem docelowym do następnej warstwy w celu transmisji. Warstwa transportowa zapewnia również *komunikację logiczną* pomiędzy procesami aplikacji uruchomionych w różnych hostach. W tym typie komunikacji procesy aplikacji w komputerach źródłowym i docelowym wprawdzie nie są połączone fizycznie, lecz komunikują się ze sobą, jakby były połączone.

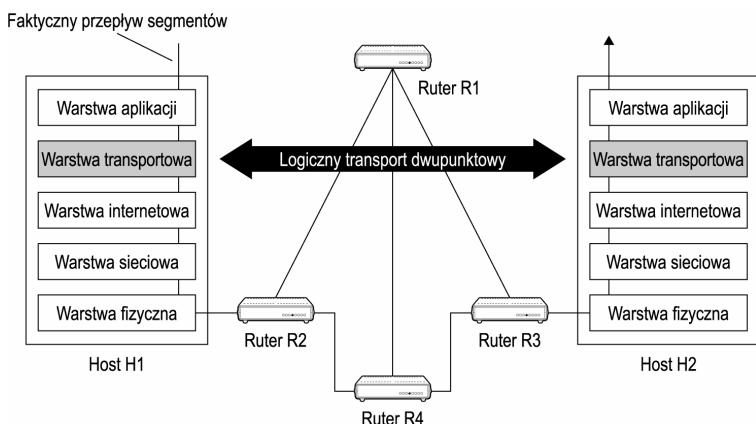
Niniejszy rozdział omawia różne typy przesyłu danych, obsługiwane przez warstwę transportową: bezpolaczeniowy protokół przesyłu danych (protokół datagramów użytkownika UDP — *User Datagram Protocol*) oraz polaczeniowy protokół przesyłu danych (protokół sterowania transmisją TCP — *Transmission Control Protocol*).

Typy przesyłu danych

Podstawowym zadaniem, które wykonuje warstwa transportowa, jest przekazywanie strumienia danych z warstwy aplikacji do warstwy transportowej w postaci segmentów. Po stronie nadawcy warstwa transportowa przekształca komunikaty odebrane od nadającego procesu aplikacji na segmenty. Segmente te zawierają dane przeznaczone do przesłania oraz nagłówki, który zawiera określone informacje, takie jak adresy źródłowy i docelowy. Po stronie odbiorcy warstwa transportowa odbiera segmenty z warstwy internetowej, składa ponownie komunikaty i przekazuje je do procesu aplikacji odbierającej dane. Procesy aplikacji wykorzystują udostępnianą przez warstwę transportową laczność logiczną do wysyłania komunikatów do siebie nawzajem, bez wchodzenia w szczególne infrastrukturę fizyczną, która służy do przesyłania komunikatów. Rysunek 6.1 przedstawia te komunikacje logiczne.

Rysunek 6.1.

Komunikacja logiczna za pomoca warstwy transportowej



Logiczna komunikacja udostepniana przez warstwe transportowa moze byc polaczeniowa (*connection-oriented*) lub bezpolaczeniowa (*connectionless*). Do polaczeniowego przesylu danych musi zostac utworzone polaczenie pomiedzy procesami aplikacji w hostach zrodlowym i docelowym, zanim bedzie mozna wyslac faktyczne dane. W bezpolaczeniowym przesyle danych nie trzeba uprzednio nawiazywac formalnych polaczen.

Przesyl danych mozna tez podzielic na wiarygodny lub nie, stanowy lub bezstanowy. *Wiarygodny* przesyl danych to taki, w którym segmenty sa dostarczone do miejsca przeznaczenia w kolejnosci, w jakiej zostaly wyslane. Z drugiej strony, przesyl *niewiarygodny* w pelni opiera sie na nizszej warstwie, wobec czego nie zapewnia dostarczenia segmentow do miejsca przeznaczenia.

Stanowy (*stateful*) przesyl danych oznacza, ze informacje zawarte w jednym zadaniu wyslanym od nadawcy do adresata moga posluzyc do modyfikacji kolejnych zadan. W przesyle *bezstanowym* informacje w konkretnym zadaniu nie moga byc wiazane z innymi, wiec nie mozna ich dalej wykorzystywac.

Istnieje kilka podstawowych problemów sieciowych, z którymi trzeba uporac sie w warstwie transportowej, aby dane byly przesypane z powodzeniem i wydajnie. Do zagadnien tych naleza:

- ◆ *Adresowanie* — aby komunikacja pomiedzy hostami byla mozliwa, musimy znac adres docelowy. Powszechnie stosowane uslugi znaja adresy wszystkich procesow w hoscie, na przyklad edytorow tekstow. Ponadto, adresy tych uslug sa znane systemowi operacyjnemu hosta. Wobec tego, gdy adresatem jest określony proces, użytkownik inicjalizujacy czynosc moze wyslac zadanie procesu do uslugi o znanym adresie. Nastepnie użytkownik pod tym adresem moze zwrócić adres procesu do użytkownika.
- ◆ *Problemy z restartem lub zerowaniem* — awarie sieci moga powodowac zerowanie lub restart polaczenia sieciowego, co z kolei moze prowadzic do utraty segmentow. Więksosc protokolow transportowych, by rozwiązac ten problem, korzysta z ponizszych sposobów:
 - ◆ W przypadku wystapienia zerowania polaczenia, uslugi sieciowe informuja o tym uczestnikow transportu przez wyslanie sygnalow. Uczestnik transportu po stronie odbiorcy potwierdza fakt zerowania i przesyla do nadawcy numer ostatniego

odebranego segmentu. Z drugiej strony nadawca powstrzymuje się od wysyłania nowych segmentów az do chwili, gdy odbierze z drugiego konca połączenia odpowiednie informacje o fakcie zerowania.

- ♦ W przypadku utraty połączenia sieciowego w nizszej warstwie, strona, która zainicjowała połączenie, musi wysłać do usług sieciowych zadanie nowego połączenia sieciowego, a następnie wysłać zadanie komunikacji do hosta po drugiej stronie.
- ♦ *Lacznosc uzywajaca kanalu bez gwarancji dostawy* — warstwa transportowa do fizycznej transmisji danych pomiędzy nadawcą i odbiorcą używa położonej ponizej warstwy internetowej, która z kolei używa warstwy sieciowej. Warstwa internetowa używa protokołu IP, który stosuje bezpołączenny mechanizm przesyłu danych bez gwarancji dostawy. Dlatego też protokół transportowy powinien w wiarygodny sposób dostarczać dane z jednej aplikacji do drugiej.
- ♦ *Multipleksowanie i demultipleksowanie* — protokoły transportowe dla komunikacji pomiędzy procesami działającymi w dwóch różnych hostach muszą udostępnić usługi multipleksowania i demultipleksowania; w przeciwnym razie laczność nie byłaby w ogóle możliwa.



Czytelnik zetknie się z pojęciami multipleksowania i demultipleksowania w różnych aspektach działania sieci komputerowych. W najbardziej ogólnym znaczeniu *multipleksowanie* oznacza łączenie wielu składników w jeden, zaś *demultipleksowanie* oznacza rozdzielenie połączonych składników.

Proces gromadzenia w hostie źródłowym danych z różnych procesów aplikacji, tworzenie segmentów i przesyłanie ich do warstwy sieciowej nosi nazwę *multipleksowania*. Proces dostarczania danych z segmentów warstwy transportowej do właściwej aplikacji nazywany jest *demultipleksowaniem*.

- ♦ *Porządkowanie*. Warstwa transportowa dzieli strumień danych, odbierany z warstwy aplikacji, na małe porcje zwane segmentami. Segmente muszą być numerowane, aby można było złożyć je razem po stronie odbiorcy. Gdyby segmenty nie były numerowane, a pakiet wysłany jako pierwszy dotarł do miejsca przeznaczenia po drugim (przez opóźnienia w sieci), dane po stronie odbiorcy zostałyby uszkodzone z uwagi na niemożność złożenia ich razem.
- ♦ *Sterowanie przepływem i buforowanie*. Warstwa transportowa w hostach po obu końcach połączenia utrzymuje ustaloną objętość pamięci — *bufor*. Jego rozmiary decydują o objętości danych, które można składować. Następnie, dane z bufora mogą być odczytywane aplikacją. Gdyby nadawca wysyłał dane nie zważając na rozmiary bufora, mogłyby nastąpić przepelnienie bufora i utratę danych, wobec czego szybkość, z jaką aplikacja odbiera dane, musi być przynajmniej równa szybkości, z jaką nadawca dane wysyła. Dopasowanie tych dwóch szybkości nosi nazwę sterowania przepływem i zapewnia wydajne dostarczanie danych bez możliwości ich utraty.
- ♦ *Kontrolowanie przeciążeń*. Gdy w sieci obecnych jest zbyt wiele pakietów, występują przeciążenia, co powoduje niską wydajność sieci. Sytuacje takie mogą wywołać kilka czynników, na przykład wolne routery lub brak wolnych buforów w routeraх. Warstwa transportowa musi podczas transmisji reagować na te problemy.
- ♦ *Powielanie*. Gdy podczas połączenia w warstwie transportowej dwie lub więcej kopii tego samego segmentu zostają wysłanych do odbiorcy, zachodzi *powielanie*.

Taka sytuacja moze wystapic z powodu odebrania wiecej niz jednego potwierdzenia dla tego samego segmentu, lub gdy nastapi ponowienie transmisji z uwagi na opóznienia w dostawie danych lub zagubione potwierdzenia. Aby uniknac bledów transmisji, warstwa transportowa musi wykrywac powielenia.

- ◆ *Strategia ponawiania transmisji* — polożony ponizej protokół IP nie gwarantuje dostawy, wobec czego warstwa transportowa potrzebuje strategii ponawiania transmisji segmentu w przypadku, gdy ten:
 - ◆ nie dotrze do miejsca przeznaczenia,
 - ◆ dotrze do miejsca przeznaczenia uszkodzony — wówczas warstwa transportowa u odbiorcy powinna wykryc błąd i odrzucic segment.
- ◆ *Przywracanie po awariach* — protokół musi radzić sobie z sytuacjami, w których jeden z systemów podczas transmisji segmentu przestanie działać. Problem staje się powazniejszy, gdy strona aktywna (nadajaca) nadal wysyla segmenty i czeka na potwierdzenia od odbiorcy, który uległ awarii.

Warstwa transportowa świadczy usługi transportowe za pomocą protokołów transportowych, do których należą UDP (*User Datagram Protocol*) oraz TCP (*Transmission Control Protocol*). Protokoły te zajmują się szeregiem podstawowych zagadnień sieciowych.

Dostawy wiarygodne i dostawy nie gwarantowane

Wiarygodne dostarczanie danych zapewnia dostawę segmentów do adresata we właściwej kolejności, bez uszkodzeń i strat. Protokół wiarygodny, taki jak TCP, bierze na siebie wszystkie problemy sieciowe, na przykład przeciążenia, sterowanie przepływem czy powielanie.

Mechanizm nie gwarantowanego dostarczania danych nie zapewnia dostawy segmentów do miejsca przeznaczenia. W tym procesie segmenty mogą ulec uszkodzeniom lub zagubieniu. Protokół bez gwarancji dostaw, jak np. UDP, zakłada, iż sieć, z której korzysta, jest całkowicie wiarygodna. W konsekwencji tego protokoły bez gwarancji nie zajmują się problemami sieciowymi typu przeciążenia, sterowanie przepływem czy powielanie. Tabela 6.1 porównuje obie metody dostarczania danych.



Wiarygodność możemy osiągnąć również stosując transport bez gwarancji dostaw, jeśli protokoły używane w warstwach poniżej transportowej są wiarygodne. Dobrym przykładem może tu być protokół TFTP (*Trivial File Transfer Protocol*).

Dostawy stanowe i bezstanowe

Stanowe dostarczanie danych opiera się na ustalaniu sesji, w których po wysłaniu porcji zadań odbierane jest potwierdzenie. Dzięki temu informacje udostępnione w jednym zadaniu mogą służyć do modyfikacji przyszłych zadań. Wyobraźmy sobie sytuację, w której musimy szukać informacji w dużej bazie danych. Jeśli użyjemy protokołu stanowego, serwer może wysłać pierwszą porcję wyników do użytkownika, pozwalając mu zacząć korzystać z informacji w czasie, gdy serwer będzie przeszukiwać resztę da-

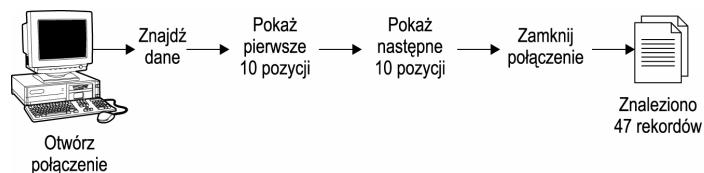
nich. Dostawy stanowe są więc wydajniejsze, ponieważ serwery tego typu mają dużą

Tabela 6.1. Porównanie mechanizmów dostaw wiarygodnych i nie gwarantowanych

Mozliwości	Dostawy wiarygodne	Dostawy nie gwarantowane
Funkcjonalność	Zapewnia dostarczenie danych do celu bez uszkodzeń i strat.	Nie zapewnia dostarczenia danych do celu.
Porządkowanie	Po stronie nadawcy pakiety są kolejno numerowane, dzięki czemu protokół wiarygodny zapewnia dostarczenie pakietów do adresata we właściwej kolejności.	Pakiety nie są numerowane, przez co dane po stronie odbiorcy mogą być pomieszczone.
Potwierdzenia	Odbiorca po otrzymaniu segmentu od nadawcy wysyła potwierdzenie, co daje wiarygodność i uniemożliwia utratę danych.	Nadawca wysyła kolejne segmenty bez potwierdzeń od odbiorcy.
Ponawianie transmisji	Jesli pakiet nie dotrze do celu lub nadjejdzie uszkodzony, nadawca ponownie wysyła pakiety zagubione lub uszkodzone.	W przypadku utraty segmentów lub wykrycia bledu transmisja nie jest ponawiana.
Wykrywanie powielień	Protokoły wiarygodne mogą wykrywać powielone pakiety, będące skutkiem retransmisji.	Dostawy nie gwarantowane nie stosują potwierdzeń i retransmisji, więc powielanie nie występuje.
Sterowanie przepływem	Protokoły wiarygodne umożliwiają sterowanie przepływem, zapobiegając utracie danych.	Protokoły bez gwarancji dostawy nie udostępniają sterowania przepływem, co może powodować utratę danych.
Kontrola przeciążeń	Wiarygodne dostarczanie danych obejmuje kontrolę przeciążeń, co rozwiązuje problemy z przeciążeniami.	Dostawy nie gwarantowane nie zapewniają kontroli przeciążeń.

wydajność; jednakże serwery stanowe są bardziej złożone z uwagi na konieczność zaimplementowania utrzymania stanów. Rysunek 6.2 przedstawia dostawę stanową. W przypadku niepowodzenia mogą zaistnieć stany niekonsekwentne (sprzeczne). Gdy z powodu awarii pojawia się stany niekonsekwentne, serwery stanowe muszą odbudować przechowywane stany, współpracując z klientami. Alternatywa może być zerwanie połączeń z klientami.

Rysunek 6.2.
Stanowe dostarczanie danych



W bezstanowym dostarczaniu danych każde zadanie jest samodzielne i nie zawiera żadnych informacji powiązanych z innymi zadaniami. Wobec tego klient musi w każdym zadaniu dostarczać do serwera pełne informacje, aby otrzymać właściwą odpowiedź, ponieważ dane w odpowiedzi opierają się jedynie na informacjach, które klient wysyła w zadaniu. Rysunek 6.3 przedstawia dostawę bezstanową. Serwery bezstanowe są proste i wytrzymałe — prawdopodobieństwo kłopotów podczas dostarczania danych jest niskie, ponieważ każde zadanie jest samodzielne. Lecz ponieważ informacji z jednego zadania nie można wykorzystać do następnych, serwery bezstanowe nie oferują zbyt

wysokiej wydajnosci. Jednakze w przypadku awarii serwery bezstanowe zachowuja sie w sposob bardziej bezproblemowy i mozna je skopowac lub zastapic.

Rysunek 6.3.

Bezstanowe dostarczanie danych

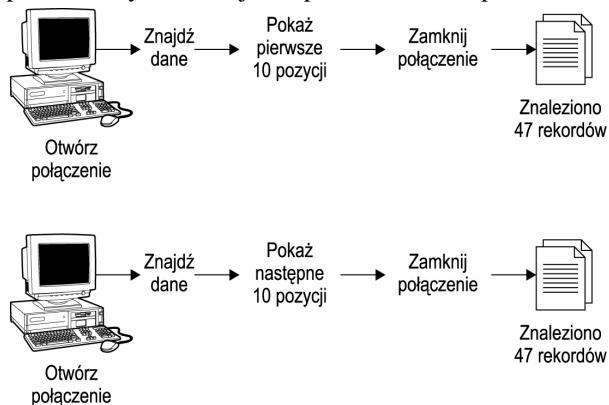


Tabela 6.2 porównuje oba typy przesyłu danych.

Tabela 6.2. Dostawy stanowe i bezstanowe — porównanie

Dostawy stanowe	Dostawy bezstanowe
Wykorzystuja mechanizm <i>sesji</i> , w których zadania sa wysylane a odpowiedzi odbierane wsadowo.	Sesje nie sa stosowane. Kazde zadanie jest niezalezne i samodzielne.
Serwery stanowe sa zlozone.	Serwery bezstanowe sa proste.
W przypadku awarii moga pojawić sie stany sprzeczne, wobec czego należy odtworzyc stany. Moze tez zostac zerwana laczosc z klientem.	W przypadku awarii serwer bezstanowy wystarczy skopowac lub zastapic.

Bezpolaczeniowe przesyłanie danych

Protokół datagramów uzytkownika (UDP) umozliwia bezpolaczeniowe, bezstanowe i nie gwarantowane dostarczanie danych pomiedzy procesami dzialajacymi w różnych hostach. Aby zapewnic laczosc pomiedzy tymi procesami, musza byc znane ich adresy. Kazdy proces w hospcie jest identyfikowany przez unikatowy ID, jednakze adresowanie procesu w hospcie moze byc problematyczne z kilku powodów:

- ◆ Procesy sa tworzone i usuwane dynamicznie.
- ◆ Proces odbierajacy dane moze zostac zastapiony przez inny bez poinformowania nadawcy. Na przyklad, wszystkie identyfikatory procesów ulegaja zmianie po restarcie komputera.
- ◆ Nadawca musi zidentyfikowac adresata na podstawie implementowanych funkcji, nie znajac procesu, który funkcje te implementuje.

UDP nie uznaje procesu jako ostatecznego miejsca przeznaczenia. Zamiast tego uzywa zbioru abstrakcyjnych punktów docelowych nazywanych *portami protokolu*. Procesy uruchomione w hospcie lacza sie z określonym portem, korzystajac z mechanizmu inter-

fejsu, udostepnionego przez lokalny system operacyjny. Porty sa buforowane, co oznacza, ze zanim proces bedzie gotowy do przyjecia danych, oprogramowanie protokolu w systemie operacyjnym przetrzymuje nadchodzace segmenty w kolejce, dopóki proces ich nie pobierze. Aby wiec komunikowac sie z obcym portem, nadawca musi znac adres IP komputera docelowego i numer portu protokolu uzywanego przez proces-adresata w tym komputerze.

UDP do przesyłania komunikatów pomiędzy komputerami uzywa protokolu IP z leżącej ponizej warstwy internetowej. Jednak UDP w przeciwienstwie do IP posiada dodatkowa zdolność rozróżniania wielu miejsc przeznaczenia w danym pojedynczym hostie, a ponadto umozliwia wyszukiwanie (lecz nie korekcje) bledów. Gdy wiec uzywamy UDP zamiast TCP, aplikacja niemal bezposrednio „rozmawia” z protokołem IP.

Dlaczego mielibysmy wybierac UDP zamiast TCP, jesli ten drugi protokół zapewnia wiarygodne przesyłanie danych? Niektóre aplikacje — na przykład DNS, SNMP i RIP — uzywaja domyslnie protokolu UDP; jest on preferowany z kilku powodów:

- ◆ *Nie trzeba nawiązywać połączeń* — dzięki temu przesyłanie danych za pomocą UDP jest szybsze. Gdyby usługa DNS uzywala protokolu połaczeniowego zamiast UDP, działałaby znacznie wolniej.
- ◆ *Nie istnieje stan połączenia* — UDP nie utrzymuje stanu połączenia i nie sledzi żadnych parametrów, takich jak bufore nadawcze i odbiorcze, parametry kontroli przekazan, czy też numery na potrzeby kolejności i potwierdzeń. Dzięki temu serwer przeznaczony na określona aplikację może obsługiwać wiele aktywnych klientów, jeśli korzysta z protokołu UDP.
- ◆ *Mniejsze rozmiary nagłówka* — każdy segment posiada 8-bajtowy nagłówek, znacznie mniejszy niż w protokole TCP. Dzięki temu dodatkowe obciążenie sieci jest mniejsze, a liczność szybka i wydajna.

Ogólnie rzecz biorąc, protokół UDP jest wybierany w przypadku zastosowań, w których szybkość i wydajność są ważniejsze od niezawodności. Aplikacje mogą jednak dokonywać wiarygodnego przesyłania danych za pomocą UDP. Samo oprogramowanie aplikacji powinno wówczas przejąć pełną odpowiedzialność za wiarygodność transmisji podczas korzystania z UDP, co obejmuje utratę komunikatów, powielanie, opóźnienia, nieudane dostawy i utratę licznosci. Proszę jednak pamiętać, że takie podejście może być niepraktyczne, ponieważ duża część odpowiedzialności spada na twórców aplikacji. Co gorsza, ponieważ oprogramowanie sieciowe jest często testowane na wiarygodnych i mało obciążonych sieciach LAN, procedura testowania może nie obejmować potencjalnych problemów. Wobec tego protokół UDP jest stosowany wszędzie tam, gdzie niezawodność sieci nie stanowi większego problemu, zas najważniejsza jest prędkość transmisji.

Protokół UDP należący do warstwy transportowej dzieli strumień danych na segmenty nazywane *datagramami użytkownika*. Rysunek 6.4 przedstawia format datagramu UDP.

Rysunek 6.4.

*Format
datagramu UDP*

0	16	31
Port źródłowy UDP	Port docelowy UDP	
Długość komunikatu UDP	Suma kontrolna UDP	
	Dane	
		...

Datagram UDP składa się z nagłówka i obszaru danych UDP. Dane aplikacji zajmują pole danych. Nagłówek zawiera cztery szesnastobitowe pola:

- ◆ *Port źródłowy* — pole to zawiera numer procesu uruchomionego w komputerze źródłowym.
- ◆ *Port docelowy* — pole to zawiera numer portu używanego przez proces w komputerze docelowym. Numery obu portów — źródłowego i docelowego — są niezbedne dla UDP w celu multipleksowania i demultipleksowania, a co za tym idzie, do przesłania danych.
- ◆ *Dlugosc* — całkowita długość datagramu.
- ◆ *Suma kontrolna UDP* — służy do wykrywania błędów. Pole to zawiera uzupełnienie do 1 sumy wszystkich 16-bitowych słów w segmencie. Uzupełnienie do 1 oznacza konwersje wszystkich zer na jedynki i odwrotnie. Na przykład, jeśli suma wszystkich 16-bitowych słów modulo 16 wynosi 1100101011001010, suma kontrolna będzie wynosić 0011010100110101. Po stronie odbiorcy wszystkie 16-bitowe słowa zostają zsumowane, łącznie z sumą kontrolną. Jeśli nie wystąpił żaden błąd, wynik powinien wynosić 1111111111111111. Obecność choćby jednego zera wskazuje na błąd.

Polaczeniowe przesyłanie danych

Należący do warstwy transportowej protokół TCP jest zorientowany na połaczenie, wiarygodny i stanowy. TCP nawiązuje połaczenie pomiędzy procesami w hostach źródłowym i docelowym, zanim wysła faktyczne segmenty zawierające dane. Po ustaleniu połaczenia dane można przesyłać pomiędzy dwoma hostami w obu kierunkach — jest to proces noszący nazwę *pełnoduplexowej transmisji danych*. Na przykład, jeśli połaczenie zostało nawiązane pomiędzy procesem A w hostie źródłowym i procesem B w hostie docelowym, dane mogą być równoczesnie przesyłane zarówno od A do B, jak i od B do A. Aby nawiązać połaczenie pomiędzy dwoma procesami w różnych hostach, wymagane są następujące źródła identyfikacji:

- ◆ *Numery portów TCP* — unikalowo identyfikują proces w danym hostie. Porty nadawcy i odbiorcy nie muszą mieć tego samego numeru.
- ◆ *Gniazda TCP* — gniazdo (*socket*) TCP stanowi połaczenie adresu IP komputera i numeru portu TCP dla procesu w tym komputerze. Aby nawiązać połaczenie TCP, aplikacja musi zazadac od protokołu TCP unikalowego gniazda TCP — ten proces nosi nazwę *otwarcia gniazda*. Aby więc połaczenie było udane, aplikacja musi znać gniazdo TCP w komputerze źródłowym i docelowym.

Ponieważ TCP jest protokołem wiarygodnym, zajmuje się wszystkimi problemami sieciowymi: kontrola przeciżenia, porządkowaniem i sterowaniem przepływem. Połączanie

TCP obejmuje zawsze pojedynczego nadawcę i pojedynczego odbiorcę — jest to połączenie dwupunktowe (*point-to-point*). Ponizszy punkt opisuje szczegółowo protokół TCP.

Iinicjacja sesji

Aplikacja w komputerze, który chce wysłać dane do innej aplikacji w innym komputerze, przesyła dane do warstwy transportowej. Protokół TCP w tej warstwie odbiera dane od aplikacji i dzieli je na małe fragmenty, zwane *segmentami TCP*. TCP zamkra te segmenty w datagramach IP, które następnie zostają przesłane przez sieć. Zanim jednak hosty zaczynają wysyłać dane, muszą dokonać wzajemnych uzgodnień. W trakcie nawiązywania połączenia trzeba ustalić pomiędzy nadawcą i odbiorcą pewne parametry jakości połączenia. Noszą one nazwę *parametrów jakości usługi* (QoS — *Quality of Service*), zas procezu uzgadniania QoS pomiędzy hostem źródłowym i docelowym nazywany jest negocjacja opcji. Parametry QoS zapewniają określony poziom standardu jakości dla transmisji danych. Poszczególne parametry QoS to:

- ◆ *Opóźnienie nawiązania połączenia (connection establishment delay)* — czas, jaki upływa pomiędzy wysłaniem zadania połączenia transportowego i otrzymaniem potwierdzenia. Im krótsze opóźnienie, tym lepsza usługa.
- ◆ *Prawdopodobienstwo niepowodzenia nawiązania połączenia (connection establishment failure probability)* — prawdopodobieństwo, iż połączenie nie zostanie ustanowione w dopuszczalnym czasie opóźnienia nawiązania połączenia.
- ◆ *Przepustowość (throughput)* — liczba bajtów danych przesyłanych w ciągu sekundy. Przepustowość mierzona jest niezależnie dla każdego kierunku transmisji.
- ◆ *Opóźnienie przejścia (transit delay)* — czas upływający od wysłania komunikatu ze źródła do odebrania komunikatu przez adresata. Parametr ten, podobnie jak przepustowość, jest mierzony odrewnie dla każdego kierunku.
- ◆ *Stopa błędów (residual error rate)* — liczba utraconych lub zniekształconych komunikatów w stosunku do całkowitej liczby komunikatów wysłanych w określonej jednostce czasu. W warunkach idealnych stopa błędów powinna być zerowa, lecz w praktyce skonczona wartość stopa błędów jest dopuszczalna i akceptowana.
- ◆ *Prawdopodobienstwo niepowodzenia przesyłu (transfer failure probability)* — podczas nawiązywania połączenia zostają uzgodnione: poziom przepustowości, opóźnieni przejścia i stopa błędów. Prawdopodobieństwo niepowodzenia przesyłu oznacza odsetek sytuacji, w których uzgodnione założenia nie zostały osiągnięte podczas czasu obserwacji.
- ◆ *Opóźnienie zwolnienia połączenia (connection release delay)* — czas upływający pomiędzy inicjacją zwolnienia połączenia i faktycznym zwolnieniem.
- ◆ *Prawdopodobienstwo niepowodzenia zwolnienia połączenia (connection release failure probability)* — prawdopodobieństwo, iż połączenie nie zostanie zwolnione w dopuszczalnym czasie.
- ◆ *Ochrona (protection)* — ten parametr podawany jest w celu ochrony przed odczytem lub modyfikacją przesyłanych danych przez niepowołane osoby trzecie (podsluch).
- ◆ *Priorytet (priority)* — ten parametr zapewnia obsługę połączeń o wysokim priorytecie przed połączeniami o niskim priorytecie.

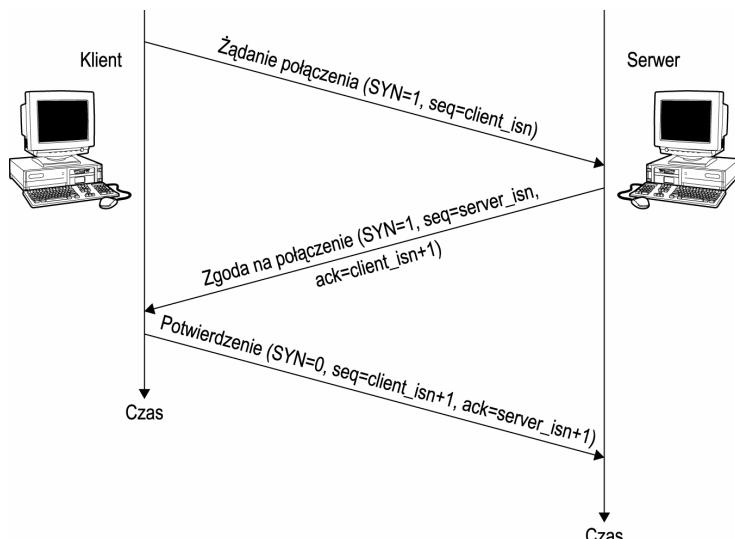
- ◆ *Odpornosc (resilience)* — ten parametr oznacza prawdopodobienstwo przerwania polaczenia przez warstwe transportowa z uwagi na problemy wewnętrzne lub przeciazenie.

Protokół TCP funkcjonuje wewnatrz hostów i jest implementowany po obu koncach logicznego polaczenia w warstwie transportowej. W trakcie nawiazywania polaczenia TCP obie jego strony inicjalizuja szereg zmiennych stanu TCP. Do zmiennych tych naleza dopuszczalna liczba nie potwierdzonych segmentów i maksymalny ruch sieciowy, jaki host moze wyslac polaczeniem skojarzonym z polaczeniem TCP. Ponizej przedstawiona jest procedura nawiazywania polaczenia pomiedzy dwoma hostami.

Host, który inicjuje polaczenie, nosi nazwe *klienta*, zas host odpowiadajacy na zadania klienta jest *serwerem*. Aplikacja klienta w pierwszej kolejnosci powiadamia protokół TCP klienta, ze chce nawiazac polaczenie z procesem w serwerze, a nastepnie TCP klienta ustanawia polaczenie TCP z protokolem TCP w serwerze. Nawiazanie polaczenia obejmuje kilka krokow, pokazanych na rysunku 6.5:

Rysunek 6.5.

Trójkierunkowe potwierdzenie TCP



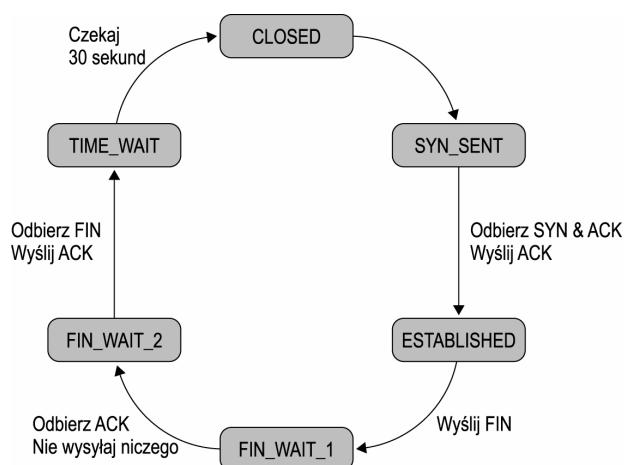
1. TCP klienta wysyla do TCP serwera specjalny segment, zapakowany w datagram IP. Ten segment zawiera poczatkowy numer sekwencji klienta (*client_isn*) oraz bit SYN o wartosci ustalonej na 1. Bit SYN okresla status synchronizacji; wartosc 1 oznacza, ze hosty nie sa zsynchronizowane i zadane jest nawiazanie polaczenia. Ten specjalny segment nosi nazwe segmentu synchronizacji (*SYN segment*) i nie zawiera zadnych danych aplikacji, poniewaz przed wyslaniem danych polaczenie musi zostac nawiazane. Oprócz tego klient wysyla rozmiar okna, który okresla po stronie klienta rozmiary bufora, sluzacego do skladowania segmentow otrzymywanych od serwera.
2. Gdy datagram IP dociera do hosta serwera, ten wyciaga z datagramu segment TCP SYN, przydziela do polaczenia bufory TCP i zmienne stanu, oraz potwierdza odbior wysylajac do TCP klienta segment „polaczenie przyznane” (*SYNACK segment*). W celu potwierdzenia serwer w segmencie SYNACK umieszcza

wartosc $client_isn + 1$. Segment SYNACK nadal posiada bit SYN ustawiony na 1 i zawiera poczatkowy numer sekwencji serwera ($server_isn$). Segment zawiera komunikat, powiadamiajacy klienta, iz serwer otrzymal pakiet SYN klienta z poczatkowym numerem sekwencji klienta ($client_isn$), oraz ze protokol TCP w serwerze zgadza sie na nawiazanie tego polaczenia z poczatkowym numerem sekwencji serwera ($server_isn$). Ponadto serwer wysyla rozmiar okna, który okresla po stronie serwera rozmiary bufora sluzacego do skladowania segmentow otrzymywanych od klienta.

3. Klient po otrzymaniu od serwera segmentu SYNACK również przydziela po swojej stronie bufory i zmienne stanu na potrzeby polaczenia. Host-klient wysyla nastepnie do serwera kolejny segment z bitem SYN ustawionym na 0, poniewaz polaczenie zostało nawiazane. Ten ostatni segment potwierdza odbiór segmentu SYNACK, gdyz zawiera wartosc $server_isn + 1$.

Procedura nawiazywania polaczenia wymaga w sumie przeslania trzech segmentow pomiedzy hostami — klientem i serwerem, dlatego proces ten nosi nazwe potwierdzenia trójkierunkowego (*three-way handshake*). Po nawiazaniu polaczenia serwer i klient moga wysylac do siebie nawzajem segmenty zawierajace dane. Jednakze protokol TCP dzialajacy w kliencie i serwerze w trakcie trwania polaczenia przechodzi kolejno różne etapy, zwane *stanami TCP*. Jak widac na rysunku 6.6, TCP klienta przechodzi sekwencje stanów TCP w nastepujacej kolejnosci:

Rysunek 6.6.
Sekwencja stanów TCP
w kliencie

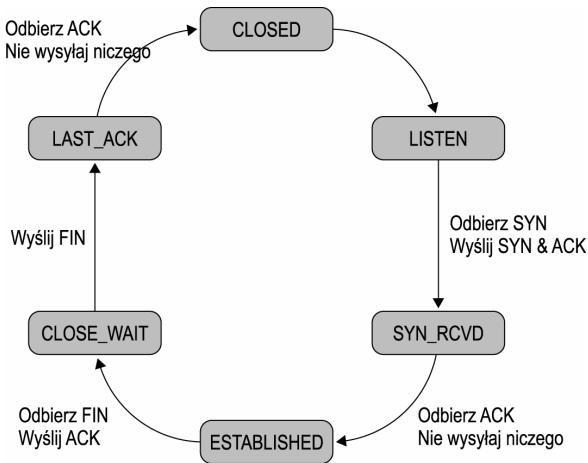


1. **CLOSED** (*polaczenie zamknięte*) — gdy proces aplikacji w jednym hostie chce zainicjując polaczenie z procesem aplikacji w innym hostie, po stronie klienta inicjowane jest owo polaczenie TCP.
2. **SYN_SENT** (*segment SYN wysłany*) — TCP klienta wysyla segment SYN do TCP serwera, po czym klient wchodzi w stan SYN_SENT. W tym stanie TCP klienta czeka na potwierdzenie od serwera, a bit SYN jest ustawiony na 1.
3. **ESTABLISHED** (*polaczenie nawiazane*) — po otrzymaniu przez klienta segmentu od serwera, klient wchodzi w stan ESTABLISHED. W tym stanie klient TCP moze wysylac i odbierac segmenty TCP zawierajace dane tworzone przez aplikacje.

4. ***FIN_WAIT_1*** (*oczekiwanie na zakoñczenie połaczenia*) — gdy aplikacja klienta zdecyduje sie zamknac połaczenie, TCP klienta wysyla do serwera segment z bitem FIN o wartosci 1. Ten stan nosi nazwe ***FIN_WAIT_1*** i TCP klienta czeka w nim na potwierdzenie z serwera.
5. ***FIN_WAIT_2*** — gdy TCP klienta otrzyma potwierdzenie, wchodzi w stan ***FIN_WAIT_2***. W stanie tym klient nie wysyla niczego do serwera i czeka na otrzymanie od serwera bitu FIN ustawionego na 1.
6. ***TIME_WAIT*** — gdy klient otrzyma od serwera bit FIN równy 1, wysyla potwierdzenie do serwera i wchodzi w stan ***TIME_WAIT***. Po odczekaniu około 30 sekund połaczenie zostaje formalnie zamkniête, wszystkie zasoby po stronie klienta zostaja zwolnione i klient wchodzi w stan ***CLOSED***.

Podobnie jak klient, serwer TCP równiez przechodzi przez rózne stany TCP. Jak widac na rysunku 6.7, serwer TCP przechodzi stany TCP w nastepujacej kolejnosci:

Rysunek 6.7.
Sekwencja stanów TCP
w serwerze



1. ***CLOSED*** (*połaczenie zamkniête*) — nie ma połaczenia pomiędzy procesami aplikacji hostów — klienta i serwera.
2. ***LISTEN*** (*oczekiwanie na transmisje*) — aplikacja w serwerze tworzy gniazdo nasluchujace i oczekuje na transmisje pod określonym numerem portu.
3. ***SYN_RCV*** (*pakiet SYN odebrany*) — po otrzymaniu segmentu SYN od klienta, serwer wchodzi w stan ***SYN_RCV***. W tym stanie serwer wysyla do klienta segment SYNACK („połaczenie przyznane”).
4. ***ESTABLISHED*** (*połaczenie nawiazane*) — serwer po odebraniu potwierdzenia segmentu SYNACK wchodzi w stan ***ESTABLISHED***.
5. ***CLOSE_WAIT*** (*oczekiwanie na zamkniecie*) — serwer po odebraniu od klienta segmentu z bitem FIN ustawionym na 1 wchodzi w stan ***CLOSE_WAIT***. Znajdując sie w tym stanie, serwer potwierdza odbiór sygnalu.
6. ***LAST_ACK*** (*ostatnie potwierdzenie*) — serwer wchodzi w stan ***LAST_ACK*** po wysłaniu bitu FIN do klienta. Po otrzymaniu od klienta ostatniego potwierdzenia, połaczenie jest formalnie zamkniête.

Maksymalny rozmiar segmentu

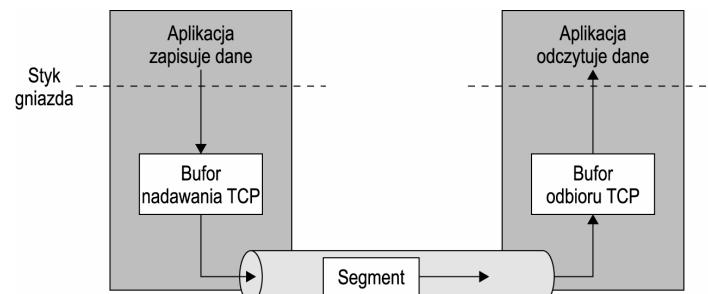
Po nawiązaniu połączenia, pomiędzy procesami aplikacji może już zaczynać się odbywać faktyczne przesyłanie danych. Jak już wspomniano, dane odebrane od aplikacji są dzielone na małe segmenty. Po stronie odbiorcy segmenty muszą zostać ponownie złożone w całość. Ponieważ jednak objętość danych, jakie można złożyć, jest ograniczona, rozmiar porcji danych musi być ograniczony do określonej wartości. Największy dopuszczalny rozmiar porcji danych nosi nazwę *maksymalnego rozmiaru segmentu* (MSS — *Maximum Segment Size*). Domyslna wartość MSS dla TCP wynosi 536 bajtów, wobec czego protokół TCP po odebraniu danych od aplikacji, dzieli je na porcje nie większe niż 536 bajtów.

W trakcie nawiązywania połączenia, TCP udostępnia opcje pozwalające ustalić MSS dopuszczalny dla danego połączenia. Parametr MSS jest przesyłany od odbiorcy do nadawcy i oznacza maksymalny rozmiar segmentu (X), jaki odbiorca może przyjąć. Wartość X może być wyższa lub niższa od domysłnej wartości MSS.

Okna nadawania i odbioru TCP

Protokół TCP implementuje sterowanie przepływem, wysyłając segmenty w zależności od rozmiaru bufora u odbiorcy. Gwarantuje to, iż bufor po stronie odbiorcy nie zostanie przepelny i segmenty nie będą tracone. Jak już wspomniano, rozmiar bufora (inaczej *rozmiar okna* lub *okno odbioru*) ustalany jest pomiędzy klientem i serwerem w trakcie nawiązywania połączenia. Rysunek 6.8 przedstawia buforowanie po stronie nadawcy i odbiorcy.

Rysunek 6.8.
Bufory nadawania
i odbioru



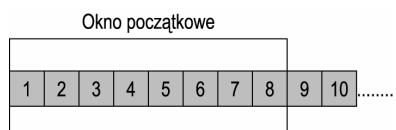
Aby dane były dostarczane w sposób wiarygodny, klient musi otrzymać od odbiorcy potwierdzenie każdego wysłanego przez siebie segmentu. Ponieważ klient musi czekać na potwierdzenie od serwera przed nadaniem kolejnego segmentu, proces ten może prowadzić do wolnego przesyłu danych oraz niepełnego wykorzystania zasobów sieciowych. Aby zminimalizować czas jadowy sieci i zapewnić wydajny i wiarygodny przesył danych, TCP wykorzystuje idee *okien przesuwnych* (*sliding window*). W oknie przesuwnym przed oczekiwaniem na potwierdzenie nadawanych jest kilka segmentów. Liczba segmentów, jaka nadawca może wysłać w określonym połączeniu, zanim otrzyma potwierdzenie od odbiorcy wskazujące, iż ten otrzymał przynajmniej jeden segment danych, nosi nazwę *okna nadawania* (*send window*).



W jednym komunikacie mozna potwierdzic odbior kilku segmentow.

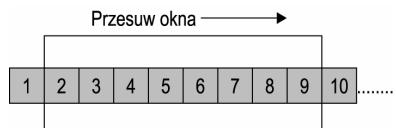
Rysunek 6.9 przedstawia okno nadawania TCP. Okno to ma staly rozmiar, a wszystkie segmenty mieszczace sie wewnatrz okna mozemy nadac nie czekajac na potwierdzenie. Na przyklad, jesli rozmiar okna wynosi 8, nadawca ma prawo wyslac 8 segmentow przed otrzymaniem potwierdzenia.

Rysunek 6.9.
Okno nadawania TCP



Gdy nadawca otrzyma potwierdzenie pierwszego segmentu w oknie nadawania, okno przesuwa sie i kolejny segment zostaje wyslany, jak na rysunku 6.10. Jesli nadawca otrzyma potwierdzenie dla kilku segmentow, na przyklad trzech, okno odpowiednio przesuwa sie i zostaja wyslane trzy segmenty. Jednakze liczba segmentow, ktore mozna wyslac, jest zalezna od okna odbioru. Po stronie serwera proces aplikacji odczytuje dane z bufora z okreslona predkoscia, wobec czego rozmiar okna odbioru jest zalezny od szybkosci, z jaka dane sa odczytywane. Gdy serwer wysyla potwierdzenie segmentow danych do klienta, razem z potwierdzeniem oglaszany jest rozmiar okna. Klient wysyla nastepnie segmenty z okna nadawania tak, by nie przepelnic okna odbioru po stronie serwera.

Rysunek 6.10.
Okno przesuwne



Protokol TCP po stronie nadawcy zawsze „pamieta”, których segmentow odbior zostal potwierdzony oraz utrzymuje osobny czasomierz dla kazdego nie potwierzonego segmentu. Jesli segment zostal utracony i dopuszczalny czas uplynie, nadawca wysyla segment ponownie.

Protokol TCP po stronie odbiorcy utrzymuje analogiczne okno, sluzace do przyjmowania i potwierdzania segmentow w miare ich nadchodzenia. Wobec tego segmenty sa podzielone na trzy zestawy:

- ◆ segmenty na lewo od okna, ktore zostaly pomyslnie przeslane i potwierdzone,
- ◆ segmenty na prawo od okna, ktore nie zostaly jeszcze wyslane,
- ◆ segmenty wewnatrz okna, bedace w trakcie przesyłania.

Okno przeciazenia

Przeciazenia powoduja opoznienia w dostarczaniu danych. Sytuacja staje sie jeszcze gorsza, gdy protokol TCP stosuje odliczanie dopuszczalnego czasu i ponowne transmisje w przypadku utraconych segmentow. Aby uniknac przeciazenia, klient musi „pamietac”

rozmiar okna odbioru. Ponadto rozmiar okna nadawania jest zmniejszany w zależności od poziomu przeciążenia. Takie zmniejszone okno nadawania nosi nazwę *limitu okna podczas przeciążenia lub okna przeciążenia*. Okno przeciążenia jest mechanizmem kontroli przeciążeń, wymuszonym przez nadawcę i opartym na szacunku przeciążenia sieci według nadawcy. Z drugiej strony okno odbioru jest mechanizmem kontrolnym stosowanym przez odbiorcę i opartym na ocenie dostępnej objętości wolnego miejsca w buforze. Dopuszczalny rozmiar okna jest zawsze mniejsza z dwóch wartości: okna odbioru ogłoszonego przez odbiorcę i okna przeciążenia.

W stanie ustalonym, gdy nie występują przeciążenia, rozmiar okna przeciążenia jest równy rozmiarowi okna odbiorcy. Jednakże w razie zatorów rozmiar okna jest zmniejszany. Do oszacowania rozmiaru okna przeciążenia protokół TCP stosuje następującą strategię:

1. Redukcja okna przeciążenia o połowe po każdej utracie segmentu.
2. Jeśli straty dalej występują, rozmiar okna zmniejszany jest wykładniczo.
3. W ostateczności transmisja zostaje ograniczona do pojedynczych segmentów, a dopuszczalne czasy oczekiwania przed retransmisją są nadal podwajane.

Algorytm powolnego startu

TCP utrzymuje kontrolę nad przeciążeniami przez wykładnicze zmniejszanie rozmiaru okna przeciążenia. Gdyby po wyeliminowaniu stanu przeciążenia protokół TCP usiłował wrócić do poprzedniego stanu przez odwrotność tej wykładniczej redukcji okna przeciążenia, mogłyby to spowodować niestabilną sytuację — oscylacje pomiędzy stanem przeciążenia i brakiem ruchu. Wobec tego, po „rozdławianiu” przeciążenia TCP powraca do poprzedniego stanu stosując tzw. *algorytm powolnego startu*. Podczas powolnego startu początkowy rozmiar okna przeciążenia TCP wynosi jeden segment i jest zwiększany o jeden segment po każdym otrzymanym potwierdzeniu. Protokół TCP rozpoczętajac przesył danych w połączeniu zawsze stosuje ten algorytm, niezależnie od tego, czy jest to nowe połączenie, czy wracające do normy po przeciążeniu.

Nagłówek TCP

Segment TCP składa się z pół nagłówka i pola danych. Dane aplikacji są dzielone na małe porcje i umieszczane w polu danych segmentów TCP. Wielkość porcji jest ograniczona przez maksymalny rozmiar segmentu (MSS), wobec czego na pełne dane aplikacji składa się większa liczba segmentów TCP. Pola nagłówka zawierają informacje związane z zarządzaniem połączeniem i sprawdzaniem błędów. Rysunek 6.11 przedstawia strukturę segmentu TCP.

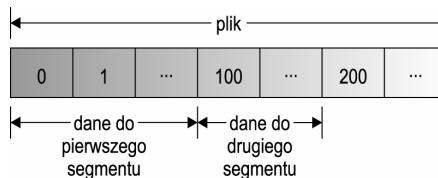
W porównaniu z nagłówkiem UDP (8 bajtów), nagłówek TCP jest długi (20 bajtów). Nagłówek TCP, podobnie jak nagłówek UDP, zawiera numery portów źródłowego i docelowego, służące do multipleksowania i demultipleksowania. I podobnie jak nagłówek UDP, zawiera pole sumy kontrolnej służącej do wykrywania błędów (lecz nie do korekcji). Oprócz wymienionych powyżej, nagłówek zawiera następujące pola:

Rysunek 6.11.
Struktura
segmentu TCP

Numer portu źródłowego	Numer portu docelowego		
Numer sekwencji			
Numer potwierdzenia			
Długość nagłówka	Nie wykorzystane	Flagi	Rozmiar okna odbioru
Suma kontrolna warstwy internetowej		Wskaźnik do pilnych danych	
Opcje		Dane	
32 bity			

- ♦ *Numer sekwencji* — 32-bitowe pole zawierajace numer sekwencji segmentu. Pierwszy bajt w potoku to *numer sekwencji* segmentu. Na przyklad, jesli potok danych zawiera plik o dlugosci 100 000 bajtów, a MSS wynosi 1000 bajtów, wówczas TCP podzieli dane na 100 segmentów. Jak widac na rysunku 6.12, pierwszy segment otrzymuje numer 0, jesli pierwszy bajt w potoku ma numer 0, nastepny segment otrzymuje numer 100 i tak dalej.

Rysunek 6.12.
Podzial
na segmenty TCP



- ♦ *Numer potwierdzenia* — 32-bitowe pole, zawierajace numer potwierdzenia segmentu. *Numer potwierdzenia* oznacza nastepny bajt, którego serwer oczekuje od klienta. Wezmy na przyklad pod uwage polaczenie TCP nawiiazane pomiędzy hostami A i B. Poniewaz TCP udostepnia transmisje pelnodupleksowa, dane moga byc przesylane w obu kierunkach — od hosta A do B i vice versa. Zalozmy, ze host A chce wyslac do hosta B segment zawierajacy bajty o numerach kolejnych od 0 do 535. Poniewaz kolejny bajt, którego oczekuje host B, na numer 536, host A umieszcza wartosc 536 w polu numeru potwierdzenia nastepnego segmentu, który zamierza wyslac. Oba pola — numeru sekwencji i numeru potwierdzenia — sa niezmiernie wzorne, poniewaz opiera sie na nich usluga wiarygodnego dostarczania danych.
- ♦ *Dlugosc nagłówka* — 4-bitowe pole, zawierajace dlugosc nagłówka TCP mierzona w 32-bitowych slowach. Nagłówek moze miec różne dlugosci, z uwagi na pole *Opcje* (omówione ponizej).
- ♦ *Flagi* — 6-bitowe pole, zawierajace określone bity znacznikowe:
 - ♦ *URG* — oznacza, ze dane w segmentie zostaly oznaczone przez warstwe aplikacji nadawcy jako „pilne”.
 - ♦ *ACK* — wskazuje, ze wartosc w polu numeru potwierdzenia segmentu jest obowiązujaca.
 - ♦ *PSH* — wskazuje, iz odbiorca powinien natychmiast przeslac dane w segmentie do warstwy powyzej.
 - ♦ *RST* — wskazuje, iz polaczenie jest w stanie zerowania.

- ♦ *SYN* — oznacza, iż należy nawiązać połączenie.
- ♦ *FIN* — oznacza konieczność zamknięcia połączenia.
- ♦ *Rozmiar okna odbioru* — 16-bitowe pole używane przez usługę sterowania przepływem protokołu TCP. Jak już wspomniano, TCP jest pełnoduplexowy i obie strony połączenia utrzymują własny bufor. Wobec tego, aby uniknąć przepelnienia danymi, rozmiar obu buforów odbiorczych nie powinien być mniejszy od objętości wysłanych danych. To pole zawiera rozmiar okna odbioru, ulegający zmianom w całym okresie istnienia połączenia. Jeśli wartość zawarta w tym polu wynosi 0, bufor odbiorczy jest pełny — a co za tym idzie, kolejny segment nie zostanie wysłany, dopóki w buforze przyjmującym ten segment nie będzie wystarczająco dużo miejsca.
- ♦ *Wskaznik do pilnych danych* — 16-bitowe pole, zawierające położenie ostatniego bajta pilnych danych — o ile ustawiony został znacznik URG, oznaczający segment jako „pilny”.
- ♦ *Opcje* — pole opcji, mające zmienną długość. Pole to jest używane podczas negocjacji wartości MSS pomiędzy klientem i serwerem.

Rozdział 7.

Warstwa aplikacji

W tym rozdziale:

- ◆ Przegląd portów
- ◆ Wprowadzenie do gniazd (*sockets*)

Internet jest siecią niejednorodną, na którą składają się różnego typu komputery i systemy operacyjne — takie jak Macintosh, Windows, Unix, OS/2 i tak dalej. Platformy te stosują odmienne konwencje nazywania plików, reprezentowania tekstu, odmienne metody szyfrowania danych i typy terminali. Warstwa aplikacji, piąta i ostatnia w modelu TCP/IP, odpowiada za pomyslną łączność i zgodność pomiędzy tymi niejednorodnymi platformami. Funkcjonalność tej warstwy jest zapewniana przez szereg różnych protokołów, na przykład: SMTP na potrzeby poczty elektronicznej, Telnet dla wirtualnych terminali oraz FTP do przesyłania plików. Warstwa aplikacji zawiera także aplikacje (programy) zdefiniowane przez użytkownika.

Z niniejszego rozdziału Czytelnik dowie się, jak aplikacje działające w odrebnym węzle komunikują się ze sobą. Omówimy porty, gniazda i ich role w komunikacji równorzędnych aplikacji. Poznamy również popularne interfejsy API gniazd, takie jak gniazda *Berkeley* (BSD), *Transport Layer Interface* (TLI), *Transport Independent Remote Procedure Calls* (TI-RPCs) oraz *Windows Sockets* (WinSock). Na koniec opisześmy role zdalnych wywołań procedur (RPC — *Remote Procedure Call*) w sieciach rozproszonych globalnie.

Przegląd portów

W procesie komunikacji samo wysłanie komunikatu do odbiorcy nie wystarczy. Nadawca musi nie tylko zapewnić dostarczenie komunikatu do zamierzzonego odbiorcy, lecz również dostarczyć go do określonego — spośród wszystkich procesów odbiorcy — procesu, który zainicjował łączność po stronie nadawcy. Protokoly sieciowe umozliwiają taką łączność dwupunktową pomiędzy użytkownikiem i aplikacją, lub pomiędzy dwiema aplikacjami, za pomocą portów i gniazd.

Odbiorca komunikatu może być inny węzeł lub usługa działająca w tym samym węźle.



Uwaga

Port jest interfejsem pomiędzy aplikacją i siecią, w której ta aplikacja działa. Inaczej mówiąc, *port* stanowi punkt końcowy komunikacji, dzięki któremu aplikacja, użytkownik końcowy lub inny węzeł w sieci uzyskuje połączenie z aplikacją. W sieciach opartych na protokole TCP/IP każda aplikacja, która chce komunikować się z inną równorzędną aplikacją działającą w innym węźle, musi używać numeru portu.



Numery portów są podobne do wewnętrznych numerów telefonicznych. Aby połączyć się z głównym budynkiem biura, wybieramy numer do firmy, a następnie wewnętrzny, aby połączyć się z określona osoba w firmie. Adres IP możemy porównać do numeru biura. Każda usługa działająca w węźle posiada stały numer portu — „numer wewnętrzny” — pod którym możemy uzyskać dostęp do usługi.

Numer portu jest liczba szesnastobitowa z zakresu od 0 do 65535. Ogólnie, na potrzeby popularnych usług komunikacyjnych używane są numery portów poniżej 1024. Wartości powyżej 1024 zarezerwowane są tylko na potrzeby węzła. Większość portów serwerów stosuje numery poniżej 1024.; praktyka ta wywodzi się z wcześniejszej historii systemu Unix, który zezwalał na wiązanie z portami o numerach poniżej 1024. jedynie procesów użytkownika uprzywilejowanego (root).

Jesli port serwera jest już w użyciu, kolejnym zadaniom przypisywane są tymczasowe numery portów. Zalóżmy, że węzeł odbiera wiele zadań FTP. W tym przypadku tylko jeden węzeł zadający usługę możełączyć się z usługą FTP na porcie 21., który jest domyślnym portem dla FTP. Innym węzłem równoczesnie zadającym dostępu przydziela się numery portów powyżej 1024. W ten sposób wiele klientów może korzystać jednocześnie z usługi FTP. Programiści systemowi i sieciowi mogą też używać portów powyżej 1024. na potrzeby własnych aplikacji.



Większość systemów operacyjnych utrzymuje plik zawierający numery portów i odpowiadające im usługi. Jednakże wartości numery portów mogą ulegać zmianom, w zależności od platformy programowej i sprzętowej, na której uruchomione jest oprogramowanie TCP.

Większość aplikacji TCP/IP używa w komunikacji modelu klient-serwer. Po stronie użytkownika klient wysyła zadanie określonej usługi poprzez jej port w serwerze, po czym serwer odpowiada na zadanie. Gdy połączenie nawiązywane jest przez port przydzielony do określonego protokołu, wówczas odpowiadające mu usługi są wywoływane szybciej, dzięki czemu stosowanie numerów portów przyspiesza łączność TCP/IP.

Dobrze znane numery portów

Numery portów są zwykle dzielone na trzy kategorie:

- ◆ *Dobrze znane numery portów (well-known port numbers)* — ta nazwa odnosi się do portów od 0. do 1023. Organizacja IANA (*Internet Assigned Numbers Authority*) opublikowała listę często używanych numerów portów i odpowiadających im usług. Na przykład, FTP jest kojarzony z portem 21., Telnet z 23., SMTP (protokół poczty elektronicznej) z 25., serwery WWW z 80., a protokół POP3 z portem 110. Tabela 7.1 zawiera listę dobrze znanych numerów portów.

Tabela 7.1. Dobre znane porty TCP/IP i ich numery

Numer portu	Usluga	Opis
1	tcpmux	Multiplekser portów usług TCP
7	echo	Echo
11	systat	Active Users
13	daytime	Daytime (RFC 867)
17	qotd	Cytat dnia
18	msp	Message Send Protocol
19	chargen	Generator znaków
20	ftp-data	Transfer plików — dane
21	ftp	Transfer plików — sterowanie
22	ssh	Protokół logowania zdalnego SSH
23	telnet	Telnet
25	smtp	Simple Mail Transfer
31	msg-auth	Uwierzytelnianie MSG
37	time	Czas
41	graphics	Obsługa grafiki
42	name	Serwer nazw hostów
43	nickname	Who Is
49	tacacs	Login Host Protocol (TACACS)
53	domain	DNS
67	bootps	Bootstrap Protocol Server
68	bootpc	Bootstrap Protocol Client
69	tftp	Trivial File Transfer Protocol
70	gopher	Gopher
79	finger	Finger
80	http	World Wide Web HTTP
101	hostname	Serwer nazw hostów NIC
103	gppitnp	Genesis Point-to-Point Trans Net
107	rtelnet	Usluga Remote Telnet
109	pop2	Post Office Protocol — wersja 2
110	pop3	Post Office Protocol — wersja 3
111	sunrpc	SUN Remote Procedure Call

Tabela 7.1. Dobre znane porty TCP/IP i ich numery (ciag dalszy)

Numer portu	Usluga	Opis
119	nntp	Network News Transfer Protocol
123	ntp	Network Time Protocol
137	netbios-ns	Usluga nazewnicza NETBIOS
138	netbios-dgm	Usluga datagramów NETBIOS
139	netbios-ssn	Usluga sesji NETBIOS
143	imap	Internet Message Access Protocol
161	snmp	SNMP
162	snmptrap	SNMPTRAP
163	cmip-man	CMIP/TCP Manager
164	cmip-agent	CMIP/TCP Agent
165	xns-courier	Xerox
179	bgp	Border Gateway Protocol
194	irc	Internet Relay Chat Protocol
199	smux	SMUX
201	at-rtmp	AppleTalk — utrzymanie tras
202	at-nbp	AppleTalk — powiazanie nazw
209	qsmtp	Quick Mail Transfer Protocol
213	ipx	IPX
372	ulistproc	ListProcessor
444	snpp	Simple Network Paging Protocol
465	urd	URL Rendesvous Directory dla SSM
465	igmpv3lite	IGMP przez UDP dla SSM
487	saft	Simple Asynchronous File Transfer
512	exec	Zdalne wykonywanie procesów
513	login	Zdalny login à la Telnet
515	printer	kolejkowanie drukowania
517	talk	talk
531	conference	chat
533	netwall	Dla rozwloszen awaryjnych
765	webster	Slownik sieciowy
873	rsync	rsync
1080	socks	Socks



Numery portów przydzielone przez IANA są unikatowe. Inna usługa (lub protokół) *nie może* używać numeru portu przydzielonego do usługi lub do odpowiadającego jej protokołu.

- ◆ *Zarejestrowane numery portów* — ta nazwa dotyczy numerów portów od 1024. do 49151.
- ◆ *Prywatne numery portów* — *numery portów od 49152. do 65535. (czasami nazywane sa też dynamicznymi numerami portów)*.



Pełna lista numerów portów dostępna jest pod adresem www.iana.org/assignments/port-numbers. Dodatkowe informacje o portach można znaleźć w dokumentach RFC 1700 i 793.

Gniazda — wprowadzenie

Gniazdo (*socket*) jest mechanizmem komunikacji między procesami, służącym jako punkt końcowy połączenia. *Gniazdo* stanowi połączenie adresu IP z numerem portu TCP. Jest to podstawowy element łączności w sieciach TCP/IP, z których tworzony jest szkielet komunikacji pomiędzy procesami wykonywanymi w tym samym hostie lub w różnych węzłach sieci. Dzięki zastosowaniu gniazd każde połączenie klient-serwer jest unikatowe i nie trzeba zapisywać danych na dysku odbiorcy podczas każdej transakcji — zamiast tego dane składowane są tymczasowo w pamięci podręcznej bufora odbiorcy.



Ponieważ numer portu dla danej usługi oraz adres IP węzła są unikatowe, numery gniazd również są unikatowe.

Gniazda funkcjonują w obrębie *domeny komunikacyjnej* (lub po prostu *domeny*). Na tę domenę składają się struktura adresowania i zestaw protokołów implementujących różne typy gniazd. Gniazda zasadniczo tworzą trzy domeny — *domenę internetową*, *domenę uniksową* i *domenę NS*. Chociaż pakiet protokołów TCP/IP może obsługiwać wszystkie trzy domeny, najczęściej używana jest domena internetowa. W tej domenie gniazdo może należeć do jednego z dwóch typów: *gniazd potokowych* i *gniazd datagramów*. Ponieważ gniazda potokowe są oparte na protokole TCP, zapewniają transmisję danych zorientowaną na połączenie, dwukierunkowa, wiarygodna, sekwencyjna i nie powielana. W przeciwnieństwie do nich, gniazda datagramów oferują wymiane danych bezpołączniowa, dwukierunkowa, nie gwarantowana, która nie zapewnia porządkowania i może w niej występuwać powielanie.

Dwukierunkowa łączność oparta na gniazdach

Gdy węzeł wysyła w sieci TCP/IP zadanie połączenia z innym węzłem, wysyła również numer gniazda. Jeśli węzeł odbiorcy może nawiązać połączenie, zwraca numer gniazda zawierający adres IP odbiorcy i numer portu usługi, która będzie obsługiwać zadanie. Proces ten nosi nazwę *wiązania* (*binding*).



Analogia komunikacji opartej na gniazdach jest lacznosc telefoniczna. Podobnie jak telefon, gniazdo stanowi punkt koncowy dwukierunkowej lacznosci, gdy polaczmy dwa gniazda, pozwoli to przesyac dane pomiedzy dwoma procesami, ktore moga dzialac w roznych wezlach (komputerach).

Po nawiazaniu polaczenia miedzy dwoma punktami, a przed wymiana danych, oba punkty koncowe dokonuja wymiany swoich tozsamosci. Informacje te sa przechowywane po obu stronach, aby mozna bylo odwolac sie do nich w dowolnej chwili podczas transmisi. Zapobiega to generowaniu duzego, niepotrzebnego ruchu w sieci, który powstawał by podczas przesyłania z kazdym pakietem danych tozsamosci gniazda nadajacego.



Kazdy wezel moze zadac od wezla docelowego wiecej niz jednego polaczenia. W takim przypadku wezel-nadawca musi uzyc roznnych numerow portow do utworzenia odrebnich gniazdz. Dzieki temu proces zadajacy polaczenia nie musi czekac az odbiorca obsluzy wczesniejsze zadania.

Gdy aplikacja klienta laczy sie z usluga dostepna w serwerze, wówczas uzywa portu hosta klienta. Klient, aby uzyskac dostep do uslugi w serwerze, musi postapic zgodnie ze standardowym procesem „powiaz-sluchaj-polacz-zaakceptuj” (*bind-listen-connect-accept*). Proces ten przebiega wedlug nastepujacego scenariusza:

- 1.** Proces serwera wiaze sie z okreslonym portem.
- 2.** Po powiazaniu z portem proces serwera zaczyna nasluchiwanie na tym porcie zadan klientow.
- 3.** Proces klienta zadajacy od serwera uslugi wiaze sie z dostepnym portem w hospie-kliencie.
- 4.** Klient uzywa tego portu, aby wyslac zadanie polaczenia z serwerem przez odpowiadajacy usludze port serwera.
- 5.** Proces serwera akceptuje polaczenie i powiadamia klienta, by ten rozpoczal transakcje.

Opisany proces dwukierunkowej lacznosci jest przedstawiony na rysunku 7.1.

Gniazda sa w wysokim stopniu zalezne od systemu i programowalne. W interfejsie sieciowym gniazda implementowane sa jako *zlacza programowe aplikacji* (API — *Application Programming Interface*). Zlacza API stanowia lacze pomiedzy protokołami warstwy sieciowej i programami warstwy aplikacji na potrzeby funkcjonalnosci sieci. API pozwalaja również programistom systemów na wykorzystanie zasobów komputera: interfejsu graficznego, systemu plików, systemów baz danych i oczywiście systemu sieci komputerowej.



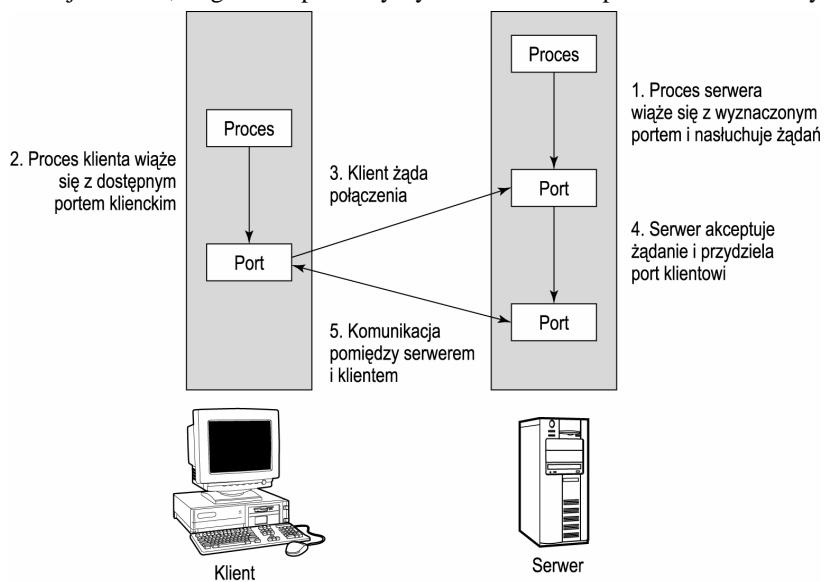
Gniazda sa implementowane na poziomie sprzętowym, wiec sa zalezne od systemu.

Chociaz gniazda sa zalezne od systemu, zlacza API gniazd powinny obslugiwac trzy podstawowe właściwości: *przezroczystosc dla protokołu warstwy sieciowej, prace asynchroniczna i sterowanie szybkoscia przesyłu danych*. Przezroczystosc dla protokołu

warstwy sieciowej oznacza, iż gniazda powinny być niezależne od protokołów warstwy

Rysunek 7.1.

Proces „powiąz-słuchaj-polacz-zaakceptuj”



sieciowej, z których korzystają. Praca asynchroniczna oznacza, że funkcje API powinny być wywoływanie przez zdarzenia, a nie sekwencyjnie. Ponadto API gniazd powinny zapewniać wystarczające szybkości przesyłu danych pomiędzy dwoma uczestnikami łączności, by nie powodować opóźnień. Do popularnych API gniazd należą:

- ◆ gniazda *Berkeley*,
- ◆ TLI (*Transport Layer Interface* — interfejs warstwy transportu),
- ◆ TI-RCPs (*Transport Independent Remote Procedure Calls* — niezależne od transportu zdalne wywołania procedur),
- ◆ WinSock (*Windows Sockets* — gniazda systemu Windows).

Gniazda i TLI oferują praktycznie taka sama funkcjonalność (dostęp do protokołów TCP i UDP) i wykluczają się wzajemnie, aczkolwiek programista systemowy może napisać obsługujący oba standardy program skompilowany warunkowo. Złącza API RPC obsługują sieciowe podprogramy standardowe za pomocą protokołu RPS firmy Sun. Systemy Microsoft Windows oferują podobne do gniazd złącza API o nazwie WinSock.



Dodatkowe informacje o gniazdach można znaleźć w dokumentach RFC 204, 1323 i 2292.

Gniazda Berkeley

Interfejs gniazd Berkeley został opracowany w University of California w Berkeley na potrzeby wersji 4.1c BSD (*Berkeley Software Distribution*). Był to prosty interfejs, używany początkowo w różnych wersjach systemu Unix: SCO, Linux i SunOS. Z czasem został też zawarty w Windows 9x, Windows NT, NetWare i innych systemach

operacyjnych. Gniazda Berkeley oferują umiarkowane szybkości przesyłu danych, ponieważ nie obsługują nakładających się funkcji wejścia-wyjścia. Dlatego interfejs sieciowy gniazd Berkeley najlepiej nadaje się do wieloprocesorowych sieciowych systemów operacyjnych.



Gniazda Berkeley są popularnie nazywane gniazdami BSD.

Gniazda Berkeley dają interfejs łatwy w użyciu; łatwo je też zaimplementować. Aplikacje, które korzystają z gniazd Berkeley mogąłączyć się z sieciowymi usługami transportowymi na dwa sposoby:

- ◆ *Dostęp zorientowany na połaczenie* — pomiędzy nadawcą i odbiorcą utrzymywany jest kanał wirtualny. W zależności od rezultatu transmisji, używane są potwierdzenia i potwierdzenia negatywne. W przypadku, gdy dane nie zostaną dostarczone do odbiorcy, inicjowana jest retransmisja danych albo wyższe warstwy zostają powiadomione o niepowodzeniu, po czym warstwy te mogą podjąć niezbędne czynności korekcyjne. Usługi zorientowane na połaczenie są obsługiwane przez TCP.
- ◆ *Dostęp bezpołaczeniowy* — dane wysyłane są do odbiorcy bez czekania na potwierdzenie. Jeśli dane (lub ich część) zostaną utracone w trakcie transmisji, nadawca nie będzie o tym wiedział. Usługi bezpołaczeniowe są realizowane przez protokół UDP.



Protokoły typu SMTP (*Simple Mail Transfer Protocol*) oparte są na TCP, natomiast protokoły takie jak Echo mogą korzystać zarówno z TCP, jak i UDP. Inne protokoły, jak np. SNMP (*Simple Network Management Protocol*), w pełni opierają się na UDP.

Z programistycznego punktu widzenia gniazda tworzone są za pomocą funkcji `socket()`, która wymaga dwóch kluczowych argumentów — *domeny* i *typu*. W sieciach TCP/IP najczęściej używana domena jest PF_NET (choćżej istnieją też inne). Dwa podstawowe typy gniazd internetowych to STREAM (potok, dla TCP) oraz DGRAM (datagram, dla UDP). Gniazda STREAM nie mogą wysyłać i odbierać danych bez nawiązania połączenia, zaś gniazda DGRAM mogą przesyłać dane natychmiast.

W komunikacji TCP zorientowanej na połaczenie nadawca używa funkcji `send()` do szczegółowania adresu IP i numeru portu odbiorcy do utworzenia aktywnego gniazda, co inicjalizuje połaczenie TCP. Odbiorca, czekając na połaczenie, używa funkcji `listen()` i `bind()`. Po wykryciu nadawanego zadania odbiorca używa funkcji `accept()`, aby zaakceptować połaczenie i utworzyć aktywne gniazdo.

W bezpołaczeniowej komunikacji UDP nadawca tworzy gniazda DGRAM i wysyła pakiety UDP za pomocą funkcji `sendto()`. Aby odebrać pakiet, adresat dodaje gniazdo do lokalnego numeru portu za pomocą funkcji `bind()`. Po dodaniu pakietu do portu UDP gniazdo może służyć do wysyłania pakietów (za pomocą funkcji `sendto()`) oraz ich odbierania (za pomocą `recvfrom()`).



Dodatkowe informacje o gniazdach Berkeley mozna znalezc w dokumentach RFC 793 i 1122.

Transport Layer Interface

API TLI (*Transport Layer Interface*) udostepnia niezalezny od protokolu interfejs dostepu do zasobow sieciowych z warstwy transportowej. TLI został wprowadzony przez Bell Labs pod koniec lat 80. w systemie AT&T UNIX System V Release 3 (UNI SVR3). Interfejs ten został opracowany na potrzeby rozproszonych aplikacji dzialajacych na różnych platformach. Do opracowania TLI w Bell Labs wykorzystano w roli modelu warstwe transportowa modelu OSI, dzieki czemu TLI jest w pełni zgodny z uslugami transportowymi OSI. Jest to tez główny powód, dla którego TLI deklaruje wyzsosc nad gniazdami (nie funkcjonujacymi w srodowisku OSI). Interfejs TLI moze obslugiwac TCP/ UDP, IPX/SPX i inne protokoly warstwy transportowej.

Chociaz TLI jest interfejsem API warstwy transportowej, udostepnia niemal identyczna funkcjonalnosc jak gniazda Berkeley i moze wspolpracowac z uslugami opartymi na gniazdach oraz IP. Jednakze w przeciwienstwie do gniazd Berkeley, uslugi oparte na TLI maja bezposredni dostep do danych wysylanych lub odbieranych w trakcie transmisi. Do komunikacji z polaczeniem sieciowym musza one uzywac serwerów baz danych lub plików, w wyniku czego interfejs TLI nie jest tak powszechnie akceptowany jak gniazda Berkeley.



TLI zaczyna powoli zyskiwac na popularnosci. Wiekszosc systemow operacyjnych, a szczegolnie Unix, obsluguje zarówno gniazda, jak i TLI. Wielu producentow preferuje jednak interfejs TLI z uwagi na szybkie, wiarygodne transakcje i zgodosc z protokolami OSI — na przyklad, w systemach Unix SRV4 i Solaris 2.x firmy Sun TLI jest preferowanym interfejsem transportowym. TLI byl zawsze interfejsem faworyzowanym we wszystkich wersjach Novell NetWare. Dodatkowe informacje o TLI mozna znalezc w RFC 1122.

Podczas transakcji TLI tworzy punkty koncowe transportu, którymi mozna łatwo manipulowac za pomoca funkcji analogicznych do funkcji gniazd. Interfejs TLI i gniazda Berkeley różnia sie jedynie skladnia.



Niedawno pojawiło się rozszerzenie TLI — *X/OPEN Transport Interface* (XTI). Interfejs XTI został opracowany w roku 1996 przez X/Open Company, Ltd. jako ulepszenie istniejącego interfejsu TLI. Poza obsługą tradycyjnych pakietów TCP/IP oraz IPX/SPX, interfejs XTI pozwala na dostęp do pakietów NetBIOS. Dodatkowe informacje o XTI mozna znalezc pod adresem:

www.tru64unix.compq.com/faqs/publications/base_doc/DOCUMENTATION/HTML/AAP2WD-TET1_html/netprog4.html.

Transport Independent Remote Procedure Call

Transport Independent Remote Procedure Call (TI-RPC) jest najnowszym dzielem w dziedzinie zdalnych wywolan procedur (RPC). TI-RPC umoziwi gladkie przejscie z jednego protokolu na drugi, oddzielajac protokol mieszczacy sie ponizej, w warstwie sieciowej. Zdolosc ta uniezaleznia specyfikacje RPC od transportu.

TI-RPC wprowadza warstwowa strukture RPC, w ktorej API RPC podzielone sa na rózne poziomy:

- ◆ *Poziom uproszczony (Simplified Level)* — wszystkie wywołania API polaczono sa w jedna procedure. Wprawdzie nie wolno dostosowywac klientow ani uslug, lecz mozna opracowac usluge RPC i odpowiadajaca jej aplikacje klienta.
- ◆ *Poziom najwyzszy (Top Level)* — klienci i uslugi moga byc łatwo modyfikowane w zależności od potrzeb. Parametry sa bardzo podobne do uzywanych w poziomie uproszczonym.
- ◆ *Poziom sredni (Intermediate Level)* — na tym poziomie zaczyna sie rozróżnienie pomiedzy warstwami, co pozwala na wyższy stopień modyfikacji i kontroli nad transportem.
- ◆ *Poziom ekspercki (Expert Level)* — najniższy poziom dostęnych API TI-RPC. Możliwości dostosowania klientów i usług sa na nim znacznie większe — dostępna jest kontrola nad transportem, rozmiarami buforów i innymi drobnymi szczegółami aplikacji.



Poziom ekspercki TI-RPC uzywa zlaczy API tlumaczenia nazw na adresy, które udostepniaja interfejs podobny do wywolan gniazd.

Mozna uzywac innych API, w polaczeniu z wszystkimi poziomami z wyjątkiem uproszczonego. Udostepniaja one metody zwracania błędów od uslugi do klienta, zwalniania przestrzeni pamieci przydzielonej do klientów i uslug oraz ulepszone metody wykrywania i zgłaszania błędów.



Dodatkowe informacje o TI-RPC mozna znalezc w dokumentach RFC 1057, 1058 i 2292.

WinSock

W srodowisku Windows uzywa sie *gniazd Windows (Windows Sockets — WinSock)*, bedacych odmiana gniazd Berkeley. Microsoft wprowadzil na rynek WinSock API (WSA) w styczniu 1993 r. jako interfejs do tworzenia w srodowisku Windows uniwersalnych aplikacji opartych na TCP/IP. Początkowo WinSock API koncentrowal sie jedynie na TCP/IP, chociaż mógł obsługiwac inne pakiety protokolów. Druga, ulepszona wersja WinSock (WinSock Version 2) została wydana w połowie 1995 r. Wersja ta obsługuje o wiele więcej pakietów protokolów — na przykład, IPX/SPX, ATM, DECnet i tak dalej. WinSock 2 zapewnia tez pełną zgodność wstecz z wcześniejszą wersją Win-Sock (1.1) i pozwala tworzyć aplikacje niezależne od protokołu sieciowego.



WinSock to biblioteka procedur, wywołan funkcji i struktur danych, która jest standardowym interfejsem dla aplikacji opartych na systemie Windows. Aplikacja WinSock 2 może wybierać protokoły w zależności od wymagań usług. Korzystając z mechanizmów udostępnionych przez WinSock 2, aplikacja może również dostosowywać się do różnic w sieciowych schematach nazw i adresowania.

Ponieważ WinSock opiera się na oryginalnych gniazdach Berkeley, łączność przez WinSock przypomina łączność przez gniazda Berkeley. Gniazda WinSock, podobnie jak Berkeley, tworzone są za pomocą funkcji `socket()`, która przyjmuje jako argumenty domenę i typ. W Internecie oraz sieciach opartych na TCP/IP najczęściej spotykana domena jest AF_NET. Argument typu może przybrać jedną z dwóch wartości — `SOCK_STREAM` (dla komunikacji opartej na TCP) lub `SOCK_DGRAM` (dla komunikacji opartej na UDP).



Dodatkowe informacje o WinSock można znaleźć w RFC 1122.

Ponieważ coraz więcej organizacji staje się obecnych na skali globalnej, zasięg sieci również się zwiększa. Obecnie sieć przedsiębiorstwa może nie ograniczać się do miasta czy nawet kraju. Sieci rozciągają się na cały świat. Tego typu sieci noszą nazwę *sistemów rozproszonych* lub *intranetów*. Ponadto, pojedynczy intranet nie musi w całości używać wspólnej platformy, typu komputery lub aplikacje. Platformy — zarówno sprzętowe, jak i programowe — mogą być tak różnorodne, jak tylko się da. Rozproszony charakter sieci mógł pojawić się w oprogramowania rozproszonego (na przykład sieciowych usług katalogowych i rozproszonych baz danych). Potrzeba funkcjonowania rozprozonego oprogramowania na różnorodnych procesorach i pod różnymi systemami operacyjnymi doprowadziła do opracowania *zdalnych wywołań procedur* (RPC — *Remote Procedure Call*).

RPC

Zdalne wywołania procedur (RPC), opracowane w CERN (*Center for Nuclear Research*), są metodą budowania rozproszonych aplikacji i systemów opartych na modelu klient-serwer. RPC pozwala aplikacji uruchomionej w jednym komputerze wywołać podprogram standardowy, który może wykonywać się w odległym komputerze. Wywołująca go aplikacja nawet nie „wie”, iż podprogram jest zdalny. Inaczej mówiąc, RPC jest metodą korzystania w sposób przezroczysty z istniejących środków komunikacji.



Podprogram standardowy jest częścią programu, wykonującej określone zadanie lub uporządkowany zbiór zadań.

RPC nie zawiera żadnego kodu związanego z komunikacją, w wyniku czego jest niezależne od:

- ◆ platform i sprzętu komunikacyjnego,
- ◆ protokołów komunikacyjnych,
- ◆ systemów operacyjnych,
- ◆ sekwencji wywołań, które musiałyby korzystać z oprogramowania komunikacyjnego.

Niezależność od interfejsu izoluje rozprozone aplikacje oparte na RPC od fizycznych i logicznych aspektów przesyłu danych i pozwala aplikacjom korzystać z różnych modeli transportu danych. Pozwala też programistom tworzącym aplikacje rozprozone ignorować szczegóły interfejsu w trakcie pisania programu. Dzięki tym cechom RPC

model srodowisk komputerowych klient-serwer staje sie bardziej efektywny i łatwiejszy do oprogramowania.

Idea RPC została pomyslnie wdrozona w wielu typach aplikacji. Aplikacjami, które docz wczesnie zaczęły korzystać z RPC, były zdalne serwery plików i baz danych. Sun Network File System firmy Sun Microsystems używa RPC Sun XDR. Aplikacje zdalnego monitorowania, na przykład GKS, oraz aplikacje zdalnego zarządzania zadaniami programowymi używane w komputerach VAX również korzystają z RPC.

Gdy zostaje wygenerowane zdalne wywołanie podprogramu standardowego, program wywołujący nazywany jest klientem, zaś wywołany podprogram gra role serwera tworzącego moduły namiastkowe. Klient i serwer potrzebują nazw procedur, zaangażowanych w transakcje, liczb parametrów, które zostaną przekazane, oraz typu danych każdego parametru. Gdy serwer jest wywoływany przez klienta, RPC sprawia, że:

- ◆ wszystkie parametry przeznaczone do przekazania do serwera zostają przesłane do odległego komputera, w którym wykonyuje się podprogram standardowy,
- ◆ wykonuje się podprogram w odległym komputerze,
- ◆ wyniki i parametry będące rezultatem wykonania podprogramu zostają przekazane z powrotem do klienta (wywołującego program).

RPC do łączności pomiędzy serwerem i klientem stosuje *moduły namiastkowe (stub module)*. Namiastka jest podprogramem, który przypomina zdalne podprogramy standardowe. Namiastki nie zawierają żadnych informacji związanych z fizycznymi adresami komputerów zaangażowanych w transakcję, wobec czego w celu znalezienia komputera docelowego używa systemu wykonawczego (RTS) RPC. Ponieważ system wykonawczy RPC zajmuje się całą komunikacją, moduł namiastkowy zawiera jedynie kod związany z aplikacją, która zainicjowała zdalne wywołanie. Cały proces komunikacji przebiega następująco:

1. Program klienta zostaje powiązany z modelem namiastkowym klienta, który jest podprogramem przyjmującym dane z procesu wywołującego i zamkającym je w komunikacie. Ten proces nosi nazwę *zestawiania (marshalling)*.
2. Moduł namiastkowy klienta wysyła komunikat do serwera (procesu) za pomocą procedury w systemie wykonawczym RPC i wchodzi natychmiast w tryb oczekiwania; czeka na komunikat odpowiedzi od modułu namiastkowego serwera, znajdującego się w odległym komputerze.
3. System wykonawczy RPC powiadamia moduł namiastkowy serwera o otrzymaniu komunikatu od klienta. Moduł namiastkowy serwera *rozmontowuje (unmarshall)* parametry otrzymane w komunikacie, wywołuje docelowy podprogram standardowy i czeka na wyniki od serwera (procesu).
4. Serwer po ukończeniu wykonywania podprogramu zwraca parametry wynikowe do modułu namiastkowego serwera. Ten z kolei zestawia zwrócone parametry w komunikat i wysyła go do modułu namiastkowego klienta.
5. Po odebraniu odpowiedzi moduł namiastkowy serwera rozmontowuje zwrócone parametry, wywołuje proces klienta jako zwykłą procedure i zamienia wartości na zmienne programu wywołującego.

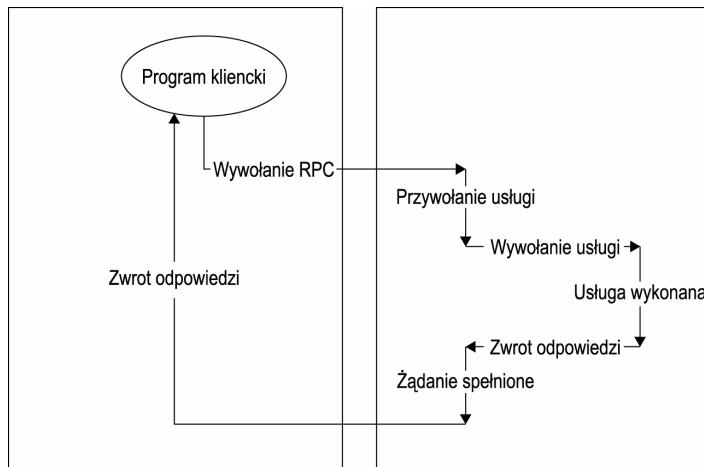
Po stronie klienta program wywolujacy pozostaje uspiony, dopóki nie otrzyma wyniku od wywolanego podprogramu standardowego. Po otrzymaniu oczekiwanych parametrów proces klienta podejmuje na nowo wykonanie. Natomiast po wysłaniu wyników w stan uspienia wchodzi podprogram standardowy. Rysunek 7.2 przedstawia cały proces komunikacji.



Chociaz proces klienta pozostaje uspiony w trakcie oczekiwania na parametry od serwera, sam klient nie jest uspiony i moze wykonywac inne zadania w trakcie oczekiwania. Dzieki temu wywolania RPC sa asynchroniczne.

Rysunek 7.2.

Lacznosc sieciowa za pomoca RPC



Do poznania wszystkich uslug RPC zarejestrowanych w określonym hostie i ich adresów mozemy uzyc polecenia powloki `rpcinfo`. Polecenie to moze tez posłuzyc do ustalenia biezacych informacji o rejestracji RPC. Administratorzy moga dzieki tym informacjom usuwac wszelkie nadmiarowe, przestarzale lub bezuzyteczne uslugi i rejestracje. Polecenia `rpcinfo` mozemy również uzyc, aby „pingowac” programy uruchomione w komputerze i ustalic, czy odpowiedz zostala otrzymana, czy nie, co z kolei pomaga w ustaleniu, czy odlegly komputer sie zawiesil. Wyniki polecenia `rpcinfo` przedstawia rysunek 7.3.

Rysunek 7.3.

Okno polecenia `rpcinfo`

```

root@localhost.localdomain: /root
File Edit Settings Help
[root@localhost /root]# rpcinfo -p
program vers proto port
 100000  2   tcp   111  portmapper
 100000  2   udp   111  portmapper
 100024  1   udp   1024  status
 100024  1   tcp   1024  status
[root@localhost /root]#
  
```

The screenshot shows a terminal window with a blue title bar containing the text "root@localhost.localdomain: /root". Below the title bar is a menu bar with "File", "Edit", "Settings", and "Help" options. The main area of the terminal displays the output of the command `rpcinfo -p`. The output lists several RPC programs and their ports:

program	vers	proto	port	service
100000	2	tcp	111	portmapper
100000	2	udp	111	portmapper
100024	1	udp	1024	status
100024	1	tcp	1024	status

RPC wykorzystuje komunikacje pomiędzy najwyższą warstwą modelu OSI — warstwa aplikacji a niższymi warstwami zajmującymi się rozproszona natura ogólnoswiatowego intranetu. RPC pełni podobną funkcję w stosunku do wyższych warstw, jak usługi transportowe dla niższych. Obecnie trwają prace nad uczynieniem standardu z RPC.



Dodatkowe informacje o RPC można znaleźć w dokumentach RFC 1050, 1057, 1700 i 1831.

Czesc II

Praca z TCP/IP

W tej czesci:

- ◆ Rozdzial 8. Instalacja i konfiguracja TCP/IP
- ◆ Rozdzial 9. Konfiguracja automatyczna
- ◆ Rozdzial 10. Znajdowanie hostow w sieci IP

Teraz, gdy Czytelnik poznal podstawy teoretyczne stosu TCP/IP, nadeszla pora na zainstalowanie i eksploatacje protokolu. Czesc II ksiazki zajmuje sie najwazniejszymi obszarami pracy z TCP/IP, do których zalicza sie instalowanie protokolu na platformach Windows i Unix oraz konfiguracje zainstalowanego protokolu. Omówiona zostala tutaj prosta reczna konfiguracja protokolu TCP/IP oraz konfiguracja automatyczna korzystajaca z uslug BOOTP i DHCP.

Rozdzial 10. dotyczy mechanizmu rozwiazywania nazw, który pozwala znajdowac hosty w Internecie i intranetach za pomoca latwych do zapamietania nazw zamiast adresów IP. Jednym z najczesciej spotykanych problemów z TCP/IP sa bledy w rozwiazywaniu nazw, wobec czego dobra znajomosc tego zagadnienia bedzie wzarna dla lektury pozostalych rozdzialow.

Rozdział 8.

Instalacja

i konfiguracja TCP/IP

W tym rozdziale:

- ◆ Konfiguracja TCP/IP w środowisku linuksowym
- ◆ Instalacja i konfiguracja TCP/IP w świecie Microsoftu

Poprzednie rozdziały omawiały warstwy protokołu TCP/IP i jego procesy komunikacyjne. Do implementacji TCP/IP w sieciach wymagana jest umiejętność instalowania i konfigurowania protokołu. Choć podstawowe podejście do tych zadań w różnych systemach operacyjnych jest takie samo, istnieją pewne różnice. Niniejszy rozdział zajmuje się instalacją i konfiguracją TCP/IP w środowiskach linuksowych i Microsoftu.

Konfiguracja TCP/IP

W różnych systemach operacyjnych stosowane jest jednakowe podstawowe podejście do konfiguracji TCP/IP. Wobec tego, przed rozpoczęciem konfiguracji TCP/IP w dowolnym systemie operacyjnym, musimy uzyskać niezbędne informacje o wszystkich komputerach w sieci. Wszystkie te informacje nie muszą być absolutnie niezbędne w czasie konfiguracji systemu — wiele będzie ustalanych automatycznie. W tym rozdziale informacje zostały podzielone na dwie kategorie: „informacje potrzebne zawsze” oraz „informacje potrzebne czasami”.

Informacje potrzebne zawsze

Niezależnie od systemu operacyjnego, w którym konfigurujemy TCP/IP, zawsze potrzebne będą następujące informacje:

- ◆ Nazwa komputera lub nazwa hosta, będąca symboliczną nazwą komputera w sieci. Nazwy hostów mogą być przydzielane jako przydomki (*nickname*) oraz pełne, złożone nazwy domen (FQDN — *Fully Qualified Domain Name*). Przydomki są aliasami adresów IP i mogą być przydzielane oraz używane przez poszczególnych użytkowników. Z drugiej strony, nazwy FQDN typu *serwer1.domena1.com* są hierarchiczne. Nazwy hostów muszą być unikatowe w obrębie lokalnej sieci, jednakże w różnych sieciach mogą znajdują się hosty o tej samej nazwie.

Nazwy hostów sa bardziej „przyjazne” niz adresy IP, wobec czego moga byc uzywane zamiast adresów IP.



Rozdzial 10. zawiera dodatkowe informacje o nazwach hostów.

- ♦ Sterownik urzadzenia — sterownik karty interfejsu sieciowego. Sterowniki urzadzen informuja system operacyjny, jak ma komunikowac sie z interfejsem sieciowym.



Najnowsze sterowniki urzadzen mozna otrzymac od producenta urzadzenia. Nawet jesli system operacyjny zawiera określony sterownik urzadzenia, powinnismy uzyskac go od producenta — co zapewni, iż bedziemy dysponowac najnowsza wersja.

- ♦ Dane konfiguracyjne karty sieciowej, które identyfikują format ramki adaptera sieciowego. Jesli adapter jest karta sieci token ring, wówczas format ramki to 802.5 lub 802.2; w przeciwnym razie Ethernet II.
- ♦ Adres IP — 32-bitowy numer, stanowiacy pelny adres komputera w sieci. Adres IP musi byc unikatowy i moze zostac przydzielony przez administratora systemu.



Dodatkowe informacje o adresach IP mozna znalezc w rozdziale 5.

- ♦ Maska sieci reprezentujaca adres IP pozbawiony identyfikatora sieci, przez co pozostaje tylko ID hosta. Maska sieci jest ciągiem bitów, sluzacym do „zamaskowania” określonej części adresu IP. Na przykład, w standardowym adresie maski sieci klasy C (255.255.255.0) ostatni oktet (0) oznacza „tutaj mieści się nazwa komputera”; pozostała część adresu jest numerem sieci. Maska sieci w sposób unikatowy identyfikuje sieć lokalną i uzywana jest przede wszystkim do podziału na podsieci.

- ♦ Adres rozgłoszeniowy, reprezentujacy adres IP złożony z samych jedynek (jak np. 255.255.255.255). Adres rozgłoszeniowy jest niezbedny, gdy trzeba rozgłosic komunikat do wszystkich komputerów w sieci, poniewaz karta sieciowa ignoruje wszystkie pakiety, które nie zawieraja jej określonego adresu IP.



Jesli w komputerze skonfigurowany jest protokół DHCP, a w sieci dostepny jest serwer DHCP, serwer ten automatycznie przydziela adres IP, maske sieci i adres rozgłoszeniowy.

Informacje potrzebne czasami

Od czasu do czasu, w trakcie konfigurowania TCP/IP system moze wymagac nastepujacych danych:

- ♦ Nazwa domeny, identyfikujaca cala sieć. Nazwa domeny jest niezbedna, gdy sieć musi laczyc sie z komputerami z zewnatrz. Administrator systemu moze udostepnic nazwe domeny, jednakże gdy sieć laczy sie z Internetem, nazwa ta musi zostac zaakceptowana przez organizacje InterNIC (*Internet Network Information Center*).

- ♦ Maksymalna liczba połączeń TCP/IP dopuszczalnych równoczesnie.
- ♦ Status bramy, który wskazuje, czy komputer pracuje w trybie bramowym. *Tryb bramowy (gateway mode)* oznacza, iż pakiety są przesyłane i przekierowywane pomiędzy różnymi sieciami. W skonfigurowanym interfejsie sieciowym (adapterze lub łączu szeregowym) domyślnie tryb ten jest wyłączony. Jeśli jednak skonfigurujemy drugi interfejs sieciowy, to system zaząda wyboru konkretnego trybu.
- ♦ Adres serwera nazw, który tłumaczy nazwy hostów na adresy IP. Hosty TCP/IP używają do komunikacji adresów IP, natomiast użytkownicy dla wygody stosują nazwy hostów. Z tego powodu, aby umożliwić komunikacje pomiędzy hostami, ich nazwy muszą zostać przetłumaczone na odpowiednie adresy IP. Proces ten określany jest mianem *rozwiązywania nazw (name resolution)*. Jeśli używamy tylko trybu petli zwrotnej (*loopback mode*), serwer nazw nie jest potrzebny. Skonfigurowanie TCP/IP w trybie petli zwrotnej pozwala na połączenia TCP/IP tylko z lokalnym komputerem.



Rozdział 10. zawiera dodatkowe informacje o rozwiązywaniu nazw.

Jak już wspomniano, w prawdzie w różnych systemach operacyjnych mogą wystąpić pewne różnice, lecz podczas konfigurowania TCP/IP musimy trzymać się określonych podstawowych kroków:

1. Uaktynić TCP/IP przez powiązanie z jadrem systemu operacyjnego lub załadowanie podczas uruchomienia komputera.
2. Podać nazwy wszystkich komputerów w sieci na potrzeby rozwiązywania nazw.
3. Utworzyć tablice tras, służące do sterowania jak i któreś pakiety będą przesyłane od źródła do miejsca przeznaczenia. Tablice tras identyfikują ponadto trasy przesyłania pakietów.
4. Skonfigurować serwer nazw domen, jeśli używany jest system rozproszonej bazy danych, taki jak BIND (*Berkeley Internet Name Daemon*). W systemie rozproszonej bazy danych serwer nazw domen pozwala klientom nazywać zasoby i obiekty oraz udostępniać te informacje innym obiektom w sieci.
5. Wyregulować parametry komputera, aby zoptymalizować wydajność.

Konfiguracja TCP/IP w świecie Linuksa

Konfigurowanie TCP/IP w środowisku uniwersyteckim wymaga modyfikacji zawartości kilku plików przez dodanie niezbędnych danych konfiguracyjnych. W różnych typach serwerów uniwersyteckich (na przykład SCO UNIX, BSD UNIX lub Linux) nazwy plików mogą się różnić, jednakże dane konfiguracyjne zapisane w tych plikach wyglądają identycznie w różnych systemach operacyjnych. Pliki konfiguracyjne, które należy zmodyfikować podczas konfigurowania TCP/IP w systemie Linux to:

- ♦ */etc/hosts* — zawiera listę nazw i adresów sieciowych wszystkich komputerów w sieci. Po dodaniu dowolnego komputera do sieci, w pliku HOSTS możemy dodać dla niego wpis na wypadek sytuacji, gdyby serwer nazw nie był

skonfigurowany i plik HOSTS sluzyl do rozwiazywania nazw. Ponizej przedstawione zostaly przykładowe wiersze z pliku HOSTS:

```
127.0.0.1 localhost.localdomain localhost  
172.17.55.51 server1.mydomin.com server1
```

- ♦ */etc/networks* — zawiera liste nazw i adresów sieci. Nazwy sieci wymagane sa jedynie wtedy, gdy uzytkownicy sieci lokalnej chca laczyc sie z innymi sieciami. Wykorzystanie tego pliku jest opcjonalne, w zaleznosci od potrzeb uzytkownika. Nazwe domeny i adres sieci mozna dodac do pliku jak w ponizszym przykladzie:

```
xserver. xdomain.com 145.205.15.1
```

- ♦ */etc/services* — zawiera informacje o wszystkich uslugach TCP i UDP udostepnianych przez system. Ponizej przedstawiono przykładowe uslugi z tego pliku:

```
echo 7/tcp  
tftp 69/udp
```

- ♦ */etc/protocols* — zawiera liste wszystkich protokolow transportowych i odpowiadajacych im numerow protokolow. Plik ten jest automatycznie aktualizowany w trakcie instalacji oprogramowania TCP/IP i zawiera nazwe protokolu, numer protokolu i dowolny alias protokolu. Ponizszy listing to przyklad z tego pliku:

```
ip 0 IP  
tcp 6 TCP  
udp 17 UDP
```

- ♦ */etc/hosts.equiv* — zawiera liste nazw komputerow i sluzy do sterowania dostepem z innych komputerow. Komputery, których nazwy wymienione sa w tym pliku, nosza nazwe *zaufanych hostow (trusted host)*. Kazdy poprawny uzytkownik (z wyjątkiem uzytkownika uprzywilejowanego root) ma prawo logowac sie zdalnie do komputera z tymi samymi danymi konta bez koniecznosci podawania hasla. Nosi to nazwe *równowaznosci uzytkownika*. W pliku *hosts.equiv* wpis dla uzytkownika z prawem dostepu reprezentowany jest przez znak +, zas znak - wskazuje, iz uzytkownik nie ma prawa dostepu. Zawartosc tego pliku ma nastepujacy format:

```
[+|-] [nazwa_hosta] [nazwa_uzytkownika]
```

- ♦ */etc/ftpusers* — zawiera liste uzytkownikow, którym nie wolno korzystac z dostepu do komputera poprzez usluge FTP. Za kazdym razem, gdy jeden z nieautoryzowanych uzytkownikow bedzie probował sie zalogowac, polaczenie zostanie natychmiast zerwane. Lista uzytkownikow FTP zawarta w tym pliku moze wygladac tak:

```
root  
mail  
news
```

- ♦ */etc/inetd.conf* — zawiera liste wszystkich procesow uruchamianych przez demona *inetd* podczas startu systemu. Procesy dzialajace stale w tle nosza nazwe procesow uslugowych lub demonow (*daemon process*). Demon *inetd* swiadczy wewnetrznie określone uslugi internetowe i uruchamia inne demony tylko w razie potrzeby, co zmniejsza obciążenie systemu. Ponizszy listing zawiera próbke tego pliku:

```

log_on_success = HOST PID
log_on_failure = HOST RECORD

♦ /etc/sysconfig/network-scripts/ifcfg-[nazwa_interfejsu] — zawiera informacje
o urzadzeniu. Urzadzenie reprezentowane jest przez nazwe interfejsu — na
przyklad, jesli urzadzenie nosi nazwe eth0, nazwa pliku brzmi ifcfg-eth0.
W zaleznosci od typu interfejsu, zawartosc tego pliku moze byc różna.
Przykładowe wiersze tego pliku moga wygladac nastepujaco:

```

```

DEVICE="eth0"
ONBOOT="yes"

```

♦ /etc/sysconfig/network — zawiera informacje o pozadanej konfiguracji sieci
w serwerze.

```

NETWORKING="yes"
HOSTNAME=serwer1.mojadomena.com
GATEWAY="172.17.55.1"
GATEWAYDEV=""
FORWARD_IPV4="yes"

```

Do parametrów konfiguracji sieci naleza ponizsze:

- ♦ *NETWORKING* — wartosc "yes" oznacza zalaczenie uslug sieciowych, "no" — wylaczenie.
- ♦ *HOSTNAME* — podaje nazwe hosta danego komputera.
- ♦ *GATEWAY* — adres IP odleglej Bramy, jesli jest dostepna.
- ♦ *GATEWAYDEV* — nazwa urzadzenia umozliwiajacego dostep do odleglej Bramy.
- ♦ *FORWARD_IPV4* — przyjmuje wartosc "yes" lub "no" w zaleznosci od tego,
czy pakiety IP maja byc przekazywane dalej.

Pozostala czesc tego podrozdzialu zajmuje sie konfiguracja TCP/IP w systemie Linux. Prosze jednak pamietac, iz podstawowe procedury konfiguracji TCP/IP sa takie same we wszystkich systemach uniksowych. Linux również jest dostepny w różnych wersjach różnych producentów, na przykład SlackWare i RedHat. Niniejszy podrozdzial przedstawia ogólna procedure konfiguracji dla systemów operacyjnych Linux.

Przed skonfigurowaniem TCP/IP w systemie Linux nalezy utworzyc i zamontowac system plików /proc. System plików oznacza metode organizacji plików na nosniku, na przykład na dyskietce lub dysku twardym. Montowanie systemu plików obejmuje wyszczególnienie urzadzenia zawierajacego system plików, typu urzadzenia i miejsca w hierarchii katalogów, gdzie nalezy zamontowac system plików. Linux daje wybór systemu plików z szeregu dostepnych typów, na przykład ext2. Poza tymi systemami plików do jadra systemu operacyjnego wbudowane sa specjalne systemy plików — jak na przykład /proc. Jadro korzysta z systemu plików /proc, aby uzyskac informacje o sieci.

W wiekszosci wersji Linuksa system plików /proc jest tworzony automatycznie podczas instalacji systemu operacyjnego. Jednak w niektórych przypadkach trzeba zmodyfikowac plik /etc/fstab przez dodanie nastepujacej dyrektywy:

```
none /proc proc defaults
```

Dyrektyna ta sluzi do wymuszenia automatycznego zamontowania systemu plików */proc*. Po upewnieniu sie, ze ten system plików zostal utworzony, musimy wybrac nazwe hosta (przydomek) dla linuksowego komputera. Do tego celu mozemy uzyc polecenia *hostname*:

```
hostname nazwa
```

W tym poleceniu *nazwa* oznacza nazwe systemu dla naszego komputera. Mozemy rowniez podac dla komputera nazwe FQDN. Na przyklad, ponizsze polecenie ustawia dla komputera nazwe *serwer1* w domenie *domena1*:

```
hostname serwer1.domena1.com
```

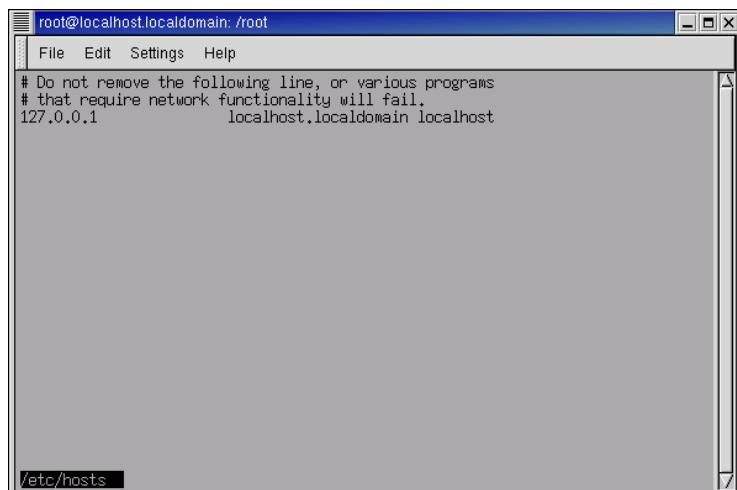
Mozemy rowniez wprowadzic wpis nazwy hosta do pliku *etc/hosts*. Mozemy go otwarcie i sprawdzic, czy zawiera nazwe naszego komputera. Do otwarcia pliku z poziomu katalogu glownego posluzy nam nastepujace polecenie:

```
vi /etc/hosts
```

Polecenie to otwiera plik w edytorze tekstu o nazwie *vi*. Przykładowy wynik został przedstawiony na rysunku 8.1. Aby wyjsc z edytora, należy nacisnąć *Esc*, wpisać *:q* i naciśnąć *Enter*.

Rysunek 8.1.

Przykładowy
plik HOSTS



The screenshot shows a terminal window titled "root@localhost.localdomain: /root". The window contains the following text:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      localhost.localdomain localhost
```

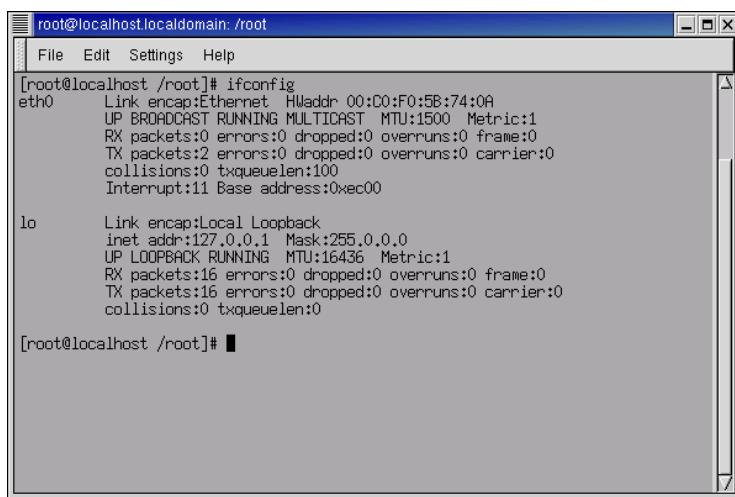
The window has a standard Linux terminal interface with a menu bar (File, Edit, Settings, Help), scroll bars, and a status bar at the bottom displaying "/etc/hosts".

Nastepnym krokiem jest konfiguracja interfejsu sieciowego za pomoca polecenia *ifconfig*. Polecenie to wywoluje wspolprace warstwy sieciowej jadra systemu operacyjnego z interfejsem sieciowym przez nadanie mu adresu IP. Po skonfigurowaniu i aktywacji interfejsu jadro moze przezen wysylac i odbierac dane. Polecenie *ifconfig* moze rowniez sluzyc do konfigurowania kilku interfejsow, na przyklad sterownika petli zwrotnej lub sterownika interfejsu Ethernet. Aby wyswietlic stan obecnie aktywnych interfejsow, jak na rysunku 8.2, wystarczy wydac polecenie *ifconfig* bez zadnych parametrów:

```
ifconfig
```

Rysunek 8.2.

Przykładowy wynik polecenia ifconfig



The screenshot shows a terminal window titled "root@localhost.localdomain: /root". The window contains the output of the "ifconfig" command. It lists two interfaces: "eth0" and "lo". The "eth0" interface is an Ethernet interface with the following details:

- Link encap:Ethernet HWaddr 00:0C:F0:5B:74:0A
- UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
- RX packets:0 errors:0 dropped:0 overruns:0 frame:0
- TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
- collisions:0 txqueuelen:100
- Interrupt:11 Base address:0x0ec00

The "lo" interface is a Local Loopback interface with the following details:

- Link encap:Local Loopback
- inet addr:127.0.0.1 Mask:255.0.0.0
- UP LOOPBACK RUNNING MTU:16436 Metric:1
- RX packets:16 errors:0 dropped:0 overruns:0 frame:0
- TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
- collisions:0 txqueuelen:0

Stan określonego interfejsu można wyświetlić za pomocą polecenia:

```
ifconfig nazwa_interfejsu
```

W powyższym poleceniu opcja *nazwa_interfejsu* jest nazwa interfejsu, którego stan chcemy wyświetlić i zwykle składa się z nazwy sterownika i numeru. Na przykład, *eth0* jest nazwa interfejsu sieci Ethernet. Aby wyświetlić stan wszystkich interfejsów, aktywnych i nieaktywnych, należy wpisać polecenie:

```
ifconfig -a
```

Ogólna składnia polecenia ifconfig jest następująca:

```
ifconfig interfejs opcja | adres
```

Do opcji używanych z poleceniem ifconfig należą:

- ♦ *up* — aktywuje interfejs,
- ♦ *down* — powoduje wyłączenie interfejsu,
- ♦ *netmask addr* — podaje maskę sieci IP dla interfejsu,
- ♦ *irq addr* — ustawia numer przerwania używany przez interfejs; polecenie niezbedne, gdy urządzenie nie jest w stanie zmienić IRQ dynamicznie,
- ♦ *mem_start address* — ustawia adres początkowej pamięci wspólnej, używanej przez interfejs,
- ♦ *address* — ustawia adres IP interfejsu.

Po aktywacji interfejsów sieciowych należy dodać lub usunąć trasy z tablicy tras jadra, aby umożliwić komputerowi znajdowanie innych komputerów w sieci. Do modyfikacji tablicy tras służy polecenie *route*. Oto składnia polecenia *route*, używana w celu dodania lub usunięcia trasy:

```
route add|del adres_IP
```

Opcja `add` sluzi do dodawania tras, zas opcja `del` do ich usuwania z tablicy tras jadra. Aby wyswietlic tablice tras jadra, wystarczy uzyc polecenia `route` bez zadnych opcji. Rysunek 8.3 przedstawia przykładowa tablice tras.

Rysunek 8.3.

Przykładowy wynik
polecenia route

```
[root@localhost /root]# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
127.0.0.0      *              255.0.0.0     U      0      0        0 lo
default         *              0.0.0.0       U      0      0        0 eth0
[root@localhost /root]#
```



Informacje o trasowaniu zawiera rozdział 5.

Za pomocą opcji `-n` można zmusić system operacyjny do wyświetlenia w tablicy tras jedynie adresów IP:

```
route -n
```

Po dodaniu tras do tablicy należy zidentyfikować ścieżkę, która pakiety będą przesyłane pomiędzy hostem źródłowym i docelowym. Do tego celu służy polecenie `traceroute`:

```
traceroute adres_docelowy
```

W tym poleceniu `adres_docelowy` oznacza adres IP lub nazwę hosta docelowego.

Konfiguracja TCP/IP na platformie linuksowej obejmuje konfiguracje kilku interfejsów. W pierwszej kolejności należy zainstalować interfejs petli zwrotniej (`loopback`), a następnie sterownik Ethernet dla sieci.

Sterownik petli zwrotnej jest zazwyczaj instalowany podczas instalacji systemu operacyjnego. Adres IP tego interfejsu jest niezmieniony — 127.0.0.1. Aby sprawdzić, czy interfejs ten istnieje w komputerze, możemy zajrzeć do pliku `/etc/hosts`, który w przypadku jego istnienia powinien zawierać następujący wiersz:

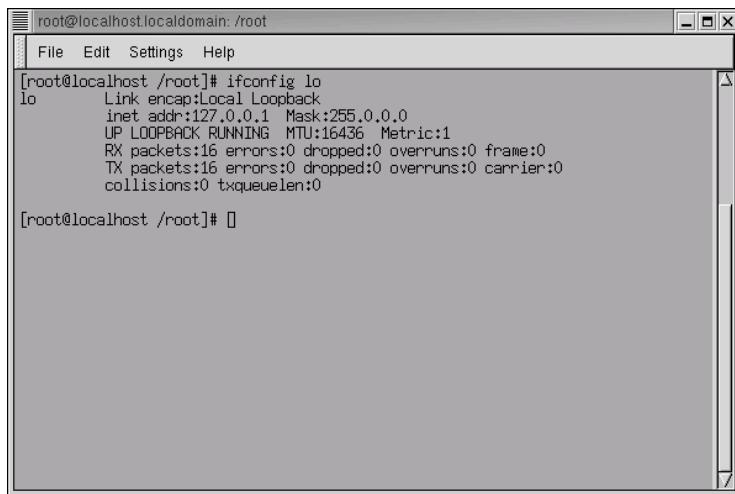
```
localhost 127.0.0.1
```

Mozemy też uzyc do tego celu polecenia `ifconfig`. Rysunek 8.4 przedstawia przykładowy wynik polecenia.

```
ifconfig lo
```

Rysunek 8.4.

Przykładowy wynik polecenia `ifconfig lo`



The screenshot shows a terminal window titled "root@localhost.localdomain: /root". The window contains the following text:

```
[root@localhost /root]# ifconfig lo
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
              UP LOOPBACK RUNNING MTU:16436 Metric:1
              RX packets:16 errors:0 dropped:0 overruns:0 frame:0
              TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              [root@localhost /root]#
```

Jesli po wydaniu tego polecenia zobaczymy komunikat o bledzie, oznacza to, ze sterownik petli zwrotnej nie istnieje. Aby utworzyc ten interfejs, nalezy uzyc polecenia `ifconfig`:

```
ifconfig lo 127.0.0.1
```

Polecenie to tworzy wpis dla sterownika petli zwrotnej w pliku `/etc/hosts`.

Po utworzeniu tego interfejsu, nalezy dodac go do tablicy tras jadra za pomoca polecenia `route`:

```
route add 127.0.0.1
```

albo:

```
route add localhost
```

Nastepnie nalezy sprawdzic trasowanie polecamiem `ping`, ktore wysyla okreslone pakietы danych do hosta sieciowego i sluzy do sprawdzania jego reakcji. Aby sprawdzic odpowiedz lokalnego hosta, mozna uzyc polecenia:

```
ping 127.0.0.1
```

albo:

```
ping localhost
```

Oba powyzsze polecenia dadza ten sam wynik. Rysunek 8.5 przedstawia przykładowy wynik polecenia `ping localhost`. Jesli polecenie to nie da odpowiedzi, trzeba ponownie sprawdzic pliki konfiguracyjne i wpisy tras, poniewaz moglo sie zdarzyc, ze adres i nazwa interfejsu petli zwrotnej nie zostaly rozpoznane. Jesli jednak pliki konfiguracyjne itablica tras zawieraja właściwe wpisy, problem moze byc powazniejszy. Wersje jadra inarzedzi sieciowych moga sie nie zgadzac, jadro sieciowe moze nie byc odpowiednio skonfigurowane, wobec czego caly proces trzeba bedzie powtorzyc.

Rysunek 8.5.

Przykładowy wynik polecenia ping localhost

```
[root@localhost /root]# ping localhost
PING localhost.localdomain (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=0 ttl=255 time=166 usec
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=1 ttl=255 time=113 usec
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=2 ttl=255 time=119 usec
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=3 ttl=255 time=106 usec
--- localhost.localdomain ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.106/0.126/0.166/0.023 ms
[root@localhost /root]#
```

Po dodaniu interfejsu petli zwrotnej i sprawdzeniu tras, nalezy dodac do jadra sterownik Ethernet. Mozna to zrobic za pomoca tej samej procedury, ktora stosowalismy w przypadku sterownika petli zwrotnej. Po pierwsze, nalezy zainstalowac i aktywować interfejs Ethernet za pomoca polecenia ifconfig:

```
ifconfig eth0 adres_IP
```

W powyzszym poleceniu eth0 jest interfejsem Ethernet. Aby sprawdzic interfejs, nalezy wydac polecenie:

```
ifconfig eth0
```

Nastepnym krokiem w konfigurowaniu sterownika Ethernet jest dodanie wpisu w tablicy tras, aby jadro poznalo adres sieciowy lokalnego komputera. Mozna tez ustawic adres sieciowy dla caiej sieci lokalnej za pomoca opcji -net:

```
route add -net adres_IP
```

Alternatywa jest uzycie pliku */etc/networks*, który zawiera liste nazw sieci i ich adresów IP. Na przyklad, jesli plik */etc/networks* zawiera wpis dla sieci o nazwie *tcp_net*, mozna te siec dodac do tablicy tras w sposob nastepujacy:

```
route add tcp_net
```

Po dodaniu wpisu trasy dla interfejsu Ethernet, mozemy sprawdzic trasowanie za pomoca polecenia ping, podobnie jak zrobilismy to wczesniej dla interfejsu petli zwrotnej.



Niekotore z najnowszych dystrybucji systemu Linux zawieraja program-polecenie o nazwie netconf. Polecenie to udostepnia interfejs graficzny, który wyświetla opcje potrzebne przy konfiguracji TCP/IP.

Instalacja i konfiguracja TCP/IP w świecie Microsoftu

Świat systemów operacyjnych Microsoftu jest dosyć duży; obejmuje takie systemy operacyjne, jak Windows 95, Windows 98, Windows NT, Windows 2000, Windows Me i najnowszy Windows XP. W tym podrozdziale poznamy sposób instalowania i konfigurowania TCP/IP w różnych systemach operacyjnych Microsoftu.

Instalacja TCP/IP w systemach operacyjnych Microsoftu

TCP/IP jest zazwyczaj instalowany razem z systemem operacyjnym, aczkolwiek w systemach operacyjnych Microsoftu można ten protokół zainstalować również później.

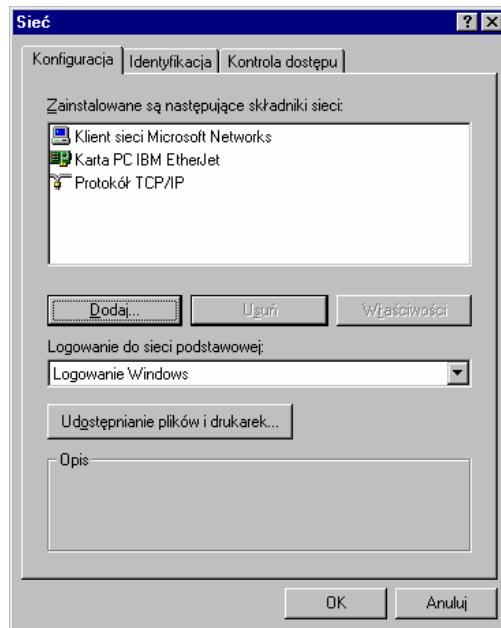
Microsoft Windows 98

Aby zainstalować TCP/IP w komputerze Windows 98:

1. Wybierz *Start/Ustawienia/Panel sterowania*, aby otworzyć okno *Panel sterowania*.
2. Kliknij dwukrotnie ikonę *Sieć*, aby otworzyć okno dialogowe *Sieć* (przedstawione na rysunku 8.6). Domyslnie aktywna jest zakładka *Konfiguracja*.

Rysunek 8.6.

Okno dialogowe Sieć

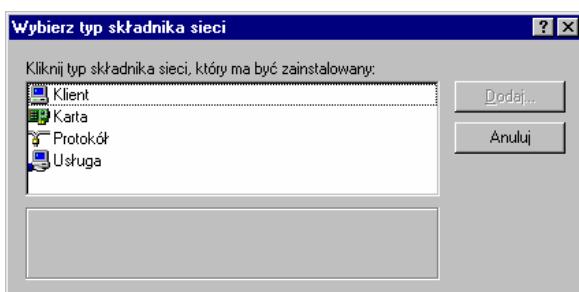


Okno dialogowe *Sieć* możemy również otworzyć przez kliknięcie prawym przyciskiem myszy ikony *Otoczenie sieciowe* na pulpicie i wybranie z menu podręcznego pozycji *Właściwości*.

- 3.** Kliknij *Dodaj...*, aby otworzyc okno dialogowe *Wybierz typ składnika sieci* (przedstawione na rysunku 8.7).

Rysunek 8.7.

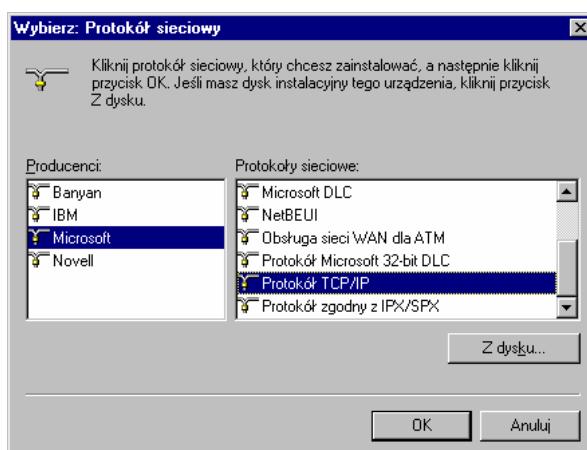
Okno dialogowe
Wybierz typ
składnika sieci



- 4.** Wybierz *Protokół*, a nastepnie kliknij *Dodaj...*, aby otworzyc okno dialogowe *Wybierz: Protokół sieciowy* — rysunek 8.8.

Rysunek 8.8.

Okno dialogowe
Wybierz: protokół
sieciowy



- 5.** Z listy *Producenci* wybierz *Microsoft*. Lista *Protokoły sieciowe* po prawej zawiera wszystkie protokoly Microsoftu.

- 6.** Z listy Protokoły sieciowe wybierz *TCP/IP* i kliknij *OK*.

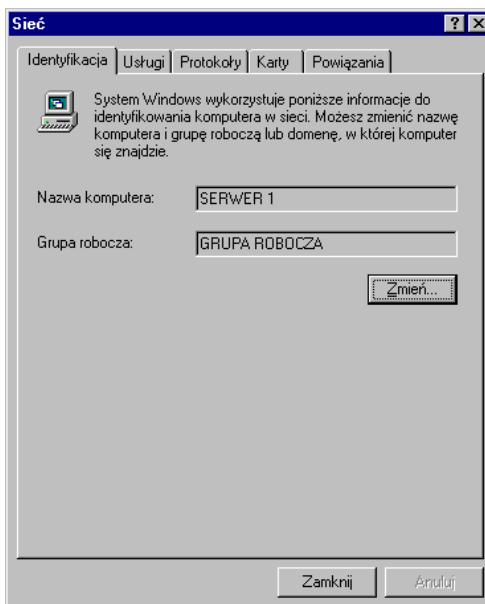
Po zainstalowaniu TCP/IP wpis dla tego protokolu pojawi sie na liscie zainstalowanych składników.

Microsoft Windows NT Server

Przed rozpoczęciem instalacji TCP/IP nalezy upewnic sie, czy jesesmy zalogowani jako Administrator lub czlonek grupy Administratorzy.

- 1.** Wybierz *Start/Ustawienia/Panel sterowania*, aby otworzyc okno *Panel sterowania*.
- 2.** Kliknij dwukrotnie ikone *Siec*, aby otworzyc okno dialogowe *Siec* (przedstawione na rysunku 8.9). Domyslnie aktywna jest zakładka *Identyfikacja*.

Rysunek 8.9.
Okno dialogowe Sieć



3. Wybierz zakładkę *Protokoly* i kliknij przycisk *Dodaj*, aby otworzyć okno dialogowe *Wybierz: Protokół sieciowy*.
4. Na liście *Protokoly sieciowe* zaznacz *Protokół TCP/IP* i kliknij *OK*.
5. Jesli serwer DHCP jest skonfigurowany, pojawi sie okno komunikatu z zapytaniem czy uzyc serwera DHCP, czy nie. Na potrzeby konfiguracji recznej wybierz *Nie*.



DHCP jest skrótem od *Dynamic Host Configuration Protocol* (Protokół dynamicznej konfiguracji hosta). Protokół ten sluzy do automatycznej konfiguracji i adresowania komputerów uzywajacych TCP/IP. Wiecej informacji o DHCP mozna znalezc w rozdziale 10.

6. W oknie dialogowym *Konfiguracja systemu Windows NT* wprowadz pelna sciezke do plików dystrybucji Windows NT i kliknij przycisk *Dalej*, aby skopiowac wszystkie niezbedne pliki na dysk twardy.



Jesli wybrane zostaly opcje instalacji SNMP i FTP, uzytkownik zostanie poprowadzony do automatycznej konfiguracji tych uslug.

Po skopiowaniu plików i zakonczeniu instalacji niezbedny jest restart komputera, po czym protokół TCP/IP pojawi sie w oknie dialogowym ustawien sieci.

Microsoft Windows 2000 Server

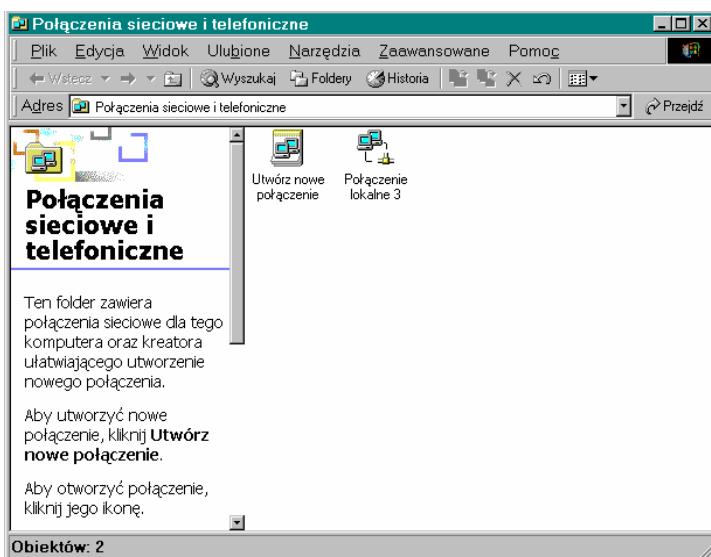
TCP/IP jest instalowany domyslnie, gdy karta adaptera sieciowego zostala wykryta automatycznie podczas instalacji i konfiguracji systemu Windows 2000 Server. Jesli jednak domyslne ustawienia dotyczace protokolu TCP/IP zostaly recznie zmienione podczas instalacji systemu operacyjnego, niezbedna bedzie instalacja TCP/IP.

Przed rozpoczęciem instalacji TCP/IP w systemie Windows 2000 Serwer musimy upewnic sie, czy jesesmy zalogowani jako Administrator lub czlonek grupy Administratorzy. Aby zainstalowac TCP/IP:

1. Wybierz *Start/Ustawienia/Polaczenia sieciowe i telefoniczne*, aby otworzyc okno dialogowe *Polaczenia sieciowe i telefoniczne* — rysunek 8.10.

Rysunek 8.10.

Okno dialogowe
Polaczenia sieciowe
i telefoniczne



2. Kliknij prawym przyciskiem myszy polaczenie, dla którego ma zostac zainstalowany TCP/IP i z menu podreznego wybierz *Wlasciwosci*. Aby skonfigurowac siec lokalna, kliknij prawym przyciskiem myszy *Połaczenie lokalne* i wybierz *Wlasciwosci* z menu podreznego, aby otworzyc okno *Wlasciwosci polaczenia lokalnego*.
3. Wybierz zakladke *Ogólne*.
4. Jesli lista zainstalowanych skladnikow nie zawiera pozycji *Protokół internetowy (TCP/IP)*, kliknij *Instaluj*, aby rozpoczac proces instalacji.
5. Kliknij *Protokół*, a nastepnie *Dodaj*, aby otworzyc okno dialogowe *Wybierz protokół sieciowy*.
6. Kliknij *Internet Protocol (TCP/IP)*, a nastepnie *OK*.

Gdy system tego zazada, podaj pelna sciezke do plików dystrybucji. Po skopiowaniu plików niezbedny bedzie restart komputera, po czym protokół TCP/IP pojawi sie na liście zainstalowanych skladnikow.

Reczna konfiguracja TCP/IP

Pokazalismy jak dotad metody instalowania TCP/IP w różnych systemach operacyjnych Microsoftu. Teraz musimy opisac, jak aktywować rózne uslugi TCP/IP, aby skonfigurowac protokół. Ponizej zostały przedstawione sposoby konfiguracji TCP/IP:

- ◆ *Konfiguracja automatyczna* — automatycznie przydziela domyslne adresy IP z zarezerwowanego zakresu od 169.254.0.1 do 169.254.255.254 z maska podsieci 255.255.0.0, jednakże brama i serwery usług WINS i DNS nie są konfigurowane automatycznie. Ta metoda jest zaprojektowana na potrzeby sieci składających się z pojedynczego segmentu i nie połączonych z Internetem, ponieważ sieci takie nie wymagają bram ani serwerów WINS i DNS.
- ◆ *Konfiguracja dynamiczna* — ta metoda wymaga obecności serwera DHCP w sieci. W metodzie konfiguracji dynamicznej hosty otrzymują adresy IP, maski podsieci i informacje o bramach, serwerach DNS i serwerach WINS od serwera DHCP.
- ◆ *Konfiguracja reczna* — gdy sieć składa się z wielu segmentów i nie posiada serwera DHCP, TCP/IP trzeba skonfigurować ręcznie. W tej metodzie trzeba ręcznie przydzielac takie informacje, jak adres IP, maska podsieci i konfiguracja usług DNS i WINS.

Aby skonfigurować TCP/IP dla systemów operacyjnych Microsoftu, należy podać następujące dane:

- ◆ *Adres IP* — każdy interfejs sieciowy w każdym hostie musi posiadać unikatowy adres IP. Ta pozycja jest niezbędna.
- ◆ *Maska podsieci* — każdy interfejs sieciowy w każdym hostie musi posiadać maskę podsieci, aby mógł otrzymać identyfikator sieci na podstawie adresu IP i maski podsieci. ID sieci powinien być taki sam dla wszystkich interfejsów sieciowych w segmencie, wobec czego maska podsieci dla wszystkich interfejsów w jednym segmencie sieci musi być identyczna. Ta pozycja jest niezbędna.
- ◆ *Brama domyslna* — brama jest lokalny ruter, który przekazuje pakiety do innych sieci. Przynajmniej jeden z interfejsów sieciowych powinien mieć skonfigurowany adres IP bramy domysłnej, aby hosty TCP/IP mogły komunikować się z innymi sieciami. Ta pozycja nie jest potrzebna, jeśli sieć składa się z pojedynczego segmentu.
- ◆ *Serwer DNS (Domain Name System)* — dla hosta TCP/IP możemy podać adres IP serwera DNS obecnego w sieci. Serwer DNS kojarzy nazwy FQDN z odpowiadającymi im adresami IP. Ten proces rozwiązywania nazw jest bardzo istotny dla komunikacji pomiędzy hostami.



Rozdział 10. zawiera bardziej szczegółowe informacje o usłudze DNS.

- ◆ *Serwer WINS* — dla hosta TCP/IP możemy skonfigurować adres IP serwera usługi WINS w sieci. Serwer WINS kojarzy nazwy NetBIOS z odpowiadającymi im adresami IP. NetBIOS jest protokołem, który pozwala programom aplikacji komunikować się ze sobą przez sieć. Programy i usługi sieciowe, takie jak udostępnianie plików i drukarki w Microsoft NT, korzystają z nazw NetBIOS.



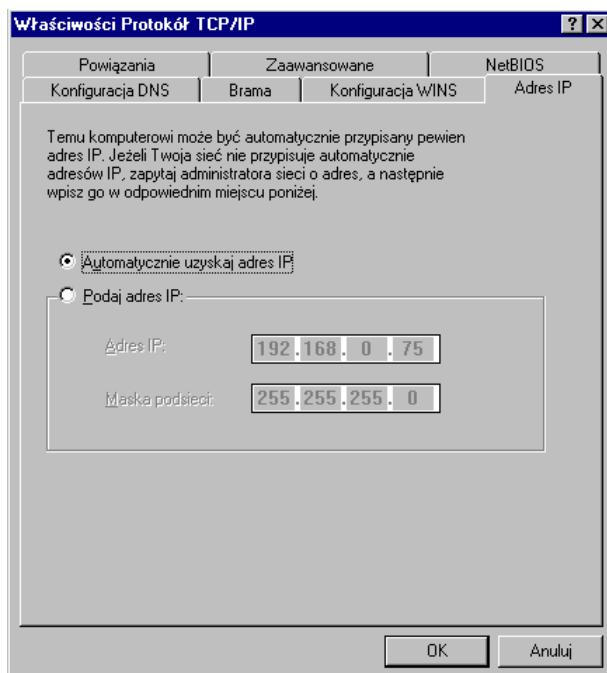
Rozdział 10. zawiera więcej informacji o NetBIOS-ie.

Microsoft Windows 98

Jesli siec zawiera serwer DHCP, protokol TCP/IP mozna skonfigurowac dynamicznie. W przeciwnym razie musimy skonfigurowac TCP/IP recznie, w nastepujacy sposob:

1. W oknie dialogowym Siec wybierz *TCP/IP* i kliknij *Wlasciwosci*, aby otworzyc okno dialogowe *Wlasciwosci Protokol TCP/IP* — pokazane na rysunku 8.11.

Rysunek 8.11.
Okno dialogowe
Wlasciwosci
Protokol TCP/IP



2. W zakladce *Adres IP* wybierz *Podaj adres IP* i wpisz adres IP oraz maske podsieci.
3. Skonfiguruj w razie potrzeby brame i serwery DNS i WINS, korzystajac odpowiednio z zakladek *Brama*, *Konfiguracja WINS* i *Konfiguracja DNS*.

Po zakonczeniu konfiguracji komputer trzeba zrestartowac, aby zmiana ustawien odniosla skutek.

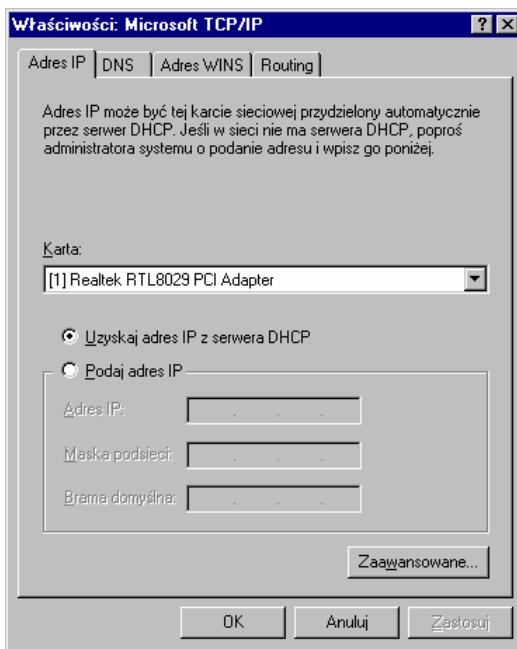
Microsoft Windows NT

Jesli w oknie dialogowym *Wlasciwosci TCP/IP* wybierzemy opcje *Zezwól na automatyczna konfiguracje DHCP*, zas w sieci dostepny jest serwer DHCP, ustawienie konfiguracji TCP/IP odbedzie sie automatycznie. W przeciwnym razie musimy skonfigurowac TCP/IP recznie:

1. W oknie dialogowym *Wlasciwosci sieci* (w polu *Oprogramowanie sieci*) wybierz *protokol TCP/IP*. Nastepnie kliknij przycisk *Wlasciwosci*, aby otworzyc okno dialogowe *Wlasciwosci: Microsoft TCP/IP* — rysunek 8.12.

Rysunek 8.12.

*Okno dialogowe
Własciwosci: Microsoft
TCP/IP*



2. Z listy *Adapter* wybierz adapter sieciowy przeznaczony do skonfigurowania.
Lista ta zawiera wszystkie adaptery zainstalowane w komputerze.
3. Wybierz *Podaj adres IP*. W polu *Adres IP* wprowadź adres IP lokalnego komputera.
W polu *Maska podsieci* wpisz adres maski podsieci, sluzacy komputerowi do podzialu adresu IP na ID hosta oraz ID sieci. W polu *Brama domyslna* wpisz adres bramy domyslnej (rutera IP), sluzacej do przekazywania pakietów do innych sieci i podsieci.

Uwaga
Jesli adres bramy domyslnej nie zostanie podany, nie bedzie mozna przesylic pakietow na zewnatrz podsieci, o ile nie wykorzystamy narzecza route.

4. Jesli serwer DNS ma byc uzywany do rozwiązywania nazw, kliknij zakładkę *DNS* i wprowadz informacje o serwerze DNS.
5. Jesli do rozwiązywania nazw ma byc uzywany serwer WINS i jest on dostepny w danej sieci, wybierz zakładkę *WINS* i wprowadz jego adres. Jezeli serwer WINS nie zostanie wyszczególniony, rozwiązywanie nazw NetBIOS bedzie ograniczone do lokalnej sieci.
6. Aby zalać trasowanie pakietów, wybierz zakładkę *Routing* i zaznacz pole wyboru *Wlacz przekazywanie IP*.

Uwaga
Protokół RIP (*Routing Information Protocol*) pozwala na statyczne i dynamiczne ustalanie tras. Usluga RIP, która umozliwia funkcjonowanie protokolu RIP, moze zostac zainstalowana z zakładki *Uslugi* w oknie dialogowym *Siec*.

7. Kliknij przycisk *OK*, aby zamknac okno *Wlasciwosci TCP/IP*.

8. Kliknij *OK*, aby zamknac okno dialogowe *Siec*.

Jezeli protokol TCP/IP jest instalowany w komputerze po raz pierwszy, niezbedny jest restart komputera, aby zmiana ustawien odniosla skutek.

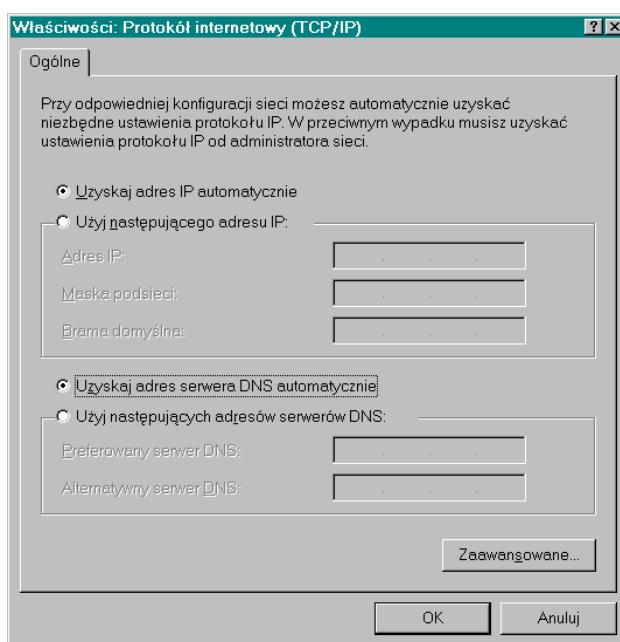
Microsoft Windows 2000 Server

Aby skonfigurowac recznie TCP/IP:

- 1.** Otwórz okno *Sieci i polaczenia telefoniczne*.
- 2.** Prawym przyciskiem myszy kliknij polaczenie sieciowe przeznaczone do skonfigurowania i wybierz *Wlasciwosci* z menu podrecznego.
- 3.** Wybierz zakladke *Ogólne*.
- 4.** Wybierz *Protokół internetowy (TCP/IP)* i kliknij *Wlasciwosci*, aby otworzyc okno dialogowe *Wlasciwosci: Protokół internetowy (TCP/IP)*; okno jest pokazane na rysunku 8.13.

Rysunek 8.13.

Okno dialogowe
Wlasciwosci:
Protokół
internetowy
(TCP/IP)



5. Wybierz *Uzyj nastepujacego adresu IP*. Jesli polaczenie jest typu lokalnego, wpisz adres IP, maske podsieci i brame domyslna (w razie potrzeby). Dla innych polaczen podaj adres IP.

W tym samym oknie mozemy skonfigurowac serwer DNS, wybierajac *Uzyj nastepujacego adresu serwera DNS*.

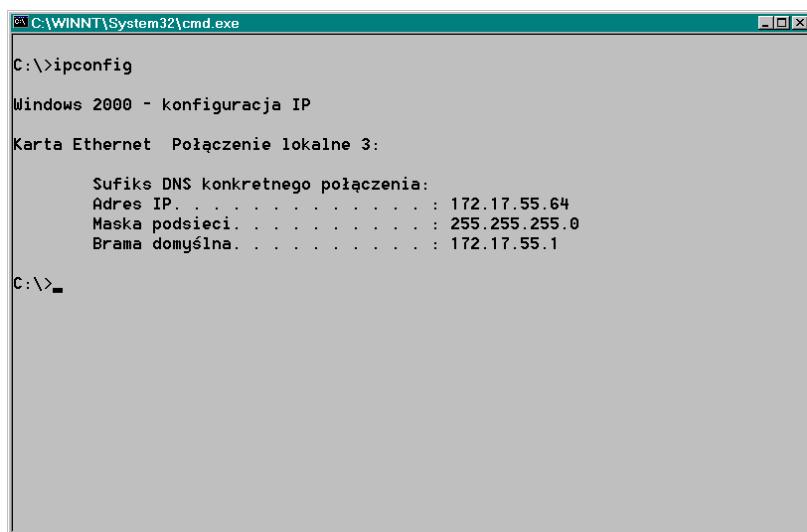


Jesli protokół TCP/IP został zainstalowany po raz pierwszy, komputer należy uruchomić ponownie, aby zmiany odniosły skutek. Jesli zmieniane są tylko ustawienia — restart komputera nie jest wymagany.

Kontrola konfiguracji IP

Gdy pojawiają się problemy z siecią, pierwszym krokiem w kierunku rozwiązania problemu jest sprawdzenie danych konfiguracyjnych IP, do których zalicza się adres IP, maska podsieci i brama domyslna. Informacje te można otrzymać za pomocą narzędzia `ipconfig` (uruchamianego z wiersza poleceń). Rysunek 8.14 przedstawia przykładowy wynik polecenia `ipconfig`.

Rysunek 8.14.
Przykładowy wynik
polecenia `ipconfig`



C:\>ipconfig

Windows 2000 - konfiguracja IP

Karta Ethernet Połączenie lokalne 3:

Sufiks DNS konkretnego połączenia:
Adres IP. : 172.17.55.64
Maska podsieci. : 255.255.255.0
Brama domyślna. : 172.17.55.1

C:\>_



Do sprawdzenia konfiguracji IP w systemach Windows 95 i Windows 98 służy narzędzie `winipcfg`.

Szczegółowe informacje możemy otrzymać, używając polecenia `ipconfig` z parametrem `/all`. Polecenie `ipconfig /all` wyświetla szczegółowy raport o konfiguracji wszystkich interfejsów.

Rozdział 9.

Konfiguracja automatyczna

W tym rozdziale:

- ◆ Wprowadzenie do konfiguracji automatycznej
- ◆ Protokół BOOTP
- ◆ Protokół DHCP

Wiekszosc z nas odkryla, iz prawidlowe zainstalowanie i konfiguracja TCP/IP na potrzeby lacnosci i eksploatacji sieci to zadania wymagajace ciaglej pracy. Administrator musi posiadac duze doswiadczenie, aby zmusic instalacje do pracy. W trakcie lektury Czytelnik zapewne zastanawial sie przynajmniej raz, czy calego tego procesu nie dalo-by sie zautomatyzowac, podobnie jak w przypadku technologii *plug-and-play*, ktora pozwala uzytkownikom korzystac z oprogramowania natychmiast po zainstalowaniu, bez koniecznosci recznej konfiguracji. W automatycznej konfiguracji TCP/IP w sieci naj-wazniejsza role graja protokol BOOTP (*Bootstrap Protocol*) oraz jego nastepca — DHCP (*Dynamic Host Configuration Protocol*).

W niniejszym rozdziale przedstawione zostana sposoby automatycznego konfigurowa-nia protokolu TCP/IP oraz rola w tym procesie protokolów BOOTP i DHCP. Omówimy proces ladowania poczatkowego (*bootstrap*), format pakietu danych BOOTP, slabe strony tego protokolu oraz rutery stosujace BOOTP. Opiszemy tutaj również proces DHCP, zasady dzierzawy, zakresy i opcje serwera DHCP, format pakietu danych DHCP i trasowanie DHCP.

Wprowadzenie do konfiguracji automatycznej

Aby zainstalowac i skonfigurowac oprogramowanie TCP/IP, potrzebne sa nastepujace informacje:

- ◆ adresy IP urzadzen TCP/IP,
- ◆ adresy sieci,
- ◆ maski podsieci,
- ◆ nazwa domeny, do ktorej nalezy urzadzenie,
- ◆ adres bramy domyslnej (rutera),

- ♦ adres serwera nazw.

Automatyczna konfiguracja w przypadku TCP/IP oznacza, ze uzytkownicy nie musza recznie wpisywac danych, zamiast tego moga korzystac z komputera zaraz po podlaczaniu do sieci.



Automatyczna konfiguracja jest często nazywana autokonfiguracją.

Korzyści z konfiguracji automatycznej

Konfiguracja automatyczna ma wiele zalet w porównaniu z tradycyjnymi metodami recznego konfigurowania TCP/IP. Korzyści te stana sie bardziej oczywiste, gdy porównamy obie metody — patrz tabela 9.1.

Tabela 9.1. Konfiguracja reczna i konfiguracja automatyczna — porównanie

Konfiguracja reczna	Konfiguracja automatyczna
Administrator sieci musi przydzielic unikatowy adres IP do kazdego urzadzenia. Jesli urzadzen w sieci jest duzo, zadanie to moze byc nuzace.	Kazde urzadzenie sieciowe automatycznie otrzymuje unikatowy adres IP. W rezultacie praca administratora jest o wiele latwiejsza.
Niewlasciwe lub powtarzajace sie adresy IP moga powodowac mnóstwo klopotów, poniewaz administrator musi recznie wyszukiwac urzadzenia o niepoprawnych adresach.	Poniewaz adresy przydzielane sa automatycznie, prawdopodobienstwo wystapienia blednych lub powtarzajacych sie adresow jest praktycznie zerowe.
Administrator oprócz adresów IP musi wprowadzac maski podsieci i adresy domyslnych ruterów (bram).	Informacje dotyczace adresu domyslnego routera oraz maski podsieci sa konfigurowane automatycznie.
W kazdym urzadzeniu dodatkowe informacje, takie jak strefa czasowa, adres IP serwera czasu, adres IP serwera inicjujacego i nazwa pliku inicjujacego musza byc konfigurowane recznie.	Dodatkowe informacje sa konfigurowane automatycznie.
Administratorzy moga miec problemy z przenoszeniem urzadzen z jednej podsieci do drugiej.	Przenoszenie urzadzen miedzy podsieciami nie stanowi zbyt duzego problemu, poniewaz przeniesione urzadzenia powinny odpowiednio skonfigurowac sie automatycznie.
Administrator musi recznie zarzadzac urzadzeniami w sieci.	Automatyczna konfiguracja pozwala na centralne zarzadzanie urzadzeniami, co oszczedza administratorowi biegania od komputera do komputera.
Konfiguracja reczna jest wysoce zalezna od administratora.	Konfiguracja automatyczna znaczaco zmniejsza zakres odpowiedzialnosci administratorow, zas uzytkownicy nie musza byc calkowicie od nich zalezni.



Adresy serwerów domen i nazw nie mogą być konfigurowane automatycznie. Musimy wpisać je ręcznie.

Konfiguracja w sieciach wielosegmentowych

Obecnie, w środowisku dużych korporacji naciąg przeniósł się z sieci lokalnych na globalne sieci wielosegmentowe. Sieć taka składa się z małych, średnich i dużych sieci lokalnych połączonych ruterami. To przesunięcie środka ciezkoci wywołało kilka problemów, związanych z automatyczna konfiguracja TCP/IP:

- ◆ Urządzenie sieciowe podczas uruchomienia do automatycznego skonfigurowania wymaga informacji, które zasadniczo otrzymuje z zewnętrznego źródła — serwera inicjującego (kazdy komputer przechowujący wymagane do rozruchu informacje jest tzw. *serwerem inicjującym — boot server*). Jeśli serwer ten mieści się w lokalnej podsieci, otrzymanie informacji może być łatwe. Jeśli jednak serwer inicjujący mieści się w innej podsieci, zadanie danych konfiguracyjnych musi zostać przesłane przez ruter. Tego typu informacje zazwyczaj nie są przekazywane przez routery.
- ◆ Jeśli serwer inicjujący jest wyłączony lub z jakiegoś powodu niedostępny, cała sieć może przestać działać, ponieważ hosty nie będą mogły otrzymać od serwera danych uruchomieniowych.

Reczna konfiguracja informacji w urządzeniach sieciowych jest nudną pracą, zwłaszcza w dużych sieciach. Ponadto routery nie są zdolne do przesyłania zadań i odpowiedzi związanych z konfiguracją, co doprowadziło do opracowania kolejno dwóch protokołów — BOOTP i DHCP. Protokoły te udostępniają niezbędne mechanizmy przesyłania danych konfiguracyjnych do hostów w sieci TCP/IP, aby wyeliminować konieczność recznej konfiguracji poszczególnych urządzeń w sieci. Inaczej mówiąc, protokoły BOOTP i DHCP pozwalają na automatyczne konfigurowanie niezbędnych informacji przez urządzenia sieciowe, na podłączenie do sieci i rozpoczęcie pracy, co odciaza w pewnym stopniu i tak zapracowanych administratorów sieciowych.

Protokół BOOTP

Każde urządzenie potrzebuje w chwili uruchomienia danych systemowych. Te informacje uruchomieniowe — inaczej *informacje inicjujące (boot information)* — są w komputerze umieszczone w sektorze ładowania początkowego dysku twardego. Jednakże w przypadku komputerów bezdyskowych informacje te nie są dostępne. W środowisku sieciowym takie komputery również potrzebują unikatowych adresów IP, wobec tego muszą je otrzymać z zewnętrznego źródła. Z tego powodu komputery bezdyskowe, tzw. „gluche terminale” (*dumb terminal*), używają protokołu RARP (*Reverse Address Resolution Protocol* — protokół odwrotnego rozwiązywania adresów) do pobrania poprawnego adresu IP i informacji inicjujących z serwera inicjującego. RARP ma jednak braki, które sprawiają, że nie nadaje się do pobierania danych konfiguracyjnych podczas rozruchu:

- ◆ Pakiet wymieniany pomiędzy serwerem i klientem zawiera tylko czterobajtowy adres IP klienta. Klient do uruchomienia potrzebuje jeszcze dodatkowych informacji, których pakiet RARP nie dostarcza.
- ◆ RARP do identyfikacji hosta uzywa jego adresu MAC, wobec tego nie nadaje sie do uzytku w sieciach, w których adresy sprzętowe sa przydzielane dynamicznie.



Wiecej informacji o protokole RARP zawiera rozdział 4.

Protokół BOOTP (*Bootstrap Protocol*) został opracowany jako srodek zaradczy na niedostatki protokołu RARP. Komunikat (pakiet) BOOTP oprócz adresu IP zawiera informacje startowe, niezbedne, aby z powodzeniem uruchomic komputer bezdyskowy. Ten sam komunikat zawiera tez adres serwera BOOTP i domyslnego routera lub bramy w sieci. Ponadto protokołu BOOTP mozemy z powodzeniem uzywac w sieciach, w których adresy sprzętowe przydzielane sa dynamicznie.

Proces ladowania poczatkowego BOOTP

Proces ladowania poczatkowego (*bootstrap*) w protokole BOOTP sklada sie z dwóch faz. Pierwsza z nich jest *faza ustalenia adresu i wyboru pliku inicjujacego*. Po otrzymaniu przez klienta adresu IP oraz wyborze wymaganego pliku inicjujacego, kontrola przejmuje faza druga — *faza przeslania pliku inicjujacego*. W jej trakcie klient uzywa protokolu transferu danych do pobrania pliku inicjujacego z serwera inicjujacego. Fazy te przebiegaja nastepujaco:

- ◆ *Faza ustalenia adresu i wyboru pliku inicjujacego* — komputer bezdyskowy podczas uruchomienia wysyla zadanie adresu IP oraz pliku inicjujacego do serwera BOOTP przez port 68. Serwer inicjujacy oczekuje na zadania BOOTP (oraz DHCP) na porcie 67. Serwer po zidentyfikowaniu stacji roboczej klienta za pomoca adresu MAC, wyslanego razem z zadaniem klienta, wybiera plik inicjujacy wstepnie skonfigurowany dla danego klienta.



Serwer inicjujacy nie musi dzialac w tym samym komputerze, który przechowuje pliki inicjujace. Ogólnie mówiac, taki serwer uzywa prostej bazy danych, w której pliki inicjujace przypisane sa do nazw lub aliasów. W zwiazku z tym pliki inicjujace moga byc skladowane w innym komputerze, z którym w razie potrzeby serwer inicjujacy sie laczy, aby pobrac odpowiedni plik inicjujacy.

- ◆ *Faza przeslania pliku inicjujacego* — po zidentyfikowaniu przez serwer BOOTP klienta i wyborze odpowiedniego pliku inicjujacego, klient przesyła (kopiuje) ten plik do swojej pamieci za pomoca odpowiedniego protokolu, na przyklad TFTP (*Trivial File Transfer Protocol*) lub FTP.

Zawartosc pakietu BOOTP

Pakiet BOOTP sklada sie z 15 pol, których dlugosc jest stala. Dzieki temu implementacja protokołu BOOTP jest prosta i wystarczajaco niewielka, by zmiescic sie w pamieci klienta. Z tego samego powodu zadania i odpowiedzi BOOTP maja wspólny format. Rysunek 9.1 przedstawia format pakietu BOOTP.

Rysunek 9.1.

*Format
pakietu BOOTP*

0	8	16	24	31
OP	HTYPE	HLEN	HOPY	
ID transakcji				
Sekundy				Nie używane
Adres IP klienta				
Twój adres IP				
Adres IP serwera				
Adres IP routera				
Adres sprzętowy klienta				
⋮				
Nazwa serwera				
⋮				
Nazwa pliku inicjującego				
⋮				
Dane producenta				
⋮				

Pola pakietu BOOTP to:

- ♦ *OP* — pole o długości 1 bajta, określające typ komunikatu. Jeśli komunikat jest zadaniem ze strony klienta, to wartość pola wynosi 1. Gdy jest odpowiedzią serwera inicjującego — wartość pola wynosi 2.
- ♦ *HTYPE (typ sprzętu)* — określa sprzętowy typ interfejsu używanego przez urządzenie nadające. Na przykład, interfejs Ethernet jest reprezentowany przez wartość 1. Pole to ma długość 1 bajta.
- ♦ *HLEN (długość adresu sprzętowego)* — określa długość adresu sprzętowego, zawartego w polu Typ sprzętu. Na przykład, wartość pola równa 6 oznacza adres interfejsu Ethernet. Pole ma długość 1 bajta.
- ♦ *HOPY* — oznacza liczbę serwerów, przez które komunikat był przesyłany. Klient ustawia wartość tego pola na 0. Gdy serwer przesyła komunikat do następnego serwera, liczba hopów (przeskoków) zwiększa się o 1. Pole o długości 1 bajta.
- ♦ *XID (ID transakcji)* — zawiera generowaną losowo liczbę całkowitą, używaną przez klienta do zestawienia swojego zadania z odpowiedzią na nie. Pole to ma długość 4 bajtów.
- ♦ *Sekundy* — liczba sekund od rozpoczęcia procesu uruchomienia klienta. Pole o długości 2 bajtów.
- ♦ *Nie używane* — pole nie używane, o długości 2 bajtów.
- ♦ *CIAddr (Adres IP klienta)* — jeśli klient zna swój adres IP, to pole zawiera ten adres; w przeciwnym razie — ma wartość równą 0. Pole o długości 4 bajtów.
- ♦ *YIAddr (Twój adres IP)* — jeśli w komunikacie zadającym inicjalizacji odebranym od klienta pole CIAddr było puste, to pole YIAddr zawiera podany przez serwer adres IP klienta. Inaczej mówiąc, serwer wypełnia to pole, gdy klient nie zna swojego adresu IP; w przeciwnym razie pole jest ignorowane. Pole o długości 4 bajtów.

- ◆ *SIAddr (Adres IP serwera)* — zawiera adres IP serwera, który moze byc wprowadzony zarówno przez serwer w komunikacie odpowiedzi, jak i przez klienta w komunikacie zadania inicjacji. Jesli klient zna adres IP serwera, od którego moze otrzymac dane uruchomieniowe, to wypelnia to pole; w przeciwnym razie pole otrzymuje wartosc 0. Dowolny serwer inicjujacy, który potrafi odpowiedziec na zadanie, w komunikacie odpowiedzi wprowadza własny adres IP. Pole o dlugosci 4 bajtów.
- ◆ *GIAddr/RIAddr (Adres IP bramy lub rutera)* — zawiera adres IP domyslnego routera lub Bramy. Pole to jest opcjonalne i wymagane tylko wtedy, gdy serwer inicjujacy mieści się w innej podsieci. Pole o dlugosci 4 bajtów.
- ◆ *CHAddr (adres sprzutowy klienta)* — zawiera adres sprzutowy (MAC) klienta. Pole wypelniane jest przez klienta i ma dlugosc 16 bajtów.
- ◆ *Nazwa serwera* — zawiera nazwe hosta serwera i moze byc wypelnione zarówno przez serwer w komunikacie odpowiedzi, jak i przez samego klienta w komunikacie zadania inicjacji. Jesli klient zna nazwe serwera, od którego moze otrzymac dane uruchomieniowe, to wypelnia to pole; w przeciwnym razie pole otrzymuje wartosc 0. Dowolny serwer inicjujacy, który potrafi odpowiedziec na zadanie, w komunikacie odpowiedzi wprowadza do tego pola własna nazwe. Uzycie pola jest nieobowiazkowe; jego dlugosc wynosi 64 bajty.
- ◆ *Nazwa pliku inicjujacego* — zawiera ogólna nazwe pliku inicjujacego, potrzebnego klientowi do pomyslnego uruchomienia. Pole to moze byc wypelnione przez klienta, jesli zna on nazwe pliku, albo przez serwer w komunikacie odpowiedzi. Nazwa zawiera pełna sciezka dostepu; dlugosc pola wynosi 128 bajtów.
- ◆ *Dane producenta* — zawiera zamieszczone przez producenta opcjonalne informacje, które musza zostac przekazane z serwera do klienta. Do tych danych w zadaniu inicjacji moze zaliczac sie typ sprzetu lub numer seryjny klienta, zas w odpowiedzi identyfikator zdalnego systemu plików. Moga sie tu również znalezc: maska podsieci dla lokalnej sieci, adres IP serwera czasu, adres IP serwera domeny lub rozmiar pliku inicjujacego. Pole o dlugosci 64 bajtów.

Rutery obsługujace protokół BOOTP

W sieciach TCP/IP rutery sluzą do laczenia urzadzen i wymiany informacji pomiedzy różnymi fizycznymi segmentami sieci, które noszą nazwe *podsieci*. Sytuacja, w której klient i serwer inicjujący położone są w różnych podsieciach jest całkiem prawdopodobna, zwłaszcza w środowiskach, gdzie używane są komputery przenosne. Aby umożliwić uruchamianie przez ruter, zadania BOOTP muszą przejść przez jeden lub kilka routerów.



Dodatkowe informacje o routeraх i podsieciach znajdują się w rozdziale 5.

Gdyby pakiety BOOTP nie były przepuszczane przez routery, administrator sieci musiałby umieścić w każdej podsieci osobny serwer — zadanie kosztowne i czasochłonne. Można jednak znaleźć na rynku routery, które rozpoznają pakiety BOOTP i pozwalają na ich przesyłanie do miejsca przeznaczenia. Są to tzw. *rutery obsługujące BOOTP lub routery obsługujące BOOTP i DHCP (BOOTP-enabled router, BOOTP/DHCP-enabled router)*.

router). Sa one dostosowane do funkcjonalnosci agentów przekazujacych BOOTP (BOOTP relay agent). Jak nazwa sugeruje, agent przekazujacy BOOTP przekazuje komunikaty pomiedzy klientami i serwerami inicjującymi polożnymi w odrebnym sieciach.



Czytelnik na stanowisku administratora sieci moze zetknac sie z sytuacją, w której uruchamianie klientów z serwera po drugiej stronie routera bedzie niezbedne, lecz ruter nie bedzie przepuszczal komunikatów BOOTP. Jesli na dodatek ograniczenia budżetu nie pozwala zainwestowac w nowy ruter obsługujacy protokół BOOTP, mozna uzyc serwera proxy (lub innego) i skonfigurowac go do roli agenta przekazujacego. W tym celu wystarczy zainstalowac sieciowy system operacyjny — na przyklad Windows NT 4.0 lub Windows 2000 — który ma wbudowanego agenta przekazujacego BOOTP i DHCP.

Wady protokolu BOOTP

Wraz ze wzrostem popularnosci komputerów przenosnych srodowisko sieciowe zmieniło swój charakter ze statycznego na dynamiczny. W srodowisku statycznym kazde urzadzenie jest na stale podlaczone do sieci i konfiguracja sieci nie ulega zmianie przez tygodnie lub nawet miesiace. Jednakze w srodowisku dynamicznym konfiguracja moze zmieniac sie codziennie, poniewaz palmtopy, notebooki i podobne urzadzenia sa latwe do przenoszenia z miejsca na miejsce.

Protokół BOOTP został opracowany dla statycznego srodowiska sieciowego, w którym raz utworzony plik konfiguracyjny BOOTP mógł byc używany do określenia parametrów wszystkich urzadzen, które informacji potrzebowaly. Plik ten zawieral odwzorowania wszystkich hostów w sieci razem z parametrami dla nich. Im wiecej hostów w sieci potrzebowalo danych inicjacyjnych z zewnetrznego źródła, tym wieksza objetosc miał plik konfiguracyjny BOOTP.

Z uwagi na statyczny charakter sieci, pliku konfiguracyjnemu BOOTP nie trzeba było aktualizowac zbyt często, jednakze w sieciach dynamicznych zarzadzanie plikiem BOOTP staje sie zajeciem pełnoetatowym. Po kazdym przeniesieniu urzadzenia z jednej lokalizacji do innej administrator sieci musi dokonac w bieżacych ustawieniach nastepujacych zmian:

- ◆ Ponownie wprowadzic parametry BOOTP dla urzadzenia — zadanie czasochlonne, jesli spora liczba urzadzen często zmienia polozenie.
- ◆ Przydzielic unikatowe adresy IP dla hostów przenoszonych do innej domeny lub podsieci.

Z powyzszych powodów BOOTP nie nadazyl za szybko rozwijajacym sie srodowiskiem sieci dynamicznych. W rezultacie organizacja IETF (*Internet Engineering Task Force*) opracowala zaawansowana wersje BOOTP o nazwie DHCP (*Dynamic Host Configuration Protocol* — protokół dynamicznej konfiguracji hosta). Protokół DHCP został zaprojektowany, by zaradzic wiekszosci niedostatków protokolu BOOTP.

DHCP

Podobnie jak BOOTP, protokół DHCP przydziela pelne dane konfiguracyjne do uruchamianego urzadzenia sieciowego. Jest jednak znacznie przydatniejszy od BOOTP, poniewaz dodatkowo pozwala urzadzeniom automatycznie pobierac adresy IP. W rezul-

tacie klient DHCP moze byc przenoszony bez koniecznosci recznej zmiany konfiguracji. Ta zdolnosc do automatycznej rekonfiguracji ma znaczenie zwlaszcza w przypadku tymczasowych przenosin, gdy host przenoszony jest do innej lokalizacji na bardzo krótki czas (na przyklad, na kilka godzin lub dzien). Oprócz dynamicznego przydzielania adresu IP, DHCP posiada wbudowane mechanizmy sluzace do zarzadzania lokalnymi klientami w sieci, rejestracji ruchu sieciowego i podstawowe zabezpieczenia. Równie ważna jest latwosc instalacji, konfiguracji i utrzymywania DHCP. Dzieki tym wszystkim zaletom zadanie zarzadzania siecia TCP/IP z pomoca DHCP staje sie stosunkowo latwe.

Protokół DHCP nie tylko rozwiązuje problem dynamicznego przydzielania adresów IP klientom w sieci TCP/IP, lecz również radzi sobie z problemem szybko kurczacej się puli unikatowych adresów IP. W statycznym srodowisku sieciowym administrator nadaje unikatowe adresy IP nowym urzadzeniom w sieci. Nawet jesli urzadzenie uzywane jest rzadko (lub tymczasowo), zadne inne urzadzenie nie moze korzystac z jego adresu IP, chocby byl bardzo potrzebny. Dzieki zastosowaniu protokołu DHCP, który dynamicznie przydziela adresy IP, adresy te sa przyznawane tylko w miare potrzeby i zwalniane, gdy nie sa potrzebne, co pozwala oszczedzic cenna przestrzen adresów IP.

Dzierzawy DHCP

Serwer DHCP utrzymuje pule poprawnych adresów IP, które moze przydzielac klientom. Pula adresów IP nosi nazwe *zakresu* (*scope*). Klient w trakcie uruchamiania roznosi zadanie adresu IP. Wszystkie serwery DHCP, które otrzymaja zadanie, zwracaja w odpowiedzi adres IP i zwiiazane z nim dane konfiguracyjne, dzieki czemu klient moze otrzymanie wiele odpowiedzi na zadanie. Nastepnie klient wybiera do swojego uzytku odpowiedni adres IP i dzierzawi go od serwera DHCP. *Dzierzawa* (*lease*) okresla czas, przez który serwer DHCP pozwala klientowi uzywac określonego adresu IP. Po potwierdzeniu dzierzawy klient staje sie częścią funkcjonujacej sieci. Gdy ustalony czas uplynie, dzierzawa jest uniewazniana przez serwer DHCP.

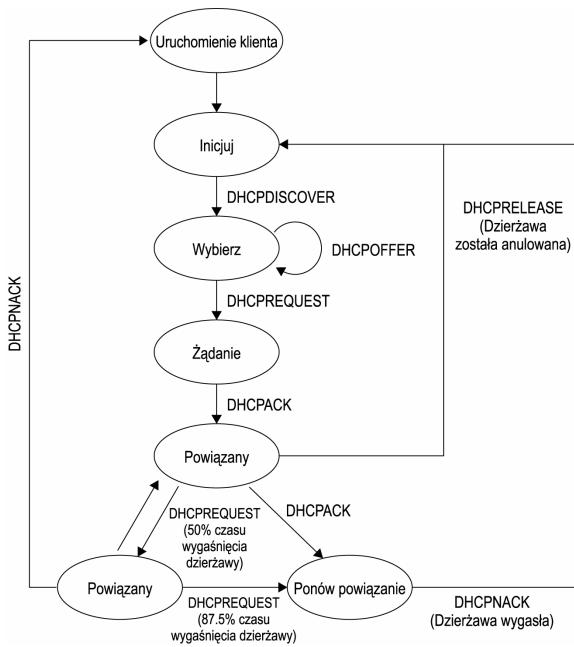
W dynamicznym srodowisku sieciowym dzierzawy sa ważne, poniewaz zapobiegaja zagarnianiu przez klienty adresów na dlugi czas. Po wygasniciu dzierzawy adres IP wraca do puli (zakresu) adresów serwera, z której adresy moga byc dzierzawione potrzebujacym ich klientom. Jesli jednak klient nadal uzywa adresu po uplynieciu terminu, serwer moze odnowic dzierzawe i pozwolic klientowi uzywac tego samego adresu. W niektórych przypadkach nieuzywane dzierzawy moga byc automatycznie zwracane do puli adresów.

Czas trwania dzierzawy zależy od sieci i wymogów klientów. Na przyklad, w sieci przedsiebiorstwa dzierzawa moze trwać dzien lub nawet tydzien, zależnie od wykonywanych zadań. Z drugiej strony, dzierzawy w kafejkach internetowych moza trwać zaledwie godzine. Z tego powodu specyfikacja DHCP nie zaleca określonego czasu dzierzawy. Jej długosc zależy od administratora sieci. DHCP pozwala również na dzierzawy na czas nieokreslony (np. jesli podamy w polu *Lease Duration* wartosc *0xffffffff*). Przyznawanie trwałych dzierzaw przypomina statyczny przydział adresów IP.

Proces dzierzawy DHCP

Proces dzierzawy DHCP, przedstawiony na rysunku 9.2, obejmuje następujące kroki:

Rysunek 9.2.
Proces dzierzawy
DHCP



- Klient zostaje uruchomiony i rozglasza w lokalnej podsieci tzw. *komunikat odkrycia* DHCP (*DHCPDISCOVER*). Ta faza nosi nazwę *stanu inicjacji*.



Jesli po drodze pomiędzy klientem i serwerem znajduje się ruter, komunikat rozgłoszenia może być przesyłany do innych podsieci. Do tego celu potrzebne są routery obsługujące protokół BOOTP.

- Wszystkie serwery, które otrzymały komunikat odkrycia i mogą wydzielować adres IP odpowiadają wysyłając *komunikat oferty* DHCP (*DHCPOFFER*). Komunikat ten zawiera adres IP i związane z nim dane konfiguracyjne.
- Klient może otrzymać wiele ofert dzierżawy, w zależności od liczby serwerów DHCP, które odpowiedziały na komunikat odkrycia. Klient wchodzi teraz w *stan wyboru*, w którym przegląda komunikaty ofert i wybiera jedną z nich.
- Klient wchodzi w *stan zadania* — wysyła do odpowiedniego serwera komunikat zadania (*DHCPREQUEST*), zadając konfiguracji zaoferowanej przez serwer.
- Serwer wysyła komunikat *pozytywnego potwierdzenia* (*DHCPACK*) komunikatu zadania, wysłanego przez klienta. Oprócz adresu IP i danych konfiguracyjnych komunikat ten zawiera informacje o dzierżawie.
- Klient po otrzymaniu potwierdzenia wchodzi w *stan powiązania*, w którym wydzielony adres IP jest związany z klientem, a klient staje się częścią sieci. W stanie powiązania klient używa trzech liczników czasu, które kontrolują wygasnięcie, odnowienie i ponowne nawiązanie dzierżawy.

7. W zaleznosci od ustawien licznika czasu wygasnienia, po uplywie 50% czasu dzierzawy — lub po jej wygasniciu — klient usiluje odnowic dzierzawe, wysylajac komunikat DHCPREQUEST do serwera, który adres wydzierzawil. Klient moze tez usilowac zakonczyc dzierzawe przed czasem, wysylajac komunikat zwolnienia (DHCPRELEASE).
8. Po wyslaniu do serwera komunikatu DHCPREQUEST, klient wchodzi w stan *odnowienia* i w tym stanie oczekuje na odpowiedz od serwera. Serwer moze w odpowiedzi albo przyjac zadanie (potwierdzajac przez *DHCPACK*), albo je odrzucic (*DCHPNACK*). W razie odrzucenia zadania klient zwalnia adres i wraca do stanu inicjacji.
9. Jesli klient nie otrzyma od serwera odpowiedzi w określonym czasie, to serwer zostaje uznany za wylaczony lub niedostepny. W tym przypadku klient po uplywie 87,5% czasu dzierzawy wchodzi w stan *ponowienia powiazania*. W tym stanie klient zaczyna ponownie rozglaszac komunikat DHCPREQUEST do wszystkich dostepnych serwerów DHCP.
10. Jesli klient otrzyma chocby jedna pozytywna odpowiedz, to wraca do stanu powiazania, natomiast jesli wszystkie serwery odpowiedza negatywnie, klient powróci do stanu inicjacji.

Strategia dzierzawy

Strategia dzierzawy okresla, jak dlugo ma trwac przecietna dzierzawa oraz czy odnawianie dzierzaw jest dopuszczalne, czy nie. Strategie dzierzawy moga sie jednak różnic dla poszczególnych kientów i grup kientów, w zaleznosci od ich wymagan. Administratorzy sieci ustanawiaja strategie dzierzawy dla calej sieci podczas wstepnej konfiguracji serwera DHCP.

Podczas określania strategii czas dzierzawy zdefiniowany przez administratora musi byc wystarczajaco krótki, aby pojedynczy klient przez zbyt dlugi okres nie zawlaszczal adresu IP, i by adresy IP wracaly do puli. Czas dzierzawy musi byc jednoczesnie na tyle dlugi, by kienty nie musialy regularnie rozglaszac zadan odnowienia dzierzaw adresów IP i zwiększac niepotrzebnie ruch w sieci. Ponadto, kienty moga wówczas w razie awarii lub braku dostepu do serwera DHCP poprawnie funkcjonowac, dopóki serwer nie zostanie przywrócony do uzytku.



Czas dzierzawy mozna ustalic szacunkowo na dwukrotne wartosc przecietnego czasu niedostepnosci serwerów DHCP w danej podsieci.

Administrator moze ustalic czas dzierzawy na określona liczbe tygodni, dni lub godzin, przez która klient ma prawo uzywac przyznanego adresu IP. Termin uplywu czasu dzierzawy adresu IP przyznanego z puli klientowi jest obliczany przez dodanie okresu dzierzawy do znacznika czasowego w komunikacie zadania klienta — DHCPREQUEST. Na przyklad, jesli przydzielimy klientowi adres IP na godzine (tzn. okres dzierzawy wynosi jedna godzina), a znacznik czasowy w komunikacie DHCPREQUEST to 20.06.2001, 14:43, wówczas dzierzawa adresu dla klienta wygasnie 20.06.2001 r. o 15:43.



Uwaga
Informacje o wygasnaniu dzierzawy dla klienta można w środowisku Windows NT 4.0 sprawdzić za pomocą narzędzia DHCP Manager. Każdy system operacyjny posiada własne narzędzie służące do tego celu.

Strategia dzierzawy definiuje również, czy klient może zadać odnowienia dzierzawy. Do ustawienia tej funkcji służy opcja negocjacji dzierzawy. Jeśli renegocjacja dzierzawy jest dozwolona przez strategię dzierzawy, klienci mogą wysyłać do serwera zadania odnowienia dzierzawy po upływie 50% okresu dzierzawy. Jednakże administrator sieci powinien przy planowaniu strategii dzierzawy wziąć pod uwagę jeden ważny fakt: jeśli liczba urządzeń w sieci przekracza całkowitą liczbę adresów IP w puli serwera, okres dzierzawy powinien być na tyle krótki, by urządzenia nie czekały nadmiernie dłużej na uzyskanie adresu IP. Jeśli w sieci nie ma niedoboru adresów IP, okresy dzierzawy powinny być wystarczająco długie, aby klienci nie musieli niepotrzebnie przerwywać trwających sesji w celu renegocjacji dzierzawy.



Ostrzeżenie
Hosty świadczące usługi sieciowe — na przykład serwery plików, poczty i drukowania — powinny posiadać adresy IP przydzielone rzecznicie, a nie dzierzawione na ustalony okres. A jeśli jest to z jakichś względów niemożliwe, hosty te powinny otrzymywać dzierzawy trwale, aby mogły świadczyć innym urządzeniom usługi w sposób nieprzerwany.

Opcje zakresu i serwera

Zakres DHCP (*DHCP scope*) oznacza pełną pulę poprawnych adresów IP, dostępnych dla wszystkich klientów DHCP w fizycznej podsieci. Każdy zakres DHCP posiada następujące właściwości:

- ♦ nazwę zakresu,
- ♦ pełny zakres adresów IP,
- ♦ maskę podsieci,
- ♦ okres dzierzawy,
- ♦ rezerwacje,
- ♦ opcje.



Opcje opisane są w punkcie „Opcje serwera DHCP” w dalszej części tego rozdziału.

Administratorzy sieci używają zakresów, aby dzielić fizyczne podsieci na większą liczbę podsieci logicznych. Serwer DHCP świadczy usługi DHCP dla każdej z tych logicznych podsieci oraz identyfikuje i przechowuje dane konfiguracyjne dla wszystkich klientów w danej podsieci. Klient w jednej logicznej podsieci może zazadac danych konfiguracyjnych również od serwerów w innych podsieciach logicznych.



Usluga DHCP w wersji Microsoftu pozwala dodatkowo administratorom sieci grupować kilka zakresów w *superzakres* (*superscope*), dzięki czemu można w jednym działaniu przydzielić strategie do wielu zakresów, o ile strategie te dla wszystkich zakresów są identyczne. Superzakresy mogą również służyć do rozwiązywania najczęstszych problemów serwerów DHCP, odczajając administratora sieci.

Od czasu do czasu grupa adresów w obrebie zakresu nie jest oferowana klientom serwera DHCP. Taka grupa nosi nazwe *zakresu wykluczenia*. Aby wykorzystac adresy IP z zakresu wykluczenia, administrator musi recznie skonfigurowac te adresy dla urzadzen sieciowych nie bedacych w stanie uzyt DHCP — na przyklad drukarek. Reszta adresów w zakresie (nie wykluczonych) tworzy *pule adresów zakresu*. Jedyne adresy z puli adresów zakresu sa oferowane klientom. Gdy host w sieci dzierzawi adres na stale, adres ten jest *zarezerwowany* dla klienta. Tylko określony host moze wydzierzawic adres dla siebie zarezerwowany.

Aby umozliwić klientom korzystanie z uslug serwera DHCP, musimy zdefiniowac i skonfigurowac zakres. Proces tworzenia zakresu DHCP przebiega wedlug nastepujacych krokow:

1. Utwórz zakres za pomocą odpowiedniego programu narzedziowego, zawartego w uzywanym systemie operacyjnym. Na przyklad, w Windows NT 4.0 i nowszych mozna uzyt narzeczia Microsoft DHCP Manager.

 Szczegółowa procedura tworzenia zakresów DHCP powinna byc opisana w dokumentacji uzywanego systemu operacyjnego.

2. W razie potrzeby zdefiniuj zakresy wykluczenia, wylaczajac z zakresu określone adresy. Adresy z zakresu wykluczenia powinny byc uzywane jedynie dla urzadzen sieciowych niezdolnych do automatycznego uzyskania adresu IP, na przyklad dla drukarek i modemów.
3. Utwórz rezerwacje dla urzadzen wymagajacych trwalej dzierzawy adresu z puli adresów. Do tych urzadzen naleza dostepne w sieci serwery różnych typów. Zarezerwowane adresy IP nalezy również przydzielic do ruterów.

 Rezerwacji nalezy dokonywac jedynie dla urzadzen sieciowych, które moga dynamicznie uzyskac adres IP dzieki zdolnosci do korzystania z uslugi DHCP.

4. Okresl okres dzierzawy. Wartosc domyslna wynosi trzy dni i w wiekszosci przypadkow jest do przyjecia. Administrator moze w miare potrzeb modyfikowac te wartosc.
5. Zdefiniuj niezbedne opcje zgodnie z wymaganiami.
6. Po pomyslnym utworzeniu i skonfigurowaniu, zakres nalezy aktywować, aby serwery DHCP mogly przetwarzac zadania dzierzaw i przydzielac dynamicznie adresy IP klientom.

Pakiet DHCP

Poniewaz protokoly DHCP i BOOTP sa do siebie bardzo podobne, format pakietu DHCP jest również bardzo zbliżony do formatu pakietu BOOTP. Jedno z pól pakietu DHCP jest jednak inaczej traktowane, a kolejne różni się od odpowiednika w BOOTP zawartoscia. Rysunek 9.3 przedstawia format pakietu DHCP.

Rysunek 9.3.
Format pakietu DHCP

OP	HTYPE	HLEN	Hops		
ID transakcji					
Sekundy		Flagi			
Adres IP klienta					
Twój adres IP					
Adres IP serwera					
Adres IP routera					
Adres sprzętowy klienta					
⋮					
Nazwa serwera					
⋮					
Nazwa pliku inicjującego					
⋮					
Opcje					
⋮					

Tylko dwa pola w pakiecie DHCP różnia się od pól w pakiecie BOOTP:

- ♦ *Flagi* — jest odpowiednikiem nie używanego pola w pakiecie BOOTP, w którym wszystkie bity pola mają wartość 0. W pakiecie DHCP wszystkie bity flagi mają wartość 0, z wyjątkiem pierwszego z lewej. Wartość tego (najbardziej znaczącego) bitu oznacza komunikat rozgłoszeniowy. Oznacza to, iż klient DHCP może od serwera DHCP zazadac wysłania odpowiedzi za pomocą komunikatu rozgłoszeniowego IP. Pole to ma długość 2 bajtów.
- ♦ *Opcje* — jest odpowiednikiem pola *Dane producenta* w komunikatach BOOTP. I podobnie jak w pakiecie BOOTP, pole to zawiera dodatkowe dane konfiguracyjne dostarczane przez producenta. Do informacji tych należą: okres dzierżawy, maska podsieci dla lokalnej sieci, adres IP serwera czasu, adres IP serwera domeny oraz rozmiar pliku inicjującego. Pole ma długość 64 bajtów.



Szczegółowy opis pozostałych pól znajduje się w punkcie „Zawartość pakietu BOOTP” we wcześniejszej części rozdziału.

Opcje serwera DHCP

Podczas operacji wydzielania adresów IP klientom DHCP, serwer DHCP może również przydzielac inne parametry konfiguracji wymagane przez klienta, zwane *opcjami DHCP* lub *opcjami serwera DHCP*. Należą do nich, na przykład, adres domyślnego routera lub bramy oraz adres serwera nazw. Podstawowe dalsze opcje konfigurowania klientów DHCP zostały wymienione poniżej:

- ♦ *Wypełnienie (kod opcji 0)* — dopelnia poniższe pola do granic pełnych słów.
- ♦ *Maska podsieci (kod opcji 1)* — przedstawia maskę podsieci dla danej podsieci fizycznej.
- ♦ *Przesunięcie czasu (kod opcji 2)* — oznacza czas UCT (*Universal Coordinated Time*) w sekundach.

- ♦ *Ruter (kod opcji 3)* — wymienia adresy IP wszystkich ruterów dostepnych w podsieci.
- ♦ *Serwery czasu (kod opcji 4)* — wymienia adresy IP wszystkich serwerów czasu dostepnych dla klienta.
- ♦ *Serwery nazw (kod opcji 5)* — wymienia adresy IP wszystkich serwerów nazw dostepnych dla klienta.
- ♦ *Serwery DNS (kod opcji 6)* — wymienia adresy IP wszystkich serwerów DNS dostepnych dla klienta.
- ♦ *Serwery dziennika (kod opcji 7)* — wymienia adresy IP wszystkich serwerów dziennika dostepnych dla klienta.
- ♦ *Serwery cookie (kod opcji 8)* — wymienia adresy IP wszystkich serwerów cookie dostepnych dla klienta.
- ♦ *Serwery LPR (kod opcji 9)* — wymienia adresy IP wszystkich serwerów drukarek wierszowych (*Line PRinter*) dostepnych dla klienta.
- ♦ *Serwery Impress (kod opcji 10)* — wymienia adresy IP wszystkich serwerów Imagen Impress dostepnych dla klienta.
- ♦ *Serwery lokalizacji zasobów (kod opcji 11)* — wymienia adresy IP wszystkich serwerów lokalizacji zasobów dostepnych dla klienta.
- ♦ *Nazwa hosta (kod opcji 12)* — przedstawia nazwe klienta, ktora moze miec dlugosc do 63 bajtow.
- ♦ *Rozmiar pliku inicjujacego (kod opcji 13)* — przedstawia rozmiar domyslnego pliku inicjujacego klienta.
- ♦ *Plik zrzutu zawartosci (kod opcji 14)* — przedstawia sciezke do pliku, do którego powinien zostac zrzucony obraz pamieci klienta w przypadku jego zalamania. Ten plik jest uzywany w sytuacjach, gdy domyslny plik inicjujacy klienta staje sie niedostepny z uwagi na awarie serwera.
- ♦ *Nazwa domeny (kod opcji 15)* — przedstawia nazwe domeny DNS, ktora powinna byc uzyta przez klienta do rozwiazania nazwy DNS hosta.
- ♦ *Serwer wymiany (kod opcji 16)* — przedstawia adres IP serwera wymiany dostepnego dla klienta.
- ♦ *Glowna sciezka dostepu (kod opcji 17)* — przedstawia sciezke do dysku systemowego klienta.
- ♦ *Sciezka do rozszerzen (kod opcji 18)* — oznacza plik zawierajacy informacje, podobnie jak pole danych producenta w komunikacie odpowiedzi BOOTP. Plik mozna pobrac za pomoca TFTP.

Trasowanie DHCP

Ruter obsługujacy protokół BOOTP potrafi zazwyczaj przesyłac również zadania i odpowiedzi DHCP pomiędzy podsieciami. W tym celu ruter musi obsługiwać usługę przekazywania DHCP i BOOTP. Dowolne urządzenie lub program, które potrafi prze-

sylac dane konfiguracyjne z jednej podsieci do drugiej, nazywane jest *agentem przekazujacym (relay agent)* — wobec tego ruter obsługujący protokół BOOTP można nazwać *agentem przekazującym DHCP/BOOTP*. Proces przesyłania zadań wygląda następująco:

1. Klient DHCP wysyła zadanie parametrów konfiguracyjnych przez port 68 TCP.
2. Agent przekazujący przechwytuje zadanie i rozpoznaje podsieć, do której zadanie trzeba przesłać.
3. W docelowej podsieci jeden lub kilka serwerów DHCP może „usłyszeć” rozgłoszenie i odpowiedzieć klientowi, podając dostępny adres IP.
4. Agent przekazujący DHCP/BOOTP przesyła odpowiedzi do klienta, który wybiera jedną z nich, wysyła komunikat zadań do odpowiedniego serwera i otrzymuje dzierżawę, która ponownie zostaje przekazana przez agenta przekazującego.

Rozdział 10.

Znajdowanie hostów w sieci

IP

W tym rozdziale:

- ◆ Wprowadzenie do Systemu nazw domen (DNS)
- ◆ Opis rozwiazywania nazw NetBIOS
- ◆ Wykorzystanie plików HOSTS i LMHOSTS
- ◆ Kolejność rozwiazywania nazw

Nazwa jest ważna częścią składową tożsamości i łatwiej ją zapamiętać niż liczby. Niestety rozdział omówią szczegółowo procesy przetwarzania przyjaznych dla użytkownika nazw, których używamy w aplikacjach, na liczby przyjazne dla komputera, takie jak adres IP. Potrzebujemy nazw. Proszę sobie wyobrazić, co działyby się, gdybysmy nie mogli używać nazw w przeglądarkach WWW — musielibyśmy znać adres IP każdej odwiedzanej witryny. Lista ulubionych adresów również wyglądałaby inaczej. Bieżący rozdział zaglebia się w szczegóły rozwiazywania nazw — procesu, dzięki któremu Internet jest przyjazny dla użytkowników; zajmuje się nazwami hostów i nazwami usług NetBIOS oraz ich rolą w ułatwianiu łączności z określonym komputerem. Wiele aplikacji, na przykład poczta elektroniczna, FTP, Telnet, przeglądarki WWW i przeglądarki grup dyskusyjnych, wymaga do swojego funkcjonowania nazw. Wobec tego zrozumienie procesu rozwiazywania nazw jest solidną podstawą do rozwiazywania problemów z różnorodnymi aplikacjami.

Przegląd nazw hostów

Nazwa hosta jest po prostu etykieta (aliasem) adresu IP. Kazde urządzenie w sieci IP posiada nazwę hosta. Korzystanie z nazw hostów przynosi następujące korzyści:

- ◆ Nazwy hostów są łatwiejsze do zapamiętania od adresów IP.
- ◆ Nazwy hostów dają stabilność w mobilnym środowisku komputerowym. Klienta możemy przenosić z jednej podsieci do drugiej, używając dla niego innego adresu IP w każdej sieci, natomiast nazwa hosta pozostaje niezmieniona w każdej sieci.

- ♦ Pojedynczy komputer moze posiadac szereg nazw hosta, z których zadna nie musi zgadzac sie z nazwa NetBIOS.
- ♦ Nazwy hostów moga byc przechowywane lokalnie w pliku HOSTS, lub — dla globalnego dostepu — w bazie danych serwera DNS.

Ktos powiedzial kiedys, ze nazwy hostów sa „meska sprawa”. W poczatkach sieci IP byla sobie grupa osób (mezczyzn), którzy chcieli wykorzystywac etykiety dla adresów IP, poniewaz adresy IP sa trudne do zapamietania. Uzywali oni nazw kolorów w roli nazw hostów: niebieski, zielony, czerwony, czarny, bialy, brazowy i pomaranczowy. Szybko jednak zdali sobie sprawe, iz zeszli na manowce, gdy zabraklo kolorów. Cóż, nie uzywali takich barw, jak brudny róż, zimne północne indygo, czy cieply bez baha-ma. To byli *prawdziwi* mezczyzni: rozpoznawali jedynie siedem kolorów, wlacznie z czernia i biela.

Historyjka dosc sympatyczna; w rzeczywistosci to Peggy Karp, pracowniczka naukowa MITRE Corp. z Washington D.C. po raz pierwszy zaproponowala korzystanie z nazw hostów w RFC 226 z 20 wrzesnia 1971. Peggy zaproponowala, by przypisac czteroliterowe kody popularnym serwerom uslugi Telnet, aby uproscic procedury dostepu. Tak narodzil sie mechanizm rozwiazywania nazw.

Wprawdzie nazwy hostów ulegly znaczacym zmianom od roku 1971, lecz pierwotny pomysl pozostal. Ludziom latwiej zapamietac nazwy niz adresy IP. Niemal wszyscy wiedza, jak znalezc witryne WWW Microsoftu za pomoca jej adresu, a ile osób zna jej adres IP? Jesli Czytelnik zna ten adres, swiadczy to tylko o nadmiarze wolnego czasu.

Nazwy hostów byly poczatkowo zaimplementowane w postaci pliku o nazwie *hosts.txt* na serwerze nazw hostów w SRI-NIC. Plik ten byl codziennie pobierany przez kazdego klienta za pomoca FTP. Z uplywem czasu proces dystrybucji pliku *hosts.txt* stal sie problematyczny z szeregu powodów:

- ♦ Proces pobierania pliku zajmował zbyt wiele przepustowosci laczy.
- ♦ Pobieranie pliku hosts.txt raz na dzien to zbyt rzadko, aby miec aktualne informacje.
- ♦ Blyskawiczny rozwój Internetu w olbrzymim stopniu zwiększył nakladы pracy na utrzymanie pliku.
- ♦ Zwiększenie objetosci pliku *hosts.txt* samo w sobie zaosztyrzyło problemy z czasem pobierania i obciążeniem laczy.
- ♦ Charakter bazy klientów zaczął się zmieniać. Zamiast współużytkować duże komputery, organizacje zaczęły laczyć stacje robocze w sieci lokalne. Organizacje te same zarządzaly własna przestrzenią nazw, lecz nadal musiały czekać za każdym razem na aktualizacje pliku hosts.txt przez SRI-NIC.

Pomysły na metody ustalania nazw hostów były proponowane w licznych dokumentach RFC, poczawszy od RFC 799, 819 i 830. Chociaż metoda implementowania „przestrzeni nazw” w każdym RFC była inna, wszystkie trzy podkresłyły potrzebę stosowania hierarchicznej bazy danych. W listopadzie 1997 System Nazw Domen (*Domain Name System*) został zdefiniowany w RFC 1034 i 1035. Od tamtej chwili ponad 20 dokumentów RFC uscisliły, zmodyfikowały definicje, lub powołały się na DNS.

Podstawowe nazwy hostów

Podstawowa nazwa hosta służy do opisania komputera lub innego urządzenia w sieci. Przykładami podstawowych nazw hostów mogą być: *slowpoke*, *ou812*, czy też wyjątkowo popularna *sparky*. Aplikacje korzystające z interfejsu gniazd (*Sockets*) lub Win-Sock API używają nazw hostów w roli końcowych punktów łączenia. Aby można było wykorzystać nazwę hosta, musi ona istnieć w pliku hostów lokalnego klienta, lub też w serwerze DNS, do którego klient ma dostęp.

Nazwy hostów mogą składać się najwyżej z 256 znaków, przy czym wielkość liter może grać rolę lub nie (zależnie od stosowanego systemu operacyjnego). Dla większości nowszych systemów operacyjnych Windows wielkość liter w nazwach hostów nie ma znaczenia, lecz niektóre systemy uniwersalne mogą wciąż rozróżnić małe i duże litery.



Rozdział 7. zawiera szczegółowe informacje o interfejsie Sockets i aplikacjach WinSock.

Pelne zlozone nazwy domen

Jesli dana organizacja posiada nazwy domeny DNS (*Domain Name System*), to nazwa ta może posłużyć do ustalenia podstawowej nazwy hosta. Nazwa domeny DNS w połączeniu z nazwą hosta tworzy pełna złożona nazwa domeny (FQDN — *Fully Qualified Domain Name*).

Załóżmy, że nazwa hosta naszego komputera brzmi *goofy*, zaś nasza zarejestrowana domena to *cartoon.com*. W takim przypadku FQDN komputera brzmi *goofy.cartoon.com*. Może istnieć wiele komputerów o nazwie *goofy*, lecz tylko jeden *goofy.cartoon.com*. Nazwa domeny służy do ustalenia nazwy hosta, dzięki czemu nazwa ta jest unikatowa, nawet jeśli istnieją inne komputery o nazwie *goofy*. Dzięki FQDN można używać nazw hostów na skali globalnej.

Rysunek 10.1 przedstawia okno konfiguracji nazwy hosta NT i nazwy domeny DNS, w którym nazwa hosta brzmi *goofy*, zaś nazwa domeny DNS to *cartoon.com*.



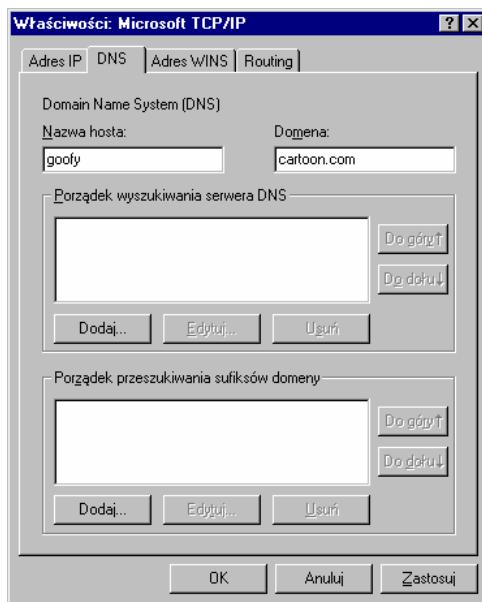
Aby utworzyć pełna złożona nazwę domeny (FQDN), należy dodać nazwę swojej domeny DNS do nazwy hosta.

Nazwy kanoniczne i aliasy

Czasami wygodnie jest odwołać się do hosta poprzez inną nazwę, a nie jego nazwę DNS. Przyjętym w Internecie standardem dla serwerów WWW jest nazwa hosta *www*. Nazwa *www* zwykle nie jest w ogóle nazwa hosta. Jest to *alias* (etykieta) rzeczywistej nazwy hosta, wskazujący na ten sam komputer pod inną nazwą.

Jedna z korzyści stosowania aliasów jest możliwość ukrycia nazwy hosta serwera przed klientem. Gdy serwer trzeba zastąpić innym, operacje takie można za pomocą aliasów ukryć przed obsługiwanyimi przez sień klientami. Aliasy pozwalają na nadmiarowość — kilka lub więcej serwerów może reagować na tę samą nazwę.

Rysunek 10.1.
Konfiguracja nazwy hosta i domeny



Na przykład, posiadamy rewelacyjny serwer WWW, na którym mieści się witryna www.tcpbible.bk. Prawdziwa nazwa hosta serwera WWW brzmi *barney.tcpbible.bk*, lecz używamy rekordu zasobu nazwy kanonicznej (CNAME) usługi DNS, by kierować wszystkie zadania dotyczące *www.tcpbible.bk* do hosta *barney*. Od czasu do czasu wyłącza się komputer *barney* w celu konserwacji, lecz zastępujemy go wówczas komputerem *wilma*, który podobnie jak *barney* posiada alias *www*. Nikt tego nie zauważa. Gdy witryna WWW jest intensywnie użytkowana, oba hosty pozostają załączone.



Alias jest po prostu przydomkiem dla nazwy hosta.

Aliases mogą być implementowane za pomocą pliku hostów lub w usłudze DNS, lecz metody te są odmienne.

„Webster’s Encyclopedic Dictionary” definiuje pojęcie „kanoniczny” jako „zgodny z, lub nakazany przez prawo kanoniczne”. Istnieją pewne niejasności w definicji nazwy kanonicznej. Nazwa kanoniczna jest zgodna z regulami i jest pełna złożona nazwa domeny (FQDN). Sprawca całego zamieszania jest rekord zasobu CNAME (nazwy kanonicznej). Niektórzy myślą, iż rekord CNAME oznacza nazwę kanoniczną, podczas gdy w rzeczywistości jedynie wskazuje na nazwę kanoniczną. Może wydawać się dziwne, iż rekord CNAME jest jedynym niekanonicznym rekordem w usłudze DNS.

W poniższym fragmencie pliku strefy DNS ostatnie dwa rekordy są rekordami CNAME. Właścicielem rekordu jest alias, mieszczący się po lewej stronie rekordu. Te części użytkownika widzi i wpisuje w swojej przeglądarce WWW. Alias możemy traktować jak „przewisko”. Nazwa kanoniczna mieści się po prawej stronie rekordu i wskazuje na FQDN docelowego hosta.

Kilka słów o standardzie WWW

Nie istnieje tak naprawdę żaden przekonujący powód techniczny, by stosować nazwę „www” dla witryn WWW, poza łatwością zapamiętania (oraz oczywistym skrótem od *World Wide Web*). Oceniając sprawę po fakcie możemy przypuszczać, iż wyglądałoby to inaczej, gdyby organizacja Internet Engineering Task Force wyobraziła sobie spikerów telewizyjnych i radiowych usiłujących wymówić adresy URL w języku angielskim, w którym skrót „www” ma 9 sylab! W porównaniu ze słowami „FTP”, „Telnet” czy „NNTP”, łatwymi do wymówienia, wymowa „www” jest wyzwaniem. W angielskim alfabetie jest tylko jedna litera, której wymowa składa się z wiecej niż jednej sylaby — i w chwili obecnej używamy jej trzykrotnie na poczatku nazw milionów witryn WWW na całej planecie. Czy decyzje podejmowane przez komitety nie są fantastyczne?

```
happy    IN  A      192.168.0.4
dopey    IN  A      192.168.0.3
sleepy   IN  A      192.168.0.2
grumpy   IN  A      192.168.0.1
dp       IN  CNAME  dopey.efs.ca
www     IN  CNAME  happy.efs.ca
```

Rekord nazwy kanonicznej kojarzy przydomek z nazwą FQDN.

Lokalny plik HOSTS

Przed wprowadzeniem usługi DNS istniała tylko jedna metoda rozwiązywania nazw innych hostów — plik *HOSTS*. Wiele uniksowych systemów operacyjnych używa nazwy pliku *hosts.txt*. Systemy operacyjne Microsoftu używają nazwy *HOSTS* bez rozszerzenia. Zarówno systemy operacyjne Microsoftu, jak i uniksowe składają plik hostów w folderze *drivers\etc* (*drivers/etc*). Plik *HOSTS* jest tablica służąca do sprawdzania odwzorowania nazw hostów na adres IP, utrzymywana lokalnie w każdym komputerze.

Format pliku HOSTS

Każdy wiersz lokalnego pliku *HOSTS* zawiera odwzorowanie adresu IP na nazwę hosta. Typowa zawartość pliku *HOSTS* może wyglądać tak:

```
172.16.23.91  bugs.cartoon.com          # serwer pomocniczy
192.168.2.123 goofy.cartoon.com        # serwer WWW
192.168.2.33  tweety.cartoon.com tweetie  # serwer pocztowy
# 192.168.2.22 sylvester.cartoon.com    # stary serwer pocztowy
172.16.23.42  bugs.cartoon.com          # serwer pomocniczy
127.0.0.1     localhost
```

Pierwszy wiersz przypisuje nazwę *bugs.cartoon.com* do adresu 172.16.23.91. Drugi wiersz przypisuje *goofy.cartoon.com* do 192.168.2.123. Symbol # oznacza, że pozostała część wiersza uznawana jest za komentarz, wobec czego *goofy* jest serwerem WWW. W trzecim wierszu serwer pocztowy, *tweety.cartoon.com*, posiada jednocześnie alias *tweetie*. Cały wiersz może być uznany za komentarz, jeśli użyjemy znaku #, jak na przykład w wierszu czwartym, zawierającym nieużywany już stary serwer pocztowy *sylvester*. Ostatnia pozycja w pliku jest domyślny wpis adresu lokata innego hosta.



Nalezy uwazac podczas edycji pliku HOSTS w srodowisku Windows. Jesli w roli edytora uzywany jest Notatnik, to trzeba upewnic sie, czy plik zostanie zapisany jako *hosts*, bez rozszerzenia, w katalogu *drivers\etc* (dla systemów Windows NT i 2000) lub katalogu głównym systemu Windows dla Windows 95 i 98. Plik HOSTS zapisany pod niewlasciwa nazwa lub w niewlasciwym miejscu bedzie ignorowany podczas rozwiazywania nazw.

Rozwiazywanie nazw

Podczas rozwiazywania nazw za pomoca pliku HOSTS, plik ten jest analizowany wiersz po wierszu od poczatku do konca. W przedstawionym powyzej przykladzie kodu pliku HOSTS pojawia sie pewien problem. Drugi wpis dla *bugs.cartoon.com* (wiersz piaty) nie zostanie nigdy uzyty, poniewaz proces przegladania pliku od poczatku **zatrzymuje** sie na pierwszym pasujacym wpisie. Jesli to drugi wpis dla komputera *bugs* jest poprawny, to host uzywajacy danego pliku HOSTS bedzie kierowac ruch przeznaczony dla komputera *bugs.cartoon.com* do uzytkownika adresu 172.16.23.91, niezale znie od jego nazwy hosta.



Aby sprawdzic poprawnosc wpisow w pliku HOSTS, nalezy zawsze sprawdzac wprowadzone do niego nazwy hostow poleceniem ping.

Rozwiazanie nazwy hosta za pomoca pliku hostow obejmuje nastepujace kroki:

1. Nazwa hosta zostaje wpisana w aplikacji lub wierszu polecen — na przyklad, URL witryny WWW w przegladarce lub nazwa serwera FTP w kliencie FTP.
2. System operacyjny sprawdza, czy nazwa docelowego hosta zgadza sie z nazwa hosta skonfigurowana lokalnie. Jesli tak, to lokalny adres IP hosta zostaje wykorzystany do lacznosci w warstwie Internetu.
3. Jesli nazwa nie zgadza sie z lokalna nazwa hosta, lokalny plik HOSTS jest analizowany od poczatku w dól. Gdy adres zostanie znaleziony w pliku, posluzy do nawiazania lacznosci w warstwie Internetu.
4. Jesli nazwa hosta nie zostanie znaleziona w pliku HOSTS, to uzytkownik otrzymuje komunikat o bledzie i przetwarzanie zostaje zakonczone.



W przypadku braku pewności, czy w docelowym srodowisku rozróżniana jest wielkosc liter w nazwach, nalezy w tym samym wierszu pliku HOSTS zawrzec różne odmiany nazwy danego hosta.

Wykorzystanie uslugi DNS do rozwiazywania nazw hostów

System nazw domen (DNS) jest trójskładnikowym systemem, który został opracowany, aby rozszerzyc zakres stosowania rozwiazywania nazw, przy jednoczesnej minimalizacji nakladów pracy na codzienna obsluge przestrzeni nazw i jej dystrybucje pomiedzy różne jednostki. DNS posiada nastepujace skladniki:

- ♦ serwer nazw,

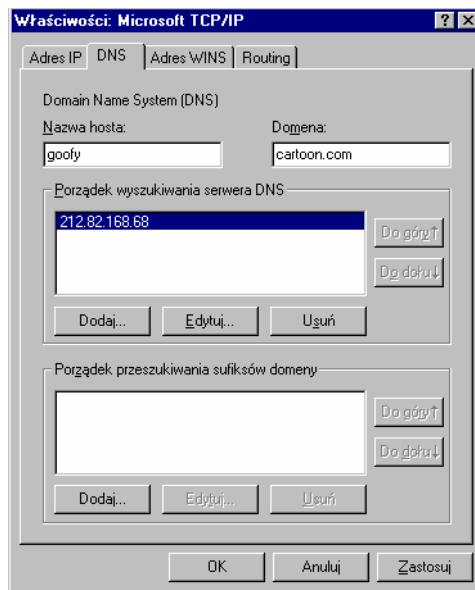
- ◆ resolwer (klient),
- ◆ przestrzen nazw.

DNS jest rozproszona baza danych, która służy TCP/IP do rozwiązywania nazw hostów na adresy IP (lub odwrotnie) dla każdego komputera na świecie, bez konieczności stosowania lokalnych plików HOSTS. Jeśli klient posiada skonfigurowany adres IP serwera DNS, to może wysyłać zadania rozwiązywania nazw do tego serwera, zamiast korzystać z własnego lokalnego pliku HOSTS. Klienci DNS zazwyczaj potrafią powtarzać zapytania kierowane do przeciwnego serwera nazw w ustalonych odstępach — piecioskundowych lub zbliżonych.

Rysunek 10.2 przedstawia konfigurację DNS-u dla typowego klienta firmy Microsoft, w której podano określony adres IP serwera DNS.

Rysunek 10.2.

Konfiguracja klienta DNS-u



Rozwiązywanie nazwy hosta za pomocą serwera DNS obejmuje następujące kroki:

1. Nazwa hosta zostaje wpisana w aplikacji gniazd (*sockets*) lub w wierszu poleceń — na przykład, URL witryny WWW w przeglądarce lub nazwa serwera FTP w kliencie FTP.
2. System operacyjny sprawdza, czy nazwa docelowego hosta zgadza się z nazwą hosta skonfigurowaną lokalnie. Jeśli tak, to lokalny adres IP hosta zostaje wykorzystany dołączenia w warstwie Internetu.
3. Jeśli nazwa nie zgadza się z nazwą lokalnego hosta, to wysyła on zadanie rozwiązywania nazwy hosta docelowego do znanego sobie serwera DNS. Resolwer może powtarzać zadania w odstępach 5, 10, 20 i 40 sekund. Jeśli serwer DNS odpowie na zadanie, to odpowiedź (zwykle adres IP) posłuży do nawiązania połączenia w warstwie Internetu.

- 4.** Gdy serwer DNS nie udzieli odpowiedzi, wówczas użytkownik otrzymuje komunikat o błędzie i przetwarzanie zostaje zakończone.



W momencie, gdy TCP/IP dysponuje adresem IP docelowego hosta, dane aplikacji są przesyłane w dół stosu, aby umożliwić trasowanie do punktu przeznaczenia. Proces ten jest zawsze taki sam, niezależnie od tego, co zachodzi w wyższych warstwach. Pomyślnie rozwiązane nazwy są w końcu trasowane do miejsca przeznaczenia przez warstwę internetową TCP/IP. Wiecej informacji o trasowaniu zawiera rozdział 5.

Czym jest domena?

Domena DNS to po prostu wezel w przestrzeni nazw. Domena ta składa się ze swojej pierwotnej nazwy i wszystkich domen położonych poniżej. Można również myśleć o domenie DNS jak o tożsamości zbiorowej. Każda nazwa organizacji w Internecie musi być unikatowa, co osiąga się za pomocą zarejestrowanych nazw domen. Popularność systemów operacyjnych Windows zaowocowała powstaniem innego typu domen: domen Windows NT. Nie mają one żadnego związku z domenami DNS, czego nie można jednak powiedzieć o domenach Windows 2000. Domeny DNS i domeny Windows 2000 mają najczęściej wspólne nazwy. Domeny Windows 2000 do funkcjonowania wymagają usługi DNS, w przeciwienstwie do domen NT.

Serwery nazw

Serwer nazw usługi DNS to komputer z uruchomiona aplikacja serwera DNS. Serwer ten może składować dane pliku strefy lokalnie lub w pamięci. Serwer DNS odpowiada na zgłoszane przez klienty zadania rozwiązania nazw, usiłując znaleźć te nazwy (oraz związane z nimi adres IP) w przestrzeni nazw. Serwer nazw wykonuje również na plikach stref operacje związane z zarządzaniem bazą danych — na przykład, aktualizacje rekordów zasobów i transfery stref.

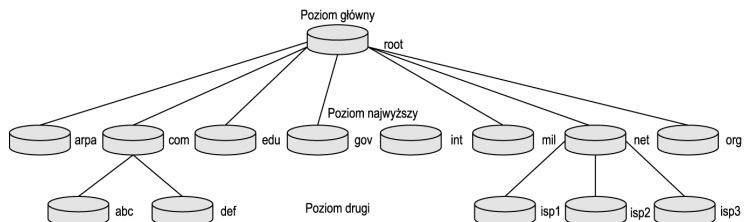
Resolwery

Resolwer to klient usługi DNS. Resolwerami mogą być stacje robocze lub serwery TCP/IP, lecz jedne i drugie muszą być skonfigurowane tak, by wysyłać zadania rozwiązania nazw pod adres IP przynajmniej jednego serwera DNS. Wielkość komputerów biurkowych w środowisku DNS gra role resolwerów. Rysunek 10.2 przedstawia wymagana konfigurację resolwera.

Przestrzeń nazw

Przestrzeń nazw DNS składa się z nienazwanego wezła głównego (*unnamed root*) oraz rozchodzących się z niego gałęzi zwanych domenami. DNS wykorzystuje organizację hierarchiczną, aby utrzymać informacje o przynależności domen. Domena główna, oznaczana przez kropkę (.), jest nadzweczna dla wszystkich pozostałych domen. Zarówno domena główna, jak i ogólne domeny najwyższego poziomu (*top-level domains*) są zarządzane przez mieszczącej się w USA organizacji Internet Corporation for Assigned Names and Numbers (ICANN). Pozostałe domeny najwyższego poziomu są zarządzane międzynarodowo. Domeny najwyższego poziomu są rozmieszczone organizacyjnie, funkcjonalnie i geograficznie poniżej domeny głównej w sposób pokazany na rysunku 10.3.

Rysunek 10.3.
Przestrzeń nazw domen
DNS



Przestrzeń nazw DNS funkcjonuje jak grupa odrebień zarządzanych baz danych, mieszczących się w różnych systemach komputerowych. Każda z tych baz jest w stanie wyszukiwać wpisy w pozostałych bazach danych i korzystać z nich. Przestrzeń nazw składa się z dużej liczby serwerów nazw, zwanych systemami, połączonymi ze sobą w związkach typu nadrzedny-podrzedny. Każdy system może być odpowiedzialny jedynie za niewielki obszar przestrzeni nazw, lecz w odpowiedzi na zadania klientów mogą być zwieracane nazwy hostów z innych systemów.

Serwery poziomu głównego

Serwery nazw domeny głównej (*root servers*) zawierają wpisy dla serwerów nazw wszystkich domen najwyższej poziomu. Zadaniem serwerów poziomu głównego jest znajdowanie serwerów nazw w domenach najwyższej poziomu i rozwiązywanie ich nazw dla innych serwerów nazw. Te z kolei zawsze korzystają z serwera poziomu głównego jako punktu wyjścia dla wyszukiwania nazw DNS. Odwołanie do serwera poziomu głównego jest „najgorszym przypadkiem” procesu rozwiązywania nazwy, ponieważ serwery nazw poszczególnych domen odpytują serwery poziomu głównego tylko wtedy, gdy nie mogą znaleźć odpowiedzi gdziekolwiek indziej. Kazdy serwer nazw w publicznym Internecie posiada tzw. *plik wskazówek głównych* (*root hints*), inaczej *plik podreczny*, który zawiera liste serwerów poziomu głównego. Rząd USA zarządza serwerami poziomu głównego poprzez prywatnego zleceniodobiorca (ICANN). Serwery te są aktualizowane codziennie.

Domeny poziomu głównego

Domeny poziomu głównego (TLD — *Top Level Domain*) służą do podziału organizacji według typu lub funkcji. Same organizacje zwykle nie rejestrują TLD. Domeny poziomu głównego służą do klasyfikacji typu organizacji — na przykład, edukacyjnej, komercyjnej lub niedochodowej.

Osiem popularnych domen poziomu głównego można podzielić na ogólne i specjalne przeznaczenia. Domeny ogólne to:

- ◆ *.com* — dla przedsiębiorstw komercyjnych
- ◆ *.net* — dla sieci
- ◆ *.org* — dla organizacji typu niedochodowego

Domeny specjalnego przeznaczenia to:

- ◆ *.edu* — dla instytucji edukacyjnych
- ◆ *.gov* — dla organizacji rządowych

- ♦ *.mil* — wojskowe
- ♦ *.int* — dla organizacji utworzonych przez umowy międzynarodowe
- ♦ *.arpa* — dla wyszukiwania wstecz (rozwiązywania adresu IP na nazwe hosta)

Oprócz tych osmiu TLD kazdy kraj posiada domene najwyższego poziomu o dwuliterowej nazwie, reprezentujacej kod nazwy kraju (np. *.pl* dla Polski, *.ca* dla Kanady, *.tw* dla Tajwanu i tak dalej), co zwiększa liczbę uzywanych TLD do ponad dwustu! W Kanadzie przestrzenia nazw domeny najwyższego poziomu *.ca* zarządza Canadian Internet Registration Authority.



Dodatek na koncu ksiazki zawiera liste obecnie uzywanych domen najwyższego poziomu.

Zadaniem wspólnych domen najwyższego poziomu jest wskazywanie domen drugiego poziomu. Na przykład, serwer nazw domeny *.com* potrafi znalezc serwer nazw dla kazdej poddomeny *.com*. Kazdy serwer domeny najwyższego poziomu *.com* posiada baze danych, która zawiera wpisy dla serwerów nazw domen drugiego poziomu oraz dla wszelkich hostów, mogacych znajdowac sie w samej domenie najwyższego poziomu. Na rysunku 10.3 serwer nazw dla *.com* posiada wpisy dla serwerów nazw w domenach *abc.com* i *def.com*, dzieki czemu moze odsyiac inne serwery nazw do tych wpisów w swojej strefie.



Miarodajna liste TLD mozna znalezc pod adresem www.alldomains.com/alltlds.html.

W chwili obecnej w ICANN — niedochodowej organizacji nadzorujacej przestrzen nazw domen — trwaja prace rozwojowe pod nazwa New TLD Program. Projekt ten obejmuje propozycje siedmiu dodatkowych domen najwyższego poziomu. Organizacja ICANN oglosila 16 listopada 2000 roku utworzenie nowych TLD, których wprowadzenie w życie zostało jednak zaplanowane na grudzien 2001. W czasie, gdy ksiazka ta byla pisana, niektóre instytucje przyjmowały juz wstępne rejestracje nowych domen w nowych TLD. Siedem nowych domen najwyższego poziomu to:

- ♦ *.aero* — dla przemyslu transportu lotniczego
- ♦ *.biz* — dla biznesu
- ♦ *.coop* — dla spółdzielni
- ♦ *.info* — bez ograniczen wykorzystania
- ♦ *.museum* — dla muzeów
- ♦ *.name* — dla osób prywatnych
- ♦ *.pro* — dla profesjonalistów: lekarzy, prawników i księgowych

Wiekszosć domen najwyższego poziomu zarządza ICANN, z wyjątkiem domen krajowych, które zarządzane są lokalnie.

Domeny drugiego poziomu

W przypadku domen drugiego poziomu zaczyna się liczyć fakt, iż przestrzeń nazw DNS ma charakter rozproszony. Domeny te nie są zarządzane przez ICANN. W domenach drugiego poziomu organizacje mogą zarządzać własną przestrzeń nazw. Domeny te mogą zawierać serwery, hosty i domeny niższych poziomów, zwane *poddomenami*. Każda domena drugiego poziomu zawiera informacje o hostach, serwerach nazw, serwerach poczty elektronicznej i serwerach ftp, znajdujących się w tej domenie.



Uwaga
Jednym z wymogów przy rejestracji domeny drugiego poziomu jest udostępnienie w Internecie dwóch serwerów DNS. Pełny zestaw wymagań można znaleźć w organizacji akredytowanej przez ICANN do rejestrowania domen lub u dostawcy usług internetowych.

Strefy w obrębie przestrzeni nazw

Strefa jest ciągim obszarem przestrzeni nazw, za który serwer nazw jest odpowiedzialny. Strefa może być niewielkim zakątkiem domeny DNS, może też rozciągać się na wiele domen. Można również posłużyć się strefą do zdefiniowania części przestrzeni nazw, która administrator powinien zarządzać. Do niedawna zarządzanie serwerem DNS wiązało się z recznym wprowadzaniem i utrzymywaniem wszystkich rekordów w strefie, wobec czego przemysły podział odpowiedzialności był kluczem do dobrze zarządzanych domen DNS. Administratorzy DNS-u odpowiedzialni za zbyt duże strefy częściej popełniają błędy i wolniej aktualizują strefy, z uwagi na zakres pracy. Strefy DNS mogą być podstawowe lub wtórne oraz mogą służyć do wyszukiwania w przód lub wstecz.



RFC 2136 definiuje protokół dynamicznych aktualizacji DNS-u (*Dynamic DNS*), który pozwala obsługującym go klientom automatycznie aktualizować informacje DNS-u w pliku strefy. Ta nowa możliwość obsługiwana jest przez BIND 8.2.3 i serwer DNS Windows 2000.

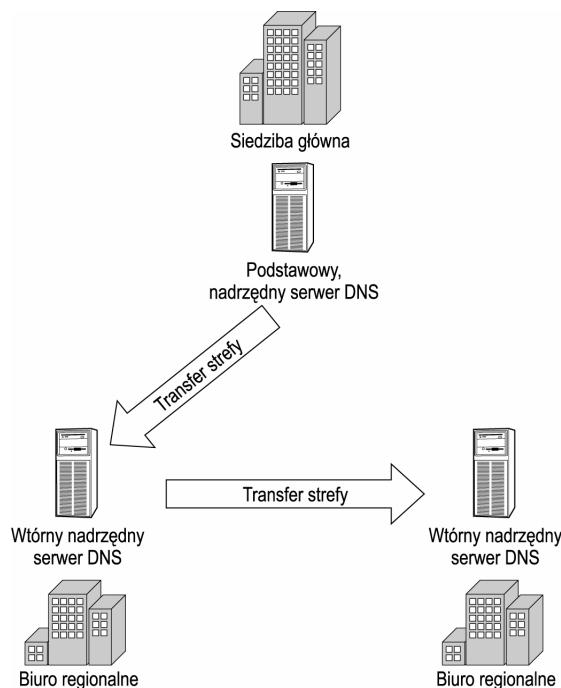
Strefy podstawowe

Strefa podstawowa składa się z rekordów zasobów i danych konfiguracyjnych, które wprowadza się i utrzymuje w serwerze DNS w lokalnie składowanym pliku. Serwery DNS zawierające podstawowy plik strefy wymagają obsługi przez wykwalifikowanych pracowników, związanej z zarządzaniem bieżącymi zmianami i uzupełnieniami bazy danych strefy. Dla każdej strefy tylko jeden serwer DNS może być podstawowym.

Strefy wtórne

Strefa wtórna składa się z rekordów zasobów i danych konfiguracyjnych, które normalnie przesyłane są w chwili uruchomienia serwera oraz w regularnych odstępach czasu innego serwera nazw DNS, który określamy jako nadziedny (master). Rysunek 10.4 pokazuje, iż nadziedny serwer DNS nie musi być serwerem podstawowym, aby można było dokonać transferu strefy. Strefy wtórne są bardzo przydatne w oddalonych lokalizacjach, które potrzebują serwera DNS, lecz nie „życzą” sobie odpowiedzialności związanej z zarządzaniem nim.

Rysunek 10.4.
Transfery stref DNS



Organizacje potrzebują zwykle wiecej serwerów nazw niż stref. Zalóżmy, iż przedsiębiorstwo posiada trzy oddziały, lecz tylko jedna domena DNS i jednego administratora DNS-u. Zainstalowanie serwera DNS w każdym oddziale wymagałoby wyszkolonego pracownika zajmującego się zarządzaniem, zas przekierowanie wszystkich klientów z wszystkich oddziałów do pojedynczego serwera DNS w jednej lokalizacji przeciążłoby ten serwer. Wykorzystanie podstawowych i wtórnego stref DNS może złagodzić ten problem.

Rysunek 10.4 przedstawia sytuację, w której jeden z trzech oddziałów instaluje strefę podstawową i zatrudnia administratora do zarządzania nią. Dwa pozostałe oddziały instalują strefy wtórne. Ponieważ transfer danych z serwera nadziedzkiego zapewnia strefę wtórną, nie jest dla niej wymagana codzienna obsługa. Dzięki takiej prostej implementacji usługi DNS każdy oddział posiada lokalnie dostępne usługi DNS przy minimalnych nakładach pracy na zarządzanie.

Jedna z silnych stron DNS-u jest wszechstronność. Istnieje mnóstwo możliwości tworzenia różnych struktur DNS-u.

Strefy wyszukiwania w przód

Strefy wyszukiwania w przód służą do rozwiązywania nazw FQDN na adresy IP. Za pomocą takiej strefy klient DNS-u może znaleźć adres IP dla danej nazwy hosta, co jest najczęściej spotykana forma zapytan w DNS-ie. Gdy wprowadzimy adres URL w przeglądarce WWW, protokół TCP/IP w naszym komputerze sformułuje zapytanie o wyszukiwanie w przód, aby rozwiązać podany URL na adres IP. Jeśli odpowiedź będzie pomyslna, to w przeglądarce pojawi się witryna WWW, jeśli nie — komunikat o błędzie.

Serwery DNS bez stref

Niekotere serwery DNS w ogóle nie zawieraja stref. Nosza one nazwe serwerów buforujacych (*caching-only*). Serwery takie przekazuja wszystkie zapytania klientow do innych serwerów, lecz „zapamietuja” (buforu) odpowiedzi na potrzeby ewentualnych przyszlych zapytan innych klientow o ten sam adres. Serwery buforujace przechowuja typowo buforowane wpisy przez przynajmniej godzine. Serwery takie wykorzystywane sa w miejscach, gdzie wymagane jest rozwiazywanie nazw, lecz ruch sieciowy zwiada-ny z transferami stref jest nie do przyjecia. Wyobrazmy sobie sytuacje, w której 20 klientow z sieci obejmujacej 10 000 uzytkownikow mieści sie w odleglej lokalizacji, polaczonej z głównym osrodkiem przez laczne WAN o ograniczonej przepustowoscia. W takim scenariuszu regularne transfery stref obejmujace wszystkie 10 000 rekordow przeciazylby laczne WAN. Przesyłanie zapytan 20 klientow poprzez serwer buforujacy daje w takim przypadku mozliwa do przyjecia szybkosc rozwiazywania nazw i cal-kojcie eliminuje transfery stref.

W niektórych systemach DNS opartych na Uniksie pliki wyszukiwania wstecz posiadaja forme *db.strefa*. Wiekosc stref wyszukiwania w przód DNS-u opartego na Windows uzywa nazw plików *strefa.dns*. Wobec tego, jesli posiadamy domene *cartoon.com* i implementujemy strefe wyszukiwania w przód w Uniksie, to mo zemy spodziewac sie, iz plik strefy bedzie nosil nazwe *db.cartoon.com*. Ta sama strefa w DNS-ie systemu Windows bedzie zawarta w pliku *cartoon.com.dns*. Strefy wyszukiwania w przód moga byc podstawowe lub wtórne.

Strefy wyszukiwania wstecz

W sytuacji, gdy klient posiada juz adres IP docelowego komputera, lecz chce przetlumaczyć ten adres na FQDN, musi wyslac zapytanie do serwera DNS zawierajacego strefe wyszukiwania wstecz. Strefy takie w odpowiedzi na zapytania zawierajace adresy IP zwracaja nazwy hostów. Dla wstecznego — „odwrotnego” wyszukiwania adresów zarezerwowana jest domena najwyzszego poziomu *.arpa*. Strefy wyszukiwania wstecz w systemach uniksowych zapisywane sa w plikach o nazwach w postaci *db.adres*, gdzie *adres*adres.in-addr.arpa.dns, gdzie *adres*db.142.204. Wersja tego samego pliku dla systemu Windows bedzie nazywac sie *204.142.in-addr.arpa*. Jak widac, 204.142 jest odwrotnie zapisanym faktycznym adresem IP 142.204.

Serwer WWW rejestrujacy adresy IP wszystkich gości moze skorzystac z wyszukiwania wstecz, aby zanotowac w dziennikach zdarzen FQDN zamiast IP.

Wiekosc serwerów DNS obecnie posiada zarówno strefy wyszukiwania w przód, jak i wstecz, podstawowe lub wtórne.

Tworzenie pliku strefy

Plik strefy sklada sie z danych nagłówka i rekordów zasobów. Dane zawarte w nagłówku określaja zachowanie strefy, natomiast rekordy zasobów sklada sie na baze danych DNS. Ponizszy listing jest przykładem typowego pliku strefy; dla wygody czytelnika dodano numery wierszy. Jak widac, komentarze oddzielone sa średnikami. Wpisy w pliku strefy zajmujace wiecej niz jeden wiersz objete sa nawiasami, jak w przypadku wierszy od 1. do 6.

```
1 @ IN SOA tweety.cartoon.com. dnsadmin.cartoon.com. (
2           20010420 ; numer seryjny
3           36000    ; interwal odswiezania (1 godzina)
4           600     ; interwal ponawiania (10 minut)
5           86400    ; interwal wygasania (1 doba)
6           3600)   ; minimalny TTL (1 godzina)
```

Kazdy plik strefy zaczyna sie od rekordu tego samego typu: rekordu poczatku pełnomocnictwa (SOA — *Start of Authority*). Wiersz 1. zawiera typ rekordu (SOA) i nazwe hosta serwera autorytatywnego (w tym przypadku *tweety.cartoon.com*), a nastepnie adres e-mail administratora odpowiedzialnego za serwer. Prosze zwrócic uwage, iz w tym adresie zamiast powszechnie dzis stosowanego symbolu @ uzyta jest kropka. Z administratorem mozna skontaktowac sie pod adresem *administrator@cartoon.com*. Nastepne wiersze (od 2. do 6.) zawieraja dane konfiguracyjne strefy.

Wiersz 2. podaje wersje pliku DNS. Liczba ta musi byc aktualizowana po kazdej modyfikacji pliku. Wtórne serwery nazw uzywaja pola wersji, aby ustalic, czy posiadaja aktualna wersje strefy, czy nie. W powyzszym przykladzie numer seryjny odpowiada dacie modyfikacji; inni administratorzy moga jednak stosowac inne metody indeksacji.

Wiersz 3. podaje interwal odswiezania (w sekundach). Zgodnie z tym ustawieniem wtórne serwery nazw beda zadac transferu strefy co godzine.

Wiersz 4. zawiera interwal ponawiania. W przypadku niepowodzenia zadania transferu strefy serwer wtórny bedzie czekac podany czas przed ponowieniem zadania transferu. W tym przypadku interwal ponawiania wynosi 10 minut.

Wiersz 5. podaje interwal wygasania, przez który serwer wtórny bedzie usiloval pobrac strefe od nadrzednego, nadal uzywajac posiadanego pliku strefy. Po uplywie okresu wygasania serwer wtórny odrzuci strefe i zacznie funkcjonowac jedynie jako buforujacy, dopóki nie bedzie mozna przeslac nowych danych strefy.

Wiersz 6. zawiera minimalny czas zycia (Min. TTL — *Time to Live*). Zapytania, rozwiazane dzieki komunikacji z innymi serwerami nazw, przechowywane sa w pamieci dla innych resolwerów, które moglyby ich potrzebowac, lecz jedynie przez godzine. Je-sli resolwer zapyta o te sama nazwe, o która inny resolwer pytal piec minut wczesniej, to serwer nazw moze zwrócic buforowany wpis, zamiast konsultowac sie z innymi serwerami nazw w celu znalezienia odpowiedzi.

```
7 @      IN  NS  tweety.cartoon.com.
8 @      IN  NS  sylvester.cartoon.com.
9 tweety  IN  A   192.168.1.7
10 sylvester  IN  A   192.168.1.8
```

Wiersze 7. i 8. identyfikuja hosty *tweety* i *sylvester* jako serwery nazw dla tej strefy. Typ rekordu NS oznacza serwer nazw (*name server*).

Wiersze 9. i 10. sa rekordami hostów (inaczej adresu), które wiazą (sklejają) nazwy hostów *tweety* i *sylvester* z odpowiadajacymi im adresami IP. Rekordy te nazywane sa czasami rekordami sklejajacymi (*glue record*).

```
11 localhost  IN  A  127.0.0.1
```

Wiersz 11. pozwala w usłudze DNS funkcjonowac zapytaniom DNS do lokalnego hosta, nawet jeśli klient nie posiada pliku HOSTS.

```
12 @      IN  MX  10   tom
13 @      IN  MX  15   jerry
14 tom    IN  A   192.168.1.17
15 jerry  IN  A   192.168.1.18
```

Wiersze od 12. do 15. identyfikują hosty *tom* i *jerry* w roli serwerów pocztowych. Proszę zauważyc, iż *tom* jest preferowanym komputerem wymieniającym poczty, ponieważ jego wartość preferencji (10) jest niższa. Host *jerry* będzie używany tylko w przypadku, gdy *tom* będzie niedostępny. Wiersze 14. i 15. wiążą nazwy hostów *tom* i *jerry* z ich adresami IP.

```
16 bugs    IN  A   192.168.1.135
17 elmer   IN  A   192.168.1.11
```

Wiersze 16. i 17. są rekordami hostów dla komputerów *bugs* i *elmer*, wiązającymi je z odpowiednimi adresami IP. Ponieważ po nazwach hostów nie występuje kropka (.), do nazw dodawany jest bezpośrednio domyślny sufiks domeny, wobec czego *bugs* staje się *bugs.cartoon.com*, zaś *elmer* — *elmer.cartoon.com*, podobnie jak pozostałe powyzsze wpisy dla hostów *weetey*, *sylvester*, *tom* i *jerry*. Gdybyśmy chcieli w pliku strefy zawsze wpisy dla hostów z innych domen, moglibyśmy skorzystać w rekordach typu A (rekordach hostów) z ich nazw FQDN, zakończonych kropką.

```
18 ftp     IN  CNAME bugs
19 www    IN  CNAME elmer
```

Wiersze 18. i 19. są rekordami nazw kanonicznych (CNAME), które pozwalały na odwołanie do hostów *bugs* i *elmer* za pomocą przydomków. Przydomkiem hosta *bugs* jest *ftp.cartoon.com*, zaś hosta *elmer* — *www.cartoon.com*. Ponieważ hosty te są już skojarzone z adresami IP w wierszach 16. i 17., posiadamy wszystkie informacje potrzebne, aby odwołać się do hostów za pomocą ich przydomków.

Gdy rozszerzony zapyta serwer *weetey* o adres *www.cartoon.com*, *weetey* wyszuka *www* w pliku strefy i ustali, iż nazwa wskazuje na komputer *elmer*. Następnie *weetey* wyszuka hosta *elmer* w pliku strefy i znajdzie jego adres IP 192.168.1.11 (w wierszu 17.). Do rozszerzona zostanie więc zwrotnie jako odpowiedź adres 192.168.1.11.

Gdyby nasza witryna WWW mieściła się na kilku serwerach, każdy z nich mógłby posiadać rekord CNAME z aliasem *www*. DNS może równoważyć obciążenie pomiędzy wszystkimi aliasami *www*. Ta popularna technika w DNS-ie nosi nazwę metody kanalowej (*round robin*).

Gdyby powyższy przykład był plikiem strefy wyszukiwania wstecz, nagłówek pozostałby niezmieniony, lecz rekordy byłyby inne. Strefy wyszukiwania wstecz używają FQDN jako obiektów-lisci. Do znalezienia nazw hostów na podstawie danych adresów IP służy rekordy wskazników (PTR).

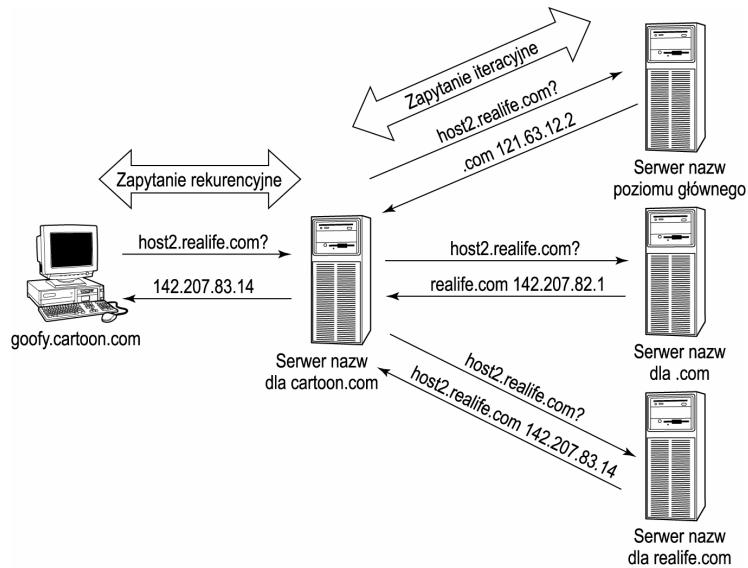
Omówiliśmy tworzenie pliku strefy w podstawowym zakresie, lecz pokazaliśmy proces ręcznego tworzenia pliku strefy, nadal powszechnie stosowany w wielu środowiskach uniwersyteckich. Wiele aplikacji — jak np. serwery DNS w systemach Windows — posiada interfejsy graficzne, które przetwarzają wprowadzane graficznie dane na wpisy w pliku strefy, dzięki czemu administrator nie musi zajmować się bezpośrednio tworzeniem i utrzymywaniem plików stref.

Zapytania iteracyjne i rekurencyjne

Resolwery wysylaja do serwerów nazw zapytania rekurencyjne. Okreslenie „rekurencyjny” odnosi sie do faktu, iz zapytanie moze przechodzic kolejno do serwerów nazw w całej globalnej przestrzeni nazw, co czasami okreslane jest terminem „kroczenie po drzewie” (*walking the tree*). Relacje miedzy resolwerem i serwerem nazw wymagaja zwrocenia jednej z dwóch mozliwych odpowiedzi na zapytanie o nazwe: (1) odpowiedzi, lub (2) komunikatu o bledzie stwierdzajacego, ze szukany host nie istnieje. Serwer nazw nie moze skierowac resolwera do innego serwera nazw. Musi znalezc odpowiedz lub stwierdzic, ze odpowiedz nie istnieje.

Gdy serwer nazw otrzymuje zapytanie od resolwera, w pierwszej kolejnosci sprawdza pamiec podreczna nazw, a nastepnie plik strefy. Jesli w tych dwóch miejscach nie jest w stanie znalezc wpisu dla nazwy lub adresu IP szukanego hosta, to serwer nazw wykorzystuje plik wskazówek głównych (inaczej plik podreczny) w polaczeniu z zapytaniami iteracyjnymi, aby przemieszczajac sie po drzewie domen znalezc odpowiedz dla resolwera. Jesli w tej samej przestrzeni nazw odpowiedz istnieje, to zostanie ona w tym procesie znaleziona; moze to jednak zajac troche czasu. Rysunek 10.5 pokazuje, jak dzialaja zapytania iteracyjne i rekurencyjne:

Rysunek 10.5.
*Zapytania iteracyjne
i rekurencyjne*



1. *Goofy.cartoon.com* odpytuje swój serwer nazw o adres IP hosta *host2.realife.com*.
2. Serwer nazw domeny *cartoon.com* sprawdza, czy posiada pelnomocnictwa (plik strefy) dla *realife.com*. Nie posiada ich, a poza tym nie ma tez odpowiedzi w pamieci podrecznej, wiec serwer formuluje zapytanie iteracyjne i wysyla je do jednego z serwerów poziomu glownego, wymienionych w pliku podrecznym.
3. Serwer poziomu glownego w odpowiedzi zwraca najlepsze z posiadanych informacji. Poniewaz serwer ten z nazwy *host2.realife.com* zna jedynie czesc *.com*, wobec tego odpowiada na zapytanie zwracajac adres IP serwera nazw domeny *.com*, którego adres IP posiada w pliku strefy glownej.

4. Serwer nazw domeny *cartoon.com* ponownie przekazuje zadanie hosta *goofy* o *host2.realife.com*, lecz tym razem do serwera nazw domeny *.com*.
5. Serwer nazw domeny *.com* odpowiada najlepiej, jak potrafi. Ponieważ jego plik strefy zawiera jedynie wpis dla serwera nazw domeny *realife.com*, więc może odesłać do serwera nazw domeny *cartoon.com* jedynie ten adres.
6. Ponownie serwer nazw dla *cartoon.com* wysyła zapytanie hosta *goofy*, lecz tym razem do serwera nazw posiadającego pełnomocnictwa dla domeny *realife.com*.
7. Serwer nazw domeny *realife.com* odpowiada adresem IP hosta *host2.realife.com*.
8. Serwer nazw dla *cartoon.com* odpowiada na zapytanie hosta *goofy*, podając adres IP dla *host2.realife.com*.

Z punktu widzenia resolwera, obsługującego go serwer nazw zna wszystkie adresy IP i nazwy hostów w globalnej przestrzeni nazw. Resolwer wysyła pytanie i otrzymuje odpowiedź za pomocą zapytania rekurencyjnego.

Z drugiej strony, serwery nazw posiadają zdolność wskazywania na siebie nawzajem na podstawie najlepszych posiadanych informacji. Takie odpytywanie iteracyjne może wymagaćłącności z wieloma serwerami nazw w celu odpowiedzi na pojedyncze zadanie resolwera.

Konfiguracja DNS-u z wykorzystaniem programu BIND

Konfiguracja DNS-u za pomocą oprogramowania Berkeley Internet Name Daemon (BIND) opiera się na istnieniu *pliku rozruchowego (boot file)*, który zawiera początkowe parametry startowe dla serwera DNS. Serwery DNS systemów Windows nie potrzebują pliku rozruchowego, ponieważ dla nich dane konfiguracyjne DNS-u są przechowywane w Rejestrze. Gdy jednak chcemy przenieść istniejącą konfigurację z programu BIND do DNS-u systemu Windows, możemy bez trudu wykorzystać plik rozruchowy.

Plik rozruchowy musi nosić nazwę *boot* i zawierać określone polecenia i opcje. Polecenia te kontrolują sposób, w jaki usługa DNS jest uruchamiana. Pliki rozruchowe programu BIND w wersjach 4 i 8 mają odmienne style. W tym punkcie zajmiemy się plikiem rozruchowym programu BIND 4.

Poniżej przedstawiony został prosty plik rozruchowy DNS. Numery wierszy zostały wstawione jedynie dla wygody czytelnika. Polecenia pliku rozruchowego zaczynają się od początku wiersza i nie są poprzedzane znakami spacji.

```
1. cache      c:\winnt\system32\dns\cache.dns
2. primary    cartoon.com  cartoon.com.dns
3. secondary   realife.com  192.168.1.22  db.realife.com
4. forwarder  192.168.1.47 192.168.1.48
5. option      no recursion
```

Polecenie *cache* w wierszu 1. określa nazwę i położenie pliku podręcznego. Plik podręczny (inaczej plik wskazówek głównych) służy do znajdowania serwerów nazw dla domeny głównej. Wiersz 2. określa, iż serwer posiada pełnomocnictwo dla strefy podstawowej — *cartoon.com*, której dane składowane są w pliku strefy o nazwie *cartoon.com.dns*. Wiersz 3. określa, iż serwer nazw posiada także pełnomocnictwo dla strefy

wtórnej *realife.com* oraz podaje nazwe lokalnego pliku sluzacego do buforowania danych tej strefy. Wiersz 4. podaje liste serwerów nazw, które zgadzaja sie rozwiazywac zapytania rekurencyjne w imieniu naszego serwera nazw. Polecenie option w wierszu 5. okresla, iz serwer nazw powinien do innych serwerów nazw wysylac zapytania nie-rekurencyjne.

Serwery nazw BIND nie dysponuja inną mozliwoscia konfiguracji, poza ta z wykorzystaniem pliku rozruchowego. Serwery nazw Windows moga byc konfigurowane za pomocą danych zawartych w Rejestrze lub pliku rozruchowym. Różne wersje serwerów DNS pod Windows stosuja odmienne metody konfiguracji pozwalajace na uruchomienie z pliku rozruchowego. Jedna z tych metod jest DNS-owy wpis w Rejestrze Boot-Method. Wartosc 1 oznacza uruchamianie z pliku, zas 2 kaze skorzystac z Rejestru.

Konfiguracja Windows 2000

DNS w Windows 2000 posiada kilka dodatkowych funkcji, takich jak dynamiczny DNS, rekordy uslug, przyrostowe transfery stref i strefy zintegrowane z Active Directory. Opcje te trzeba odpowiednio skonfigurowac, aby usluga DNS poprawnie funkcjonowala.

Dynamiczny DNS

Wymagajacym najwiekszych nakladów pracy i najbardziej podatnym na pomyłki aspektem zarzadzania serwerem DNS jest reczne wprowadzanie kazdego rekordu zasobu. Dynamiczny DNS (DDNS) rozwiazuje ten problem, pozwalajac komputerom klienckim na wprowadzanie przy uruchomieniu własnych rekordów zasobów do stref DNS. Klienty Windows 2000 uzywaja standardu DDNS. Dane innych klientów nadal trzeba wprowadzac recznie; jesli jednak zaimplementujemy serwer DHCP w Windows 2000, uzyskamy funkcjonalosc DDNS dla takich klientów. Standard DDNS jest opisany w RFC 2136. Dynamiczny DNS mozna konfigurowac dla poszczególnych stref. Rysunek 10.6 pokazuje, iz strefa *cartoon.com* korzysta z DDNS-u.

Przyrostowe transfery stref

Kazdy plik strefy posiada pole numeru seryjnego w rekordzie SOA, sluzace do kontroli wersji. Poniewaz istnieje tylko jeden numer seryjny dla calego pliku, nie mozna za pomocą tej pojedynczej wartosci sledzic zmian poszczególnych rekordów. W przeszlosci kazdy transfer strefy obejmował pelny zbiór wszystkich rekordów w danej strefie, niezaleznie od liczby rekordów, które ulegly zmianie. Taki transfer strefy nosi nazwe AXFR.

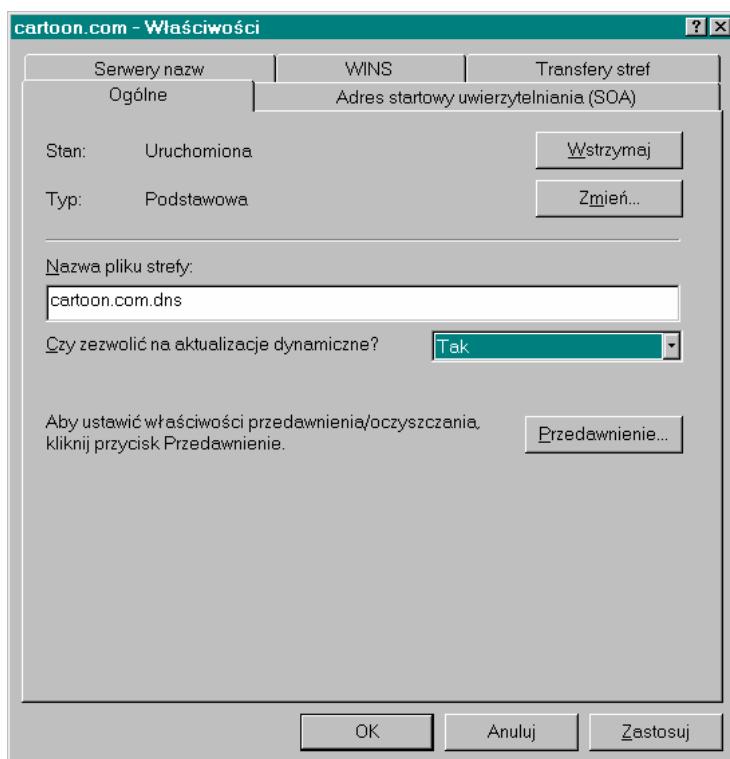
RFC 1995 opisuje nowy typ transferu stref, w którym wysylane sa za kazdym razem tylko rekordy zasobów, które uległy zmianie. DNS Windows 2000 oraz BIND w wersji 8 obsluguja przyrostowe transfery stref zgodne z RFC 1995.

Strefy zintegrowane z Active Directory

Windows 2000 obsluguje strefy zintegrowane z Active Directory oraz standardowe strefy podstawowe i wtórne. DNS Windows 2000 mozna uruchomic w dowolnym serwerze Windows 2000, lecz strefy zintegrowane z Active Directory dostepne sa jedynie w kontrolerach domen Windows 2000. Plik strefy jest w ich przypadku przechowywany w Active Directory, zamiast, typowo, w `%systemroot%\system32\dns`.

Rysunek 10.6.

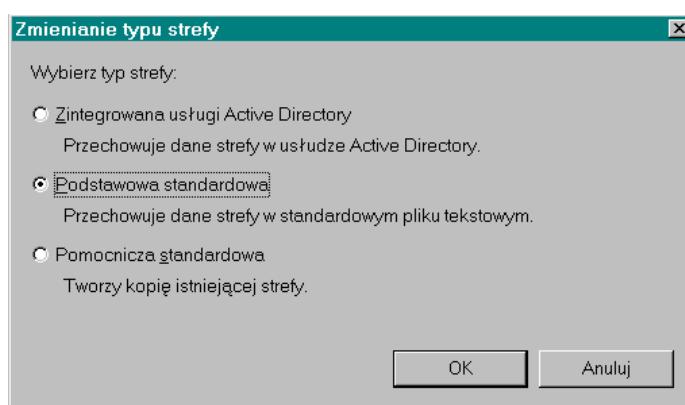
Konfiguracja dynamicznego DNS-u



Typ strefy mozna dowolnie przelaczac pomiędzy podstawowa, wtórna i zintegrowana z Active Directory. Rysunek 10.7 pokazuje, jak mozna tego dokonac w DNS-ie Windows 2000.

Rysunek 10.7.

Typy stref DNS w Windows 2000



Strefy zintegrowane z AD maja dwie zalety w porównaniu ze strefami standardowymi:

1. Wyeliminowana zostaje potrzeba transferów stref DNS pomiędzy komputerami Windows 2000, poniewaz dane Active Directory sa i tak replikowane do wszystkich kontrolerów domeny w danej domenie. Strefy zintegrowane z Active Directory obsługują transfery do serwerów wtórnego BIND.

2. Dynamiczne aktualizacje DNS-u mozna zabezpieczyc. Kazdy rekord zasobu w strefie zintegrowanej z AD moze byc chroniony przez liste kontrolna dostepu (ACL — *Access Control List*), w ktorej mozna ustalic, kto ma prawo aktualizowac lub usunac dany rekord.

Rozwiazywanie nazw NetBIOS

Oprócz mnóstwa aplikacji napisanych dla interfejsu gniazd (*sockets*), wykorzystywanych w Internecie, istnieja aplikacje NetBIOS. Edytory tekstu i arkusze kalkulacyjne dla dowolnego systemu Windows sa najprawdopodobniej aplikacjami NetBIOS. W istocie, dawno, dawno temu, systemy operacyjne Windows posiadalry jedynie interfejs aplikacji NetBIOS. Aby korzystac z aplikacji internetowych — na przyklad przegladarek WWW, trzeba bylo dodac interfejs Sockets. Autorzy pamietaja czasy, kiedy korzystali z Trumpeet Winsock uzupełniajacego pakiet protokolów TCP/IP, który musieli *kupic* dla Windows 3.1, aby uzyskac dostep do Internetu.

Chociaz protokół NetBIOS został oryginalnie opracowany w 1983 roku dla firmy IBM, kazdy system operacyjny Microsoftu, z którym kiedykolwiek pracowalismy, uzywal NetBIOS-u. NetBIOS byl odpowiedzialny za wprowadzenie grup roboczych w Windows for Workgroups. NetBIOS jest zasadniczo implementowany zarowno jako protokół warstwy sesji, jak i zlaczce programowe aplikacji (API — *Application Programming Interface*).

Aplikacje NetBIOS w roli punktów koncowych lacznosci korzystaja z przyjaznych dla uzytkownika nazw, zamiast adresów IP. TCP/IP odpowiada nastepnie za przekształcenie przyjaznych dla uzytkownika nazw NetBIOS na przyjazne dla sieci adresy IP, w podobny sposob, jak czyni to DNS. Rozwiazywanie nazw NetBIOS jest opisane w RFC 1001 i 1002, które zostały napisane w marcu 1987 roku, jako szczególowe zalecenia implementacji protokolu NetBIOS w srodowisku TCP/IP.

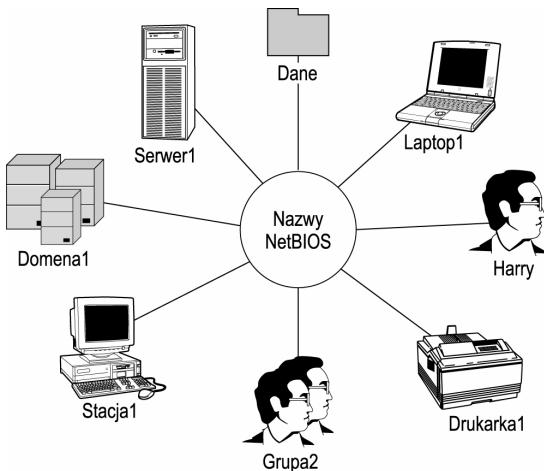
Nazwy NetBIOS — co to jest?

Nazwy NetBIOS moga reprezentowac różne obiekty: uzytkowników, komputery, grupy robocze, uslugi NT, a nawet domeny. Wszystkie te obiekty maja jedna wspólna ceche — moga sluzyc jako punkty koncowe lacznosci. Aplikacje NetBIOS korzystaja na potrzeby komunikacji z nazw NetBIOS.

Komputer w systemie Windows posiada nazwe komputera — gdy udostepnia w sieci foldery, nazwa NetBIOS komputera sluzy do znalezienia listy udostepnionych folderów, dostepnych w danym komputerze Windows. Drukarki sieciowe sa identyfikowane w sieci poprzez swoje nazwy NetBIOS. Domeny Windows NT i grupy robocze sa nazwami NetBIOS. Wszystkie „sieciowe” narzedzia protokolu SMB (*Server Message Block* — blok komunikatów serwera), Eksplorator i Menedzer plików w komunikacji uzywaja nazw NetBIOS.

Rysunek 10.8 pokazuje, iz prawie wszystko w sieci Windows posiada nazwy NetBIOS. Hosty uniksowe nie uzywaja nazw NetBIOS, poniewaz nie korzystaja w lacznosci z protokolom SMB. Jednakze wiele systemow operacyjnych Unix obsluguje aplikacje rozszerzajaca o nazwie SAMBA, która pozwala na lacznosc z wykorzystaniem nazw NetBIOS i protokolu SMB.

Rysunek 10.8.
Nazwy NetBIOS



Nazwy NetBIOS mają następujące właściwości:

- ◆ Nie rozróżniają wielkości liter.
- ◆ Długość do 15 znaków.
- ◆ Gdy opisują usługę NT, są dopełniane do 15 znaków i uzupełniane liczbą szesnastkową.
- ◆ Są alfanumeryczne — nie dopuszczają spacji, kropek i symboli.

Składniki sieciowe Microsoftu

W większości systemów operacyjnych role grane w łączności przez różne składniki struktury sieciowej są wyraźnie zdefiniowane. Serwery nasłuchują wywołań ze strony klientów, lecz same nie zgłoszają żadnych zasobów. Klienci zadają dostępu do zasobów serwera, lecz same nie rozmawiają swoich możliwości. W systemach Unix i NetWare serwer jest tylko serwerem a klient tylko klientem — lecz w systemach operacyjnych Microsoftu jest inaczej.

Przy tworzeniu Windows for Workgroups Microsoft wprowadził inny model komunikacji: grupę roboczą. Wszyscy członkowie grupy roboczej mogą w tym samym komputerze użytkować zarówno składnik serwera, jak i klienta jednocześnie. Ten model komunikacji jest nadal popularny. Choć w większości przypadków domeny zastąpiły grupy robocze, model łączności przetrwał: każdy komputer w sieci jest zdolny do pełnienia funkcji serwera i klienta.

Najbardziej podstawowe składniki sieciowe Microsoftu obejmują serwer i stację roboczą (inne to składniki przesyłania wiadomości i przeglądania). Składniki te są implementowane jako usługi lub dodatkowe pliki, w zależności od używanego systemu operacyjnego. Niektóre składniki sieciowe Windows NT są widoczne dla systemu operacyjnego jako systemy plików, takie jak sieciowe usługi. Usługi takie korzystają z nazw NetBIOS użytkowników, komputerów, grup lub domen ze specjalnymi identyfikatorami liczbowymi, aby móc komunikować się i być identyfikowane w sieci. Wobec tego, choć komputer NetBIOS posiada tylko jedno nazwe komputera (w przeciwieństwie do systemów unik-

sowych, które mogą mieć wiele nazw hostów), komputery NetBIOS mogą używać szersci i wiecej różnych nazw w celu identyfikacji swoich składników w sieci. Tabela 10.1 zawiera listę najczęściej spotykanych usług NetBIOS.

Tabela 10.1. Najczęściej spotykane usługi NetBIOS

Nazwa usługi	Nazwa NetBIOS	Sufiks liczbowy	Typ
Stacja robocza	Nazwa komputera	00	UNIQUE
Serwer	Nazwa komputera	20	UNIQUE
Przeglądarka główna	Nazwa domeny	1B	UNIQUE
Wymuszenie elekcji	Grupa robocza lub domena	1E	UNIQUE
Kontroler domeny	Domena	1C	GROUP
Przesyłanie wiadomości	Użytkownik, komputer lub domena	03	UNIQUE

Podczas uruchomienia systemu nazwy NetBIOS są rejestrowane w lokalnej usłudze nazewniczej NetBIOS oraz, opcjonalnie, w skonfigurowanym serwerze nazw NetBIOS. Lokalna usługa nazewnicza NetBIOS lub skonfigurowany serwer nazw zapewniają, iż nie wystąpią dwie jednakowe unikatowe nazwy NetBIOS. Usługa ta może wysyłać komunikaty o błędach i odmawiać rejestracji w przypadku prób rejestracji dwóch identycznych nazw.

Rozwiązywanie nazw NetBIOS przed Windows 2000

Przed wprowadzeniem systemu Windows 2000 sieci Microsoft Networking używały NetBIOS-u. Aby ustawić łączność przez sieć, należy przekształcić nazwę NetBIOS na adres IP. Do rozwiązywania nazw NetBIOS na adresy IP używane są powszechnie poniższe metody:

Rozglaszanie

Najbardziej podstawowa metoda rozwiązywania nazw NetBIOS jest rozglaszanie nazwy NetBIOS pozadanego hosta docelowego z nadzieją, iż odpowie, zwracając swój adres IP. Metoda ta jest powszechnie stosowana w małych środowiskach (jednosegmentowych), ponieważ routery zasadniczo odrzucają rozmowy NetBIOS.

WINS

RFC 1002 definiuje serwer nazw NetBIOS (NBNS — *NetBIOS Name Server*) — aplikacje, która może przyjmować rejestracje nazw NetBIOS z wielu segmentów, pozwalać klientom w celu rejestracji korzystać z ruchu kierowanego zamiast rozmów. Windows Internet Name Server (WINS) jest serwerem nazw NetBIOS Microsoftu. Klienci usługi WINS rejestrują swoje nazwy NetBIOS podczas uruchamiania, mogą też odkrywać bazę danych WINS o rozwiązywanie innych zarejestrowanych nazw.

LMHOSTS

LMHOSTS jest prostym plikiem tekstowym, mieszczącym się w folderze *drivers\etc* w klientach NetBIOS-u. Plik ten zawiera odwzorowania adresów IP na nazwy NetBIOS dla komputerów w innych segmentach. Plik LMHOSTS stosowany jest podobnie jak plik HOSTS, z ta różnicą, iż LMHOSTS służy jedynie dla nazw zdalnych. Podczas rozwiązywania nazw plik LMHOSTS jest przeszukiwany od początku w dół. Gdy zaś wiersza podwojone wpisy, użyty będzie jedynie wpis położony wyżej; należy więc przetestować każdy nowy wpis za pomocą polecenia net use.

HOSTS

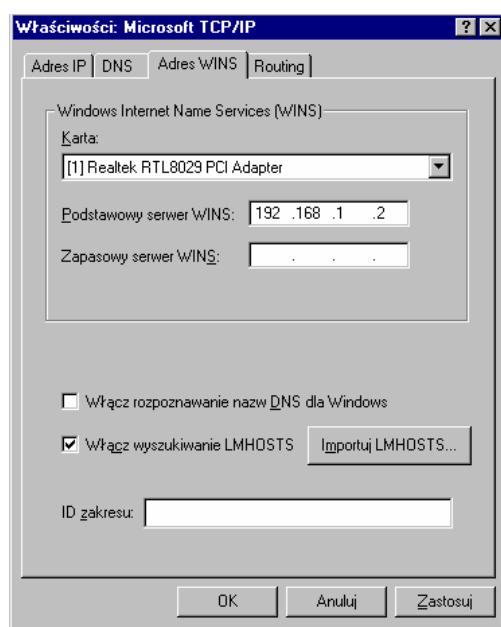
Przechowywany lokalnie plik HOSTS może posłużyć do rozwiązywania nazw NetBIOS, jeśli system zostanie do tego odpowiednio skonfigurowany. Plik ten mieści się w folderze *\drivers\etc* i przeszukiwany jest jednokrotnie od początku w dół.

DNS

Gdy skonfigurujemy odpowiednio klienta, wówczas serwer DNS może posłużyć do rozwiązywania nazw NetBIOS. Rysunek 10.9 przedstawia wymagane dane konfiguracyjne dla klienta NT.

Rysunek 10.9.

Sterowanie
rozwiązywaniem
nazw NetBIOS



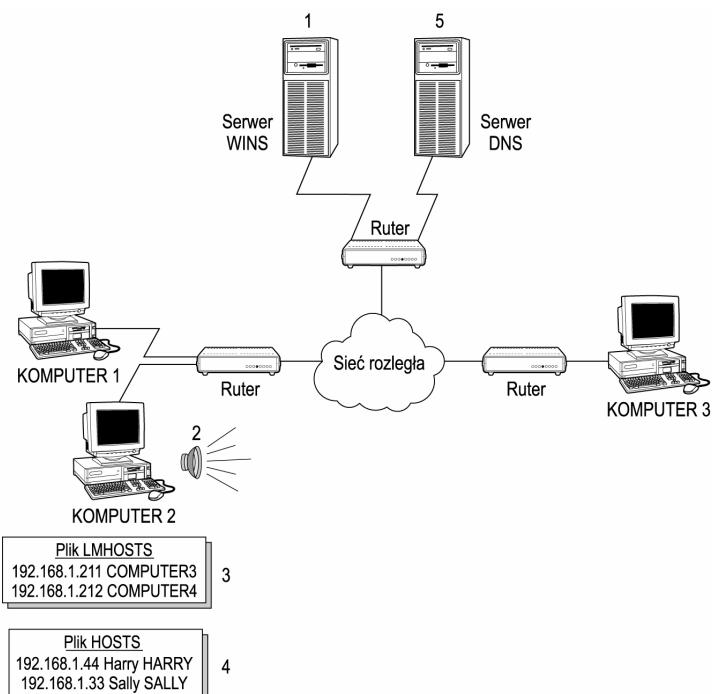
Pola dialogowe *Podstawowy serwer WINS* i *Zapasowy serwer WINS* zawierają adres IP przynajmniej jednego serwera WINS, aby umożliwić rejestracje i zapytania w usłudze WINS. Pole wyboru *Włącz wyszukiwanie LMHOSTS* pozwala ustalić, czy plik LMHOSTS będzie przeszukiwany podczas rozwiązywania nazw, czy nie. Pole wyboru *Włącz rozpoznawanie nazw DNS dla Windows* pozwala na używanie takiej samej pisowni plików HOSTS i DNS-u do rozwiązywania nazw NetBIOS.

Poniewaz dostepnych jest wiele narzedzi rozwiazywania nazw NetBIOS, nalezy upewnic sie co do kolejnosci ich stosowania, aby z powodzeniem znajdowac problemy. Aplikacje NetBIOS uzywaja rozwiazywania nazw NetBIOS. Aplikacje WinSock uzywaja plików HOSTS i DNS-u. Aby wiec poznac kolejnosc rozwiazywania nazwy, nalezy uruchomic aplikacje NetBIOS. Moze to byc Eksplorator, Menedzer plików lub nawet polecenie net use; nie nalezy jednak uzywac poleceni ping czy telnet, poniewaz sa one aplikacjami Sockets.

Rysunek 10.10 pokazuje, w jaki sposob PC2 probuje rozwiazac nazwe NetBIOS komputera PC3.

Rysunek 10.10.

*Kolejnosc
rozwiazywania
nazw NetBIOS*



Uzytkownik maszyny PC2 wydal wlasnie polecenie net use x: \\PC3\APLIKACJE, usilujac przypisac X: do udzialu z aplikacjami w PC3. Jesli PC2 laczyl sie z PC3 w ciagu kilku ostatnich minut, to odwzorowanie nazwy NetBIOS powinno byc dostepne w pamieci podrecznej nazw NetBIOS komputera PC2; w naszym przypadku jednak tak nie jest.

1. Poniewaz PC2 nie posiada w pamieci podrecznej wpisu dla komputera PC3 i jest skonfigurowany jako klient uslugi WINS, PC2 formuluje zadanie rozwiazania nazwy NetBIOS i wysyla je do serwera WINS. Serwer WINS jest dostepny i jesli posiada w bazie danych WINS wpis uslugi serwera w komputerze PC3, to skojarzony z nia adres IP zostanie odeslany do PC2 w odpowiedzi na zapytanie, a lacznosc zostanie nawiazana. Serwer WINS nie posiada niestety wpisu dla PC3.
2. Komputer PC2 wysyla rozwiazanie do lokalnej uslugi nazewniczej NetBIOS, ktora zawiera nazwe PC3. Jesli PC3 znajduje sie w lokalnym segmencie,

to odpowie na rozgłoszenie swoim adresem IP i łączność zostanie nawiązana. Oczywiście PC3 nie znajduje się w tym samym segmencie co PC2.

- 3.** PC2 przeszukuje jednokrotnie lokalny plik LMHOSTS od początku w dół. W pliku znajduje się wpis dla PC3. Adres IP w tym wpisie posłuży do nawiązania łączności z PC3. Gdyby jednak wpisu dla PC3 nie było w pliku LMHOSTS, proces rozwiązywania nazwy trwałby dalej.
- 4.** Jeśli PC2 jest skonfigurowany tak, by korzystać z DNS-u w Windows Networking, to będzie przeszukiwać lokalny plik HOSTS w nadziei na znalezienie hosta PC3. Gdy wpis dla PC3 zostanie znaleziony, zawarty w nim adres IP posłuży do nawiązania łączności z PC3.
- 5.** Jeśli PC3 nie zostanie znaleziony w pliku HOSTS, natomiast PC2 jest skonfigurowany jako resolwer, to PC2 wysła do szczegółowego w swojej konfiguracji serwera DNS zapytanie o PC3. Gdy serwer DNSwróci odpowiedź, adres IP w niej zawarty posłuży do nawiązania łączności z PC3.

Jeśli adres IP komputera PC3 nie zostanie rozwiązany za pomocą zadnej z powyższych metod, to PC2 wyświetli komunikat o błędzie i nie beda podejmowane zadne dalsze czynności.

Typy wezłów NetBIOS

Kolejność stosowania dwóch metod rozwiązywania nazw: WINS i rozgłoszenia może być zmieniona poprzez modyfikacje typu wezła klienta NetBIOS.

RFC 1002 podaje cztery typy wezłów NetBIOS: B dla samych rozgłoszeń (*Broadcast*), P dla samego serwera nazw NetBIOS, H dla kolejności: najpierw serwer nazw, następnie rozgłoszenie (wezel hybrydowy) oraz M dla odwrotnej kolejności. Typ wezła NetBIOS może być konfigurowany za pomocą opcji DHCP lub ręcznie.

Microsoft rozszerza możliwości tych czterech typów wezłów, zgodnych z RFC, dodając zdolność korzystania z pliku LMHOSTS, przekazniki WINS i wywołania Windows Sockets, które pozwalają na użycie DNS-u i plików HOSTS do rozwiązywania nazw NetBIOS.

Pokażalismy, że klient z uruchomiona aplikacja NetBIOS korzysta z metod rozwiązywania nazw w określonej kolejności. Kolejność te, poprzez zmianę typu wezła NetBIOS, możemy zmodyfikować tak, by dostosować ją do warunków w otaczającej sieci. Typy wezłów służą jedynie do zmiany kolejności, w jakiej klienci używają usługi WINS i rozgłoszeń.

Tabela 10.2 przedstawia wpływ czterech typów wezłów NetBIOS na rozwiązywanie nazw.

Tabela 10.2. Typy wezłów NetBIOS

Typ wezła	Kolejność rozwiązywania nazw
Wezel typu B (B-Node)	Tylko rozgłoszenia
Wezel typu P (P-Node)	Tylko serwer nazw NetBIOS

Wezel typu H (H-Node)	Serwer nazw, nastepnie rozgloszenie
Wezel typu M (M-Node)	Rozgloszenie, a nastepnie serwer nazw

Jesli dana siec nie korzysta z serwera nazw NetBIOS, to najprawdopodobniej stosowane sa w niej wezly typu rozgloszeniowego, domyslne dla systemu Windows.

Wyobrazmy sobie, iz mamy zaimplementowac usluge WINS w sieci lokalnej zlozonej z trzech segmentow, w pojedynczej lokalizacji i posiadajacej 600 uzytkownikow. Nie wiemy, jakie bedzie obciazenie serwera WINS, wiec zastosowanie uslugi WINS we wszystkich klientach od razu nie jest pozadane. Jak wiec postapic z implementacija?

Jednym ze sposobow jest reczne skonfigurowanie wszystkich klientow. Wprawdzie podejscie takie pozwala na stopniowa implementacje, lecz jest nierealne w przypadku sieci zlozonej z 600 klientow.

Lepszym podejsciem moze byc uzycie serwerow DHCP w kazdej podsieci do skonfigurowania wszystkich klientow pod WINS, jedna podsiec po drugiej. Nadal jest to implementacja niejednaczesa, lecz wymaga znacznie mniejszych nakladow pracy. DHCP moze rowniez posluzyc do modyfikacji typow wezlow klientow. Z uwagi na wymog implementacji etapowej, najlepszym poczatkowym ustawieniem moze byc wezel typu M.

Jesli dla klientow zastosujemy tylko wezly typu M, to zakonczone niepowodzeniem zadania rozwiazania nazw przez rozgloszenie beda wysylane do serwera WINS. Moze to znaczaco zmniejszyc obciazenie serwera. W trakcie implementacji, gdy we wszystkich segmentach zostanie udostepniona usluga WINS, mozna bedzie na podstawie parametrów obciazenia serwera decydowac o przechodzeniu na wezly typu H.

Administratorzy niektórych sieci wydali wojne ruchowi sieciowemu powodowanemu przez rozgloszenia. Mogą oni wybrać wezel typu P, przez co klienci będą korzystać jedynie z usługi WINS, nie generując żadnych rozgłoszeń.

Rozwiązywanie nazw NetBIOS w Windows 2000

W Windows 2000 NetBIOS można stosować opcjonalnie, lecz DNS jest obowiązkowy. W środowisku złożonym tylko z systemów Windows 2000 NetBIOS jest zbytni. Podstawowym narzędziem identyfikacji w Windows 2000 stała się usługa DNS.

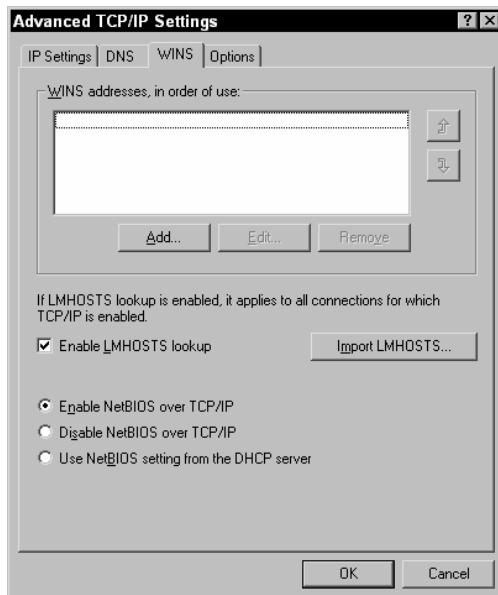
Metody służące do rozwiązywania nazw NetBIOS w Windows 2000 są podobne do standardowych metod opisanych wcześniej, z kilkoma różnicami.

Po pierwsze, NetBIOS można wyłączyć lub kontrolować za pomocą dynamicznych danych konfiguracyjnych otrzymanych z DHCP. Rysunek 10.11 przedstawia okno, w którym można konfigurować funkcjonalność NetBIOS-u w kliencie Windows 2000 Professional. W tym oknie można też konfigurować wyszukiwanie za pomocą pliku LMHOSTS.

Gdy NetBIOS jest aktywny, można stosować wszystkie cztery typy wezłów zgodne z RFC; jednakże w Windows 2000 nie można złączyć stosowania usługi DNS do rozwiązywania nazw Windows. Jesli chcemy wykorzystać DNS lub plik HOSTS do roz-

wiązywania nazw Windows w Windows 2000, trzeba w lokalnym Rejestrze ustawić parametr *EnableDns* w katalogu *Netbt\parameters* na 1.

Rysunek 10.11.
Konfiguracja NetBIOS-u w Windows 2000



Dalsze informacje o protokole DHCP można znaleźć w rozdziale 9.

W Windows 2000 nawet gdy NetBIOS nie jest złączony, protokół SMB nadal funkcjonuje, dzięki nowemu interfejsowi o nazwie Urządzenie SMB. Interfejs ten pozwala na połączenia Windows Networking bez korzystania z usługi NetBIOS. Działa jak aplikacja Sockets — używa usługi DNS i plików HOSTS oraz portu TCP 445 zamiast standardowego dla usługi NetBIOS portu TCP 139.

Na koniec, NetBIOS przez TCP/IP w Windows 2000 sprawdza, czy nazwa zawiera znak kropki lub więcej niż 15 znaków. Gdy dowolny z tych warunków jest spełniony, do rozwiązania nazwy zostaje użyty najpierw DNS, a po nim NetBIOS.

Poniższa procedura przedstawia rozwiązanie nazw NetBIOS w komputerze Windows 2000. Zakładamy, iż NetBIOS jest złączony, że używana aplikacja jest aplikacja NetBIOS-u, oraz że właśnie wprowadziliśmy do niej nazwę.

1. Zostaje sprawdzony typ wezła NetBIOS lokalnego komputera.
2. Długość i zawartość podanej nazwy zostają skontrolowane. Używane są standardowe metody rozwiązania nazw NetBIOS, najpierw pamięć podręczna nazw, a następnie usługa WINS i (lub) rozgłoszenie — zależnie od typu wezła. Jeśli nazwa ma więcej niż 15 znaków lub zawiera kropkę, to zostaje uznana za nazwę hosta i wysypane jest zapytanie DNS.

3. Gdy zapytanie DNS nie powiedzie sie, wówczas stosowane sa typowe metody rozwiązania nazw NetBIOS — najpierw pamiec podreczna nazw, a nastepnie usluga WINS i (lub) rozgłoszenie, zależnie od typu wezla.
4. Gdy wszystkie inne metody zawiodą, przeszukiwany jest plik LMHOSTS, jeżeli opcja ta jest załączona (patrz rysunek 10.11).
5. Jeżeli w Rejestrze lokalnego komputera jest wprowadzony i aktywowany parametr EnableDns, to w celu rozwiązania nazwy NetBIOS jest przeszukiwany plik HOSTS.
6. Jeżeli nazwa nie zostanie znaleziona w pliku HOSTS, to stosowana jest ponownie usługa DNS.

Gdy żadna z powyższych metod nie pozwoli rozwiązać nazwy NetBIOS na adres IP, wówczas do lokalnego komputera wysyłany jest komunikat o błędzie i rozwiązywanie nazwy kończy się niepowodzeniem.

Czesc III

Popularne aplikacje TCP/IP

W tej czesci:

- ◆ Rozdzial 11. Dostep do Internetu
- ◆ Rozdzial 12. Narzedzia do obslugi plikow
- ◆ Rozdzial 13. Narzedzia zdalnego wykonywania polecen
- ◆ Rozdzial 14. Drukowanie przez siec
- ◆ Rozdzial 15. Aplikacje i protokoly WWW
- ◆ Rozdzial 16. Dostep do poczty i grup dyskusyjnych
- ◆ Rozdzial 17. Uslugi informacyjne dla przedsiebiorstw

Teraz, gdy protokol TCP/IP zostal zainstalowany i skonfigurowany, pora przyjrzec sie sposobom jego wykorzystania. Czesc III omawia róznorodne aplikacje, których moze my uzyc z protokołem TCP/IP, aby zapewnic funkcjonalnosc sieci.

Rozdzial 11. pokazuje, jak polaczyc siec z Internetem i jak budowac własna siec za pomoca technologii polaczen dwupunktowych. Nastepnie przejdziemy do przesyłania danych przez siec lokalna i Internet, co obejmuje przesyłanie plików (rozdzial 12.), zadan drukowania (rozdzial 14.), stron WWW (rozdzial 15.) oraz innych danych — zarówno bezpośrednio, jak i za pomoca adresowania grupowego.

Rozdzial 13. zajmuje sie zdalnym korzystaniem z systemu, przedstawiajac proste protokoly — takie jak Telnet — oraz bardziej zlozone rozwiązania, na przyklad uslugi terminalowe. Rozdzial 17. koncentruje sie na systemach informacyjnych przedsiebiorstw, miedzy innymi NIS, StreetTalk, NDS i Active Directory, które opieraja sie na standarzie X.500.

Rozdział 11.

Dostęp do Internetu

W tym rozdziale:

- ◆ Przegląd miedzysieci prywatnych i publicznych
- ◆ Laczenie sie z Internetem
- ◆ Wykorzystanie zapór firewall
- ◆ Tłumaczenie adresów sieciowych (NAT)
- ◆ Wirtualne sieci prywatne

Internet, wpływając na wszystkie dziedziny życia w nowoczesnym społeczeństwie — od nauki po rozrywkę, stał się zjawiskiem wszechobecnym. Ponieważ można w nim znaleźć większość produktów i usług, rewolucjonizował nawet sposoby prowadzenia interesów. Dostęp do Internetu przestaje być przywilejem i staje się koniecznością, w wyniku czego coraz więcej osób prywatnych oraz firm (dużych i małych) łączy się z Internetem.

Największa zaleta Internetu jest istniejąca w nim „kultura otwarta”, przez którą sieć bywa nazywana „Utopia” i „prawdziwa demokracja”. Z drugiej strony, ta sama dostępność i otwartość może być szkodliwa, ponieważ co niektóre jednostki wykorzystują Internet z egoistycznych lub nikczemnych побudek. Na przykład, Internet stał się najbardziej popularnym medium rozprzestrzeniania wirusów, które są w stanie rozłożyć całą sieć przedsiębiorstwa. Na skutek tego coraz więcej sieci należących do firm jest stale narządzonych na kradzież, uszkodzenie lub nieupoważniona modyfikacja ważnych danych.

W tym rozdziale Czytelnik zapozna się z sieciami prywatnymi, należącymi do przedsiębiorstw i organizacji. Przedstawimy schemat adresowania stosowany w sieciach prywatnych i ograniczenia, jakie nakłada obecny schemat adresowania IPv4. Pokażemy, jak połączyć się z Internetem za pomocą konta i bramy udostępnianych przez dostawcę usług internetowych (ISP — *Internet Services Provider*), który pozwala na połączenie firmy lub osoby prywatnej z najbardziej znana z wszystkich sieci — Internetem. Czytelnik dowie się również o szybko rozwijającym się rynku dostawców usług aplikacji (ASP — *Application Service Provider*), którzy oferują organizacjom dzierżawe różnorodnego oprogramowania, od systemów Windows do bardzo złożonych i kosztownych programów planowania zasobów przedsiębiorstwa (ERP — *Enterprise Resource Planning*).

Podlaczanie do Internetu moze wiazac sie dla sieci z wysokim zagrozeniem bezpieczenia. Omowimy tutaj zapory firewall, ktore zabezpieczaja siec przedsiebiorstwa przed nieupowaznionym dostepem i hakerami. Nastepnie omowimy mechanizm tlumaczenia adresow sieciowych NAT (*Network Address Translation*) i jego role w walce administratorow sieciowych ze stalem problemem gwałtownie kurczacych sie zasobów dostepnych adresów IP oraz rosnacej podatnosci sieci na zlosliwe ataki. Czytelnik zapozna sie również z serwerami proxy oraz wspoldzieleniem polaczen internetowych w systemach Microsoftu. Na koniec przedstawimy zdobywajace szybko popularnosci i bardzo bezpieczne wirtualne sieci prywatne (VPN — *Virtual Private Network*), ktore pozwala na bezpieczne i ekonomiczne przesyłanie waznych danych w sieciach przedsiebiorstw o geograficznie rozrzuconych oddzialach.

Przeglad miedzysieci prywatnych i publicznych

Poniewaz TCP/IP został na calym swiecie zaakceptowany w roli standardu sieciowego, olbrzymia liczba sieci korzysta z tego pakietu protokolów i jego zdolnosci adresowania w lacznosci miedzy sieciami i intranetowej. W sieci TCP/IP hosty mozna podzielic na trzy kategorie:

- ◆ Hosty nie potrzebujace zdalnego dostepu do zasobów i uslug swiadczonych przez hosty z innej sieci lub samego Internetu.
- ◆ Hosty wymagajace dostepu do ograniczonych zasobów lub uslug, udostepnianych przez „zaufane” sieci lub hosty spoza wlasnej sieci. Do takich uslug moga zaliczac sie: poczta elektroniczna, ftp, zdalne logowanie i tak dalej.
- ◆ Hosty wymagajace stalego i nieograniczonego dostepu do zasobów i uslug swiadczonych przez inne sieci lub hosty spoza sieci macierzystej.

Hosty z pierwszej i drugiej kategorii naleza do sieci prywatnych. Dobrym przykladem takiej sieci moze byc siec organizacji bankowej, ktorej oddzialy moga miec sie w różnych miastach na calym swiecie. Aby organizacja funkcjonowala dobrze, sieci wszystkich oddzialow musza byc ze soba polaczone. Aby jednak zapewnic nietykalnosci i bezpieczenstwo transakcji i danych, osoby z zewnatrz nie powinny miec prawa dostepu do sieci. Hosty nalezace do trzeciej kategorii tworza siec publiczna. Internet, do którego kazdy ma dostep z dowolnego miejsca na swiecie, jest najlepiej znanym przykładem sieci publicznej.

Kazde urzadzenie w sieci TCP/IP otrzymuje unikatowy adres IP na potrzeby identyfikacji i wlasciwego funkcjonowania. Wszystkie komputery w sieci publicznej uzywaja globalnie unikatowych adresów IP, przyznanych przez internetowa organizacje rejestrujaca. Komputery te moga komunikowac sie z wszystkimi innymi, nalezacymi do sieci, jak również komputerami nalezacymi do innych sieci publicznych; nie maja jednak lacznosci z komputerami w sieciach prywatnych.



Kilka upowaznionych do tego jednostek, takich jak InterNIC, odpowiada za przydzielanie adresów IP dostawcom uslug internetowych (ISP) i firmom, aby zapewnic unikatowosc adresów przyznawanych w Internecie i innych sieciach publicznych.

Wiele firm dla wygody, pełnej kontroli i redukcji kosztów stosuje własny schemat adresowania IP w obrębie swoich sieci prywatnych. Taki schemat adresowania funkcjonuje skutecznie, dopóki w sieci z niezależnym schematem adresowania nie zaistnieje potrzeba komunikowania się z innymi sieciami. Przeniesienie komputera z domeny prywatnej do publicznej — lub vice versa — wymaga zmiany jego adresu IP, wpisów w DNS-ie oraz w innych plikach w innych komputerach, które odwołują się do danego komputera używając jego adresu IP.



Więcej informacji o usłudze DNS zawiera rozdział 10.

Adresowanie w sieciach prywatnych

Sieć prywatna używa *adresów nie trasowanych* (*non-routable*), inaczej zwanych *prywatnymi adresami IP*. Jak sugeruje nazwa, routery nie przesyłają ruchu do takich adresów IP. Adresy te nie mogą być trasowane w Internecie i innych sieciach publicznych. W pewnym sensie sieć prywatna jest „odcięta” od innych sieci i Internetu.

Organizacja IANA (*Internet Assigned Numbers Authority*) zarezerwowała trzy bloki nie trasowanych adresów na użytko sieci prywatnych:

- ◆ od 10.0.0.0 do 10.255.255.255,
- ◆ od 172.16.0.0 do 172.31.255.255,
- ◆ od 192.168.0.0 do 192.168.255.255



Dodatkowe informacje o adresach sieci prywatnych znajdują się w dokumencie RFC 1918.

Adresy z pierwszego bloku należą do pojedynczej sieci klasy A. Drugi blok stanowi zestaw 16 kolejnych adresów sieci klasy B, zas trzeci jest zestawem 256 kolejnych adresów sieci klasy C. Każda sieć prywatna może użyć tych trzech bloków adresów na potrzeby adresowania wewnętrznego. Oznacza to, że adresy z puli prywatnej są unikatowe tylko w obrębie sieci prywatnej lub zbioru sieci, które muszą komunikować się ze sobą w obrębie prywatnej miedzysieci. Adresy prywatne poza daną siecią nie mają żadnego globalnego znaczenia, a co za tym idzie, informacje o trasach związane z sieciami prywatnymi nie są propagowane do łącz internetowych, zas pakiety danych o prywatnych adresach źródłowych lub docelowych nie są przekazywane do routera. Ogólnie mówiąc, routery w sieciach publicznych, zwłaszcza należące do dostawców usług internetowych, skonfigurowane są tak, by odrzucać dane tras dotyczące sieci prywatnych.



Według RFC 1918, jeśli ruter w sieci publicznej otrzyma informacje dotyczące sieci prywatnej, to odrzucenie takich informacji nie jest uznane za błąd protokołu trasowania.

Komputery w sieci prywatnej mogą komunikować się z wszystkimi pozostałymi komputerami z tej sieci. Choć nie mają dostępu do komputerów spoza sieci własnej lub zaufanej, nadal posiadają dostęp do zewnętrznych usług za pomocą bram. W przeci-

wienstwie do nich, w sieciach publicznych wymagana jest globalnie unikatowa przestrzeń adresów, która można otrzymać od internetowego rejestratora. Adresy IP służące do łączności na zewnątrz nigdy nie są przydzielane z bloku adresów prywatnych.



Dodatkowe informacje o adresowaniu IP i klasach adresów IP (A, B i C) zawiera rozdział 5.

Wytyczne do projektu sieci prywatnej

Podczas projektowania sieci prywatnej musimy pamiętać o kilku sprawach:

- ◆ W dzisiejszych warunkach wiecej niż jeden komputer musi być podłączony do Internetu lub innej sieci publicznej w sposób trwałym, nawet jeśli należy do sieci prywatnej. W związku z tym, najlepiej zacząć od projektu prywatnej części sieci, a następnie przejść do publicznej podsieci.



Projekt sieci prywatnej nie powinien być trwałym, ponieważ jeden lub wiele komputerów może wymagać zmiany statusu z prywatnego na publiczny lub na odwrotnie. zaleca się więc grupować hosty o podobnych wymaganiach dotyczących łączności w odrębne podsieci. Pomoże to administratorowi sieci unikać poważnych przerw w jej funkcjonowaniu.

- ◆ Należy unikać podłączania hostów o adresach publicznych i prywatnych do wspólnego nosnika fizycznego.
- ◆ Routery łączące sieć prywatną z innymi sieciami, zwłaszcza publicznymi, powinny mieć skonfigurowane odpowiednie filtry pakietów i tras po obu stronach. Zapobieganie to przeciekom pakietów i informacji o trasowaniu.
- ◆ Jeśli dwie sieci prywatne łączą się ze sobą za pomocą niezaufanej sieci publicznej, należy zaimplementować pakowanie danych (*data encapsulation*), co zminimalizuje ryzyko nieupoważnionego dostępu.



Sieć nie zaufana oznacza sieć zewnętrzna, w której przesyłu danych nie można uznać za bezpieczny i godny zaufania.

- ◆ Aby zapobiec konfliktom adresów podczas komunikacji pomiędzy dwiema sieciami prywatnymi, organizacja powinna wybrać losowo adresy z puli adresów prywatnych.



Dodatkowe informacje o podsieciach znajdują się w rozdziale 5.

Zalety i wady przestrzeni prywatnych adresów sieciowych

Korzystanie z przestrzeni prywatnych adresów sieciowych przynosi kilka korzyści:

- ◆ Pozwala oszczędzać malejące szybko zasoby dostępnych adresów globalnie unikatowych, ponieważ są używane tylko w razie konieczności.
- ◆ Pozwala na tworzenie wygodnych w eksploatacji i zarządzaniu schematów adresowania i na łatwiejszą rozbudowę, ponieważ administratorzy sieci mają do dyspozycji większa przestrzeń adresów niż otrzymana z puli globalnej.

- ◆ Zwiększa bezpieczeństwo, chroniąc przed nieupoważnionym dostępem
— ponieważ sieć jest niewidoczna z zewnątrz.

Stosowanie przestrzeni adresów prywatnych ma również kilka poważnych wad:

- ◆ Przy połączeniu z innymi sieciami poprzez Internet mogą pojawić się problemy związane z adresowaniem. Założymy, że sieć prywatna używa adresów IP nie przydzielonych przez IANA lub innego rejestratora internetowego. Możliwe, iż ta sama przestrzeń adresów jest legalnie przyznana innej sieci. Jeśli więc dana sieć prywatna zostanie połączona później z Internetem, to wystąpi kolizja adresów powodująca poważne problemy z trasowaniem.
- ◆ Gdy sieć prywatna łączy się z Internetem i innymi sieciami publicznymi, kilka (lub wszystkie) komputerów w tej sieci musi zmienić adresy, co powoduje dodatkowe wydatki ze strony firmy. Koszt zmiany adresów jest wprost proporcjonalny do liczby hostów, które trzeba będzie przenieść z sieci prywatnej do publicznej.



Problemy ze zmianą adresów można złagodzić za pomocą mechanizmu tłumaczenia adresów sieciowych (NAT).

- ◆ W przypadku połączenia dwóch lub kilku sieci prywatnych, kilka adresów w nowej sieci może się powtarzać. Ponownie pojawia się problem zmiany kolidujących adresów komputerów.
- ◆ Więcej pracy mają administratorzy w sieci prywatnej, ponieważ wszystkim komputerom trzeba przyznać unikatowe adresy IP z przestrzeni adresów prywatnych. Im większej liczby hostów trzeba nadać adresy, tym większe nakłady pracy administratorów sieci.

W ciągu ostatnich lat, zwłaszcza w minionej dekadzie, wykorzystanie Internetu wzrosło gwałtownie. Firmy, które chciały dodać zachować prywatność własnych intranetów, obecnie zwrócili się w stronę Internetu i łączności globalnej. Jednakże ogromny rozwój Internetu sprawia problemy ze skalowaniem; poza tym ujawnił luki w istniejącym schemacie adresowania IP.

Ograniczenia IPv4

Protokół IP w wersji 5 (IPv4), potocznie nazywany IP, został opracowany pod koniec lat 70. jako podstawowy mechanizm komunikacji w pakiecie protokołów TCP/IP. Jest w istocie najbardziej popularnym mechanizmem komunikacji w Internecie. Jego konstrukcja jest na tyle stabilna, wydajna i elastyczna, że przetrwała do dzisiaj w praktycznie niezmienionej postaci.



Przed IPv4 opracowane zostały inne wersje protokołu IP, lecz żadna z nich nie otrzymała formalnej nazwy.

W ostatniej dekadzie technologie informacyjne rozwijają się wielkimi krokami. Wzrosła szybkość procesorów. Komputery posiadają pamięć RAM mierzoną w gigabajtach.

Szybsze technologie LAN, takie jak ATM i FDDI, zastapily starsze i wolniejsze, takie jak Ethernet. A co najwazniejsze, liczba hostów (komputerów) w Internecie przekroczyła pulap 100 milionów. Z najnowszych badan wynika, iż rozmiary Internetu podwajaja się co dziewiec miesiecy. W wyniku tego rozwoju technologicznego protokół IPv4 — mimo solidnej konstrukcji — napotkał dwa poważne problemy ze skalowaniem, starać się nadazyc za szybkim rozwojem technologii i równoczesnie zapewnic ciągły i nieprzerwany wzrost rozmiarów sieci:

- ◆ *Ostateczne wyczerpanie przestrzeni adresów* — IPv4 został zaprojektowany tak, by udostępniać 32-bitowa przestrzeń adresów. Oznacza to, że dostępnych jest tylko 4 294 967 296 (232) adresów IP. Na początku, gdy bardzo niewiele organizacji posiadało sieci lokalne, a jeszcze mniej miało dostęp do sieci globalnych, liczba ta wydawała się ogromna. Jednakże przy zakładanym rozwoju Internetu ta skonczona liczba adresów IP ulegnie w końcu wyczerpaniu, a ponadto część przestrzeni adresów IP nie została przydzielona wydajnie. Jeśli obecna polityka przydziału adresów nie zostanie szybko zmieniona, nowi użytkownicy nie będą w ogóle mogli łączyć się z Internetem.
- ◆ *Wzrost rozmiarów internetowych tablic tras* — routery w Internecie muszą utrzymywać pełne informacje o trasach. W ciągu ostatnich lat rozmiary tablic tras rosły wykładniczo, w miarę przyłączania się do Internetu kolejnych osób i organizacji. Sytuacje pogarszały jeszcze takie czynniki, jak zdolność procesora do przetwarzania zmian związanych z trasami, dynamiczny charakter połączeń międzymiejsiowych, wpływ dynamicznego charakteru tras na pamięć podręczną i sama objętość informacji, którymi trzeba zarządzać recznie i mechanicznie. Tego problemu nie da się rozwiązać przez prostą rozbudowę pamięci routera lub zwiększenie rozmiarów tablic tras. Jeśli liczba wpisów w globalnych tablicach tras będzie mogła rosnąć bez żadnych ograniczeń, to routery sieci szkieletowej Internetu będą zmuszone do odrzucania tras, przez co fragmenty Internetu staną się niedostępne.

Organizacje internetowe, a zwłaszcza IETF (*Internet Engineering Task Force*) oraz IAB (*Internet Architecture Board*) od kilku lat pracują nad problemami związanymi z IPv4. W wyniku tych prac ograniczenia IPv4 zostały rozwiązane w nowej, szóstej wersji protokołu IP (*IPv6*, inaczej *IP Next Generation, IPng*).

Adresy IPv6 rozwiązują wszystkie problemy stwarzane przez IPv4. Protokół ten obsługuje 128-bitowe adresy, elastyczny format nagłówka, umożliwiający przydział zasobów oraz rezerwuje miejsce na dalsze rozszerzenia. Migracja istniejącej infrastruktury sieciowej do IPv6 potrwa jednak jeszcze kilka lat.

Sieć używająca adresów prywatnych jest siecią prywatną, natomiast sieć używająca nieprywatnych adresów IP nosi nazwę sieci publicznej. Różnice pomiędzy sieciami prywatnymi i publicznymi przedstawia tabela 11.1.

Mozemy dzisiaj z łatwością połączyć się z Internetem z domu lub z pracy. W przypadku dużych przedsiębiorstw łatwość zapewniają prywatne bramy. Małe firmy zwykle korzystają z oferty dostawców usług internetowych (ISP — *Internet Service Provider*),

Tabela 11.1. Sieci prywatne i publiczne — porównanie

Sieci prywatne	Sieci publiczne
Siec prywatna jest własnością jednej organizacji, która ją zarządza i eksploatuje. Działa niezależnie od innych sieci.	Siec publiczna stanowi własność posredniczącego operatora, na przykład sieci PSTN (Public Services Telephone Network).
Laczność z sieciami zewnętrznymi i publicznymi, na przykład z Internetem, jest ograniczona i skojarzona kontrolowana.	Laczność z innymi sieciami nie podlega ograniczeniom.
Siec prywatna jest bardzo odporna na złośliwe ataki — na przykład na wirusy lub działalność hakerów.	Siec publiczna jest wyjątkowo wrażliwa na ataki wirusów i hakerów.

czyli firm dających swoim klientom laczność z Internetem. Połączenia z domu również odbywają się za pośrednictwem ISP — wystarczy mieć linie telefoniczną, modem i aktywne konto u ISP.

Laczenie się z Internetem

Z Internetem laczymy się zasadniczo po to, aby skorzystać z usług — poczty elektronicznej, FTP, Telnetu, WWW, pogawędek i grup dyskusyjnych Usenet. Powinnismy rozważnie wybrać sposób połączenia, w zależności od intensywności korzystania z dostępu do Internetu oraz typów potrzebnych usług.



Chociaż każdy ISP obiecuje udostępniać wszelkie usługi i typy dostępu, nie zawsze tak jest. Musimy więc uważnie wybrać dostawcę, który spełnia nasze wymagania.

Na potrzeby różnych usług można wybierać różne sposoby połączenia z Internetem:

- ◆ *Bezpośrednio przez sluzacy do tego komputer* — metoda kosztowna, lecz daje pełny dostęp do wszystkich usług internetowych. Ta metoda zalecana jest dla dużych przedsiębiorstw.
- ◆ *Przez zdalna brame* — bardzo ekonomiczna metoda, wykorzystująca cudze połączenie z siecią szkieletową Internetu. Daje pełny dostęp do wszystkich usług internetowych. Ta metoda jest zalecana dla studentów, którzy mogą wykorzystać bramę uczelnianą, oraz dla pracowników firm, którzy na potrzeby dostępu do Internetu mogą wykorzystać bramę swojej organizacji.
- ◆ *Poprzez ISP lub dostawce usług online* — ta metoda pozwala na dostęp do Internetu osobie lub firmie po opłaceniu miesięcznych kosztów połączenia i eksploatacji. W zależności od dostawcy, możemy uzyskać pełny dostęp do usług internetowych lub z ograniczeniami dotyczącymi dostępu do poszczególnych usług. Ta metoda jest zalecana dla osób laczących się z Internetem z domu.
- ◆ *Darmowy dostęp do Internetu* — ISP i dostawcy usług online zwykle oferują te metody, aby przyciągnąć klientów. Jedynym kosztem dla użytkownika jest rachunek telefoniczny za impulsy zużyte podczas pracy w Internecie. Dostęp opłacany jest przez reklamodawców. Darmowy dostęp do Internetu ma jednak wady — czas online jest mocno ograniczony a użytkownik nie ma wyboru usług.

Tylko duze firmy i korporacje stac na infrastrukturę niezbedna, aby na stale polaczyc sie z Internetem. Jesli wiec Czytelnik chce uzyskac polaczenie jako osoba indywidualna, to najlepszym wyborem moze byc ISP, dostawca uslug online lub zezwolenie na wykorzystanie czystej bramy. Aby jednak skorzystac z cudzej infrastruktury, trzeba byc studentem lub pracownikiem przedsiębiorstwa zapewniajacego polaczenie z Internetem.

Dla uzytkowników, którzy na potrzeby prywatne chca uzyskac z domu dostep do Internetu, najlepszym rozwiazaniem jest ISP lub dostawca uslug online. Oba typy dostawcow pozwalaja uzytkownikom pracowac w Internecie w dowolnym rytmie. Prosze jednak pamietac, ze:

- ◆ Niektórzy dostawcy nie zapewniaja pełnego dostepu do wszystkich uslug.
- ◆ Oplaty za polaczenia i eksplotacje moga byc bardzo wysokie. Radzimy dokladnie sprawdzic tabele oplat przed podpisaniem umowy.
- ◆ Warto sprawdzic szybkosc dostepu, jaka oferuje dostawca. Jesli proponowana szybkosc transmisji nie przekracza 9600 bodów, lepiej poszukac innego ISP.



Czeste korzystanie z dostepu do Internetu z domu moze skonczyc sie niebotycznymi rachunkami dla ISP lub dostawcy uslug online.

Po zapewnieniu wymaganej infrastruktury i uzyskaniu lacnosci z Internetem pora skonfigurowac w oprogramowaniu komunikacyjnym dostarczonym przez ISP nazwe logowania i dostepowy numer telefonu. Nastepnie, aby polaczyc sie z Internetem, nalezy:

1. W przypadku polaczenia telefonicznego uruchomic oprogramowanie dostarczone przez ISP, aby nawiazac polaczenie.
2. Gdy polaczenie z brama ISP powiodlo sie, oprogramowanie zapyta o haslo do konta internetowego. Po wpisaniu prawidlowego hasla zostaniemy polaczeni z Internetem.



Jesli haslo zostało wczesniej skonfigurowane w oprogramowaniu, to zapytanie o haslo moze sie nie pojawić.

3. Teraz mozemy uruchamiac aplikacje, pozwalajace korzystac z różnorodnych uslug udostepnionych przez ISP. Przeglądarka WWW (Internet Explorer, Netscape Navigator itp.) moze posluzyc do przeglądzania różnych dostepnych w Internecie witryn, inne programy posluza do pogawedek z innymi uzytkownikami lub do korzystania z poczty elektronicznej.



Dostep do Internetu poprzez brame uczelni lub firmy (albo przez laczne dzierzawione) moze nie wymagac tych kroków.

Dostawcy usług internetowych

Dostawca usług internetowych (ISP — *Internet Services Provider*) to firma, która udostępnia *odmierzany dostęp do Internetu* małym firmom i użytkownikom indywidualnym, przede wszystkim laczącym się z domu. ISP zapewnia łączność z Internetem, udostępniając klientom swoje internetowe bramy lub rutery. Ponieważ jednak użytkownik musi podpisać z ISP umowę i comiesięcznie opłacić połączenie i jego wykorzystanie, dostęp jest odmierzany.



Jesli Czytelnik musi pozostawać zalogowany do Internetu przez dłuższe okresy, dostęp odmierzany nie jest zalecany, ponieważ może być kosztowny. Dostęp odmierzany jest najlepszy dla domowych użytkowników — w celach rozrywkowych lub gdy korzystają od czasu do czasu.

Czytelnik musi otrzymać o koncie internetowym poniższe informacje od ISP, aby pomysłnie połączyć się z Internetem:

- ◆ nazwa użytkownika lub ID logowania,
- ◆ hasło,
- ◆ numer dostępowy telefonu,
- ◆ nazwy hosta i domeny,
- ◆ adres serwera DNS,
- ◆ adres IP i ewentualnie maska podsieci,
- ◆ adres bramy domysłnej,
- ◆ proces uwierzytelniania.



Jesli ISP oferuje konto PPP (*Point-to-Point Protocol*), warto z niego skorzystać. Dostęp do takich kont jest o wiele szybszy niż w przypadku kont SLIP (*Serial Line Interface Protocol*).

Różni dostawcy oferują różne pakiety usług, dostosowane do potrzeb użytkownika. Niektórzy dostawcy oferują wszystkie usługi, inni nie. Należy wybrać pakiet najlepiej dostosowany do naszych potrzeb. Proszę pamiętać, że szybkość połączenia w dużym stopniu zależy od łączności udostępnianej przez ISP. Proszę też uwzględnić zasady sprawiające wrażenie arbitralnych. Jesli nie zrozumieć zasad, należy je koniecznie wyjaśnić, gdyż w przeciwnym razie możemy zapłacić znacznie więcej, niż planowaliśmy.

Użytkownik końcowy może łączyć się z ISP z komputera dowolnego typu (PC, Macintosh lub komputery uniwersyteckie). Potrzebna jest tylko łączność telefoniczna, modem, czynne konto u ISP oraz oprogramowanie obsługujące protokoły połączeniowe. Użytkownicy zasadniczo łączą się z ISP za pomocą *połączeń telefonicznych (dial-up connection)*, co oznacza, że użytkownik musi za pomocą modemu wybrać numer dostarczony przez ISP. Po połączeniu z ISP użytkownik jest połączony z Internetem, najczęściej za pomocą protokołu PPP (*Point-to-Point Protocol* — protokół dwupunktowy), lub SLIP (*Serial Line Interface Protocol* — protokół interfejsu łączności szeregowego). Ponieważ wielu użytkowników potrzebuje jedynie poczty elektronicznej, niektórzy dostawcy usług internetowych obsługują też protokół UUCP (*Unix-to-Unix Copy Protocol* — protokół kopiowania pomiędzy komputerami uniwersyteckimi).



Dostepny jest spory wybór pakietów oprogramowania, pomagajacych uzytkownikowi polaczyc sie z ISP. Nalezy wybrac pakiet z góry skonfigurowany dla uzywanego systemu operacyjnego. Do popularnych programów tego typu naleza Internet Anywhere firmy MKS oraz Dialup Networking Microsoftu, jednakze ten ostatni jest dostepny tylko razem z systemami operacyjnymi Windows.

Wiekszosc z nas uwaza laczenie sie z Internetem za proces „jednokierunkowy”. Zapominamy, ze gdy nasz komputer jest podlaczony do Internetu, kazda inna osoba z dostepem do Sieci moze również uzyskac dostep do naszego komputera i jego zasobów (plików, poczty elektronicznej i tak dalej). Wiekszosc komputerów jest wrażliwa na ataki, poniewaz podstawowa architektura komputerów nie zapewnia ochrony przed atakami z zewnatrz. Co wiecej, technologie typu Java i ActiveX jeszcze obniżają poziom bezpieczeństwa, poniewaz zwykle podczas wykonywania przejmują kontrolę nad środowiskiem i zasobami komputera. W wiekszosći przypadków nie mamy nawet wykryć takiego sterowania, dotyczy to szczególnie appletów w języku Java wykonywanych w naszym komputerze. Może wystarczyc odwiedzenie witryny, z której applet zaladuje się automatycznie i zacznie wykonywać. Taka sytuacja może prowadzić do poważnego narżenia bezpieczeństwa, poniewaz ważne informacje — dotyczące zarówno poszczególnych osób, jak i firm — mogą zostać skradzione.

Zapora firewall (dosłownie: sciana przeciwpozarowa) jest skutecznym środkiem ochrony sieci przed wiekszością zagrożeń bezpieczeństwa pochodzących z Internetu. Zapora taka chroni przed nieautoryzowanym dostępu do sieci lub komputera. Gdy system jest atakowany, zapora firewall zapobiega przedostawaniu się szkód (podsluch, złośliwe programy, uszkodzenia plików) z jednej strony do reszty sieci. Bez zapór firewall problemy z bezpieczeństwem sieci wymknęłyby się spod kontroli, prowadząc do zniszczeń wśród coraz większej liczby systemów.

Wykorzystanie zapór firewall

Podłączenie prywatnej sieci przedsiębiorstwa do Internetu i innych sieci publicznych stanowi duże zagrożenie dla bezpieczeństwa z uwagi na możliwość nieupoważnionego dostępu — na przykład ataków hakerów. Taki niebezpieczny dostęp może prowadzić do zalamania działania sieci, po którym przywrócenie ruchu może wymagać całego dnia lub kilku dni pracy. Oznacza to, że firma, która nie zabezpieczy wystarczająco swojej sieci, może ponieść ogromne straty, nie wspominając już o zagrożeniu poufnych danych. Rozwiązaniem, które zapobiega szkodliwym wypadkom naruszenia bezpieczeństwa, jest *zapora firewall*.



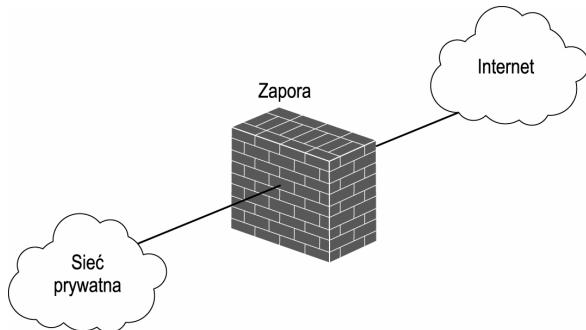
Według badań przeprowadzonych przez Warroom Research, Inc., 58 procent przepytanych firm (z ogólnej liczby 236) doświadczyło ataków hakerów na swoje sieci w przeciągu dwunastu miesięcy. 57 procent firm było atakowanych w tym czasie przynajmniej jedenastokrotnie. Jedna trzecia z tych ataków spowodowała straty, których naprawa kosztowała firmy przynajmniej milion dolarów.

Rola zapór firewall

Zapory firewall to mechanizm kontroli dostępu, zabezpieczający sieć przed nieuchronnym dostęmem. Zapora firewall jest w najprostszym ujęciu ruterem lub zestawem routerów umieszczonych w miejscu połączenia sieci prywatnej z publiczną. Położenie zapory

firewall przedstawia rysunek 11.1. Komputery w sieci prywatnej nie są wystawione bezpośrednio na „widok publiczny”. Kazda niepożądana próba dostępu do nich wymaga przedostania się przez zapory. W ten sposób zapora firewall pełni funkcje zabezpieczenia (bufora) pomiędzy siecią prywatną i publiczną — na przykład Internetem. Inaczej mówiąc, zapora chroni sieć przed niepowolonym dostępem.

Rysunek 11.1.
Tradycyjne położenie
zapory firewall



Ruterzy służace jako zapory firewall zwykle w celu zabezpieczenia sieci stosują *zasady kontroli dostępu* (lub, mówiąc prościej, *listy dostępu*). Zasady kontroli dostępu opierają się na dwóch mechanizmach: jeden z nich odpowiada za blokowanie niechcianego ruchu, drugi — za wpuszczanie do sieci reszty „niegroźnych” transmisji. Gdyby zasady kontroli dostępu nie były odpowiednio precyzyjne (to znaczy, gdyby administratorzy sieci nie za bardzo wiedzieli, jaki ruch przepuszczać, a jaki zatrzymywać), zapora firewall nie byłaby zbyt przydatna.



Zapora firewall, oprócz pełnienia funkcji kurtyny ochronnej dla intranetu organizacji, może posłużyć też do oddzielania ważnych części sieci od użytkowników, nawet zaufanych. Na przykład, część sieci używana przez dział zatrudnienia można oddzielić zaporą, aby chronić ważne dane dotyczące wyplat i osobowe przed resztą organizacji. Taki typ zapory nazywany jest często *intranetowa zapora firewall*.

Do utworzenia systemu zapory firewall może posłużyć ruter lub zestaw ruterów filtrających pakiety oraz *hosty bastionowe*. Host bastionowy to silnie zabezpieczony host (lub serwer), zezwalający na ograniczony dostęp z zewnątrz. Każdy gość z zewnątrz ma prawo dostępu do określonych danych lub aplikacji w hostie, lecz z ograniczonymi prawami, wobec czego nie jest w stanie zaszkodzić systemowi. Do przykładów hostów bastionowych należą serwery WWW, serwery anonimowych usług FTP, serwery DNS oraz węzły TACACS (*Terminal Access Controller Access Control System*).

Poza kontrolą i rejestraniem ruchu pomiędzy sieciami zapora firewall może spełniać inne funkcje, na przykład:

- ◆ tworząc wirtualne sieci prywatne (VPN — *Virtual Private Network*),
- ◆ sprawdzając poczta elektroniczna na obecność wirusów,
- ◆ filtrować adresy URL (*Uniform Resource Locator*), zabraniając dostępu do nieautoryzowanych witryn,
- ◆ filtrować aplikacje, blokując zdalny dostęp do zdalnych aplikacji, które mogą być dla sieci niebezpieczne.

Wprawdzie zapory sa poteznym mechanizmem chroniacym przed niepozadanym dostepem do sieci prywatnych i przed atakami z zewnatrz, lecz nie sa w stanie zabezpieczac zasobow organizacji przed wszystkimi atakami i wlamaniami.

- ♦ Zapory nie daja zadnej ochrony przed atakami pochodzacymi z sieci, ktora chronia.
Moga jedynie zabezpieczac granice sieci.
- ♦ Zapory firewall nie radza sobie ze zlosliwymi programami, konni trojanskimi i wirusami, poniewaz w sieciach istnieje zbyt duzo metod kodowania plikow binarnych, a na dodatek liczba istniejacych wirusow jest przytlaczajaca. Zagrozenia te mozemy jednak do pewnego stopnia ograniczyc, instalujac skuteczne oprogramowanie antywirusowe w zaporach oraz w kazdym komputerze w sieci, który jest podatny na ataki ze strony danych. Ataki tego typu polegaja na przesyłaniu poczta elektroniczna lub skopiowaniu do hosta w sieci wewnętrznej wirusów i innych zlosliwych programów.
- ♦ Zapory firewall moga byc niewlasciwie skonfigurowane. Po skonfigurowaniu zapory czesto ignoruje sie testy i weryfikacje regul. W eksplotowanych zaporach musza byc wprowadzane zmiany konfiguracji, zas dzienniki zdarzen powinny byc uwaznie i regularnie kontrolowane, aby stwierdzic, czy reguly sa prawidlowo stosowane.

Typy zapór firewall

Z teoretycznego punktu widzenia, uzywane sa trzy typy zapór firewall na różnych poziomach modelu odniesienia OSI: *zapory filtrujace pakiety*, *zapory badajace stan pakietow* i *zapory przejscia w warstwie aplikacji*. Zapory filtrujace pakiety dzialaja w trzeciej warstwie modelu OSI, zas dwa pozostałe typy w warstwach od piątej do siódmej. Wiele dostepnych na rynku zapór dazy do polaczenia funkcji dwóch lub wiecej typów. Administrator musi rozwaznie ustalic wymagania bezpieczenstwa organizacji i wedlug nich dobrac zapore.



Ogólnie mówiąc, im nizsza warstwa, w której funkcjonuje zapora, tym mniejsze jej možliwości, poniewaz moze kontrolowac nadchodzacy ruch jedynie w ograniczonym zakresie.

Obecnie projektowane sa zapory firewall przyszlosci, laczace najlepsze cechy istniejących konstrukcji. Projektanci daza do konstruowania szybkich, odsiewajacych pakiety zapór, które beda rejestrowac i analizowac przechodzacy przez nie ruch. Ponadto, coraz częściej stosuje sie w zaporach dwupunktowe szyfrowanie danych, aby chronic dane przesylane przez Internet. Tego typu zapory firewall nosza nazwe *zapór hybrydowych*, poniewaz lacza cechy wszystkich istniejacych typów. Wprawdzie sa one drozsze od pozostałych, lecz sa tez wyjątkowo wydajne i zalecane przez ekspertów.

Zapory filtrujace pakiety

Kazdy pakiet IP, zarówno przesyłany do komputera na sasiednim biurku, jak i na inny kontynent, musi zawierac adres przeznaczenia i docelowy numer portu. Kazdy pakiet IP musi tez zawierac adres IP i numer portu komputera, z którego jest wysylany, aby odbiorca wiedzial, kto ten pakiet wyslal. Inaczej mówiąc, kazdy pakiet podrózujacy w Internecie zawiera przede wszystkim pelne adresy nadawcy i odbiorcy.



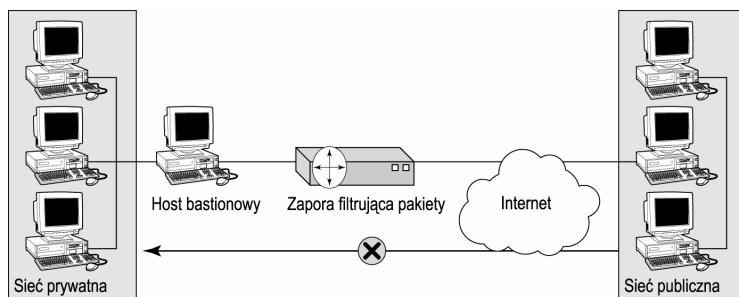
Więcej informacji o pakietach IP zawiera rozdział 5.

Zapora firewall filtrująca pakiety (*packet filter firewall*) decyduje czy pakiet przepuszcic, czy zatrzymać, na podstawie następujących informacji w nim zawartych:

- ◆ adresu źródłowego,
- ◆ adresu docelowego,
- ◆ portów (źródłowego i docelowego).

Rutery należą tradycyjnie do kategorii zapór filtrujących pakiety, ponieważ funkcjonują opierając się na wymienionych powyżej informacjach. Zapory filtrujące pakiety wyróżnia fakt, że przekazują ruch z jednej swojej strony na drugą. Aby więc pakiet pomyslnie przeszedł zapory, musi należec do bloku adresów IP, które zgodnie z konfiguracją zapory może przepuścić lub też musi używać adresu IP z sieci prywatnej, która zapora ochronia. Rysunek 11.2 przedstawia typowa konfiguracja zapory filtrującej pakiety.

Rysunek 11.2.
Konfiguracja zapory
filtrującej pakiety



Zapory tego typu mają kilka zalet:

- ◆ Są szybkie, ponieważ działają jedynie na podstawie adresów IP i numerów portów TCP, ignorując zawartość pakietów.
- ◆ Są niezależne od aplikacji, ponieważ ignorują dane zawarte w pakietach.
- ◆ Są najtanszym typem zapór.
- ◆ Nie wymagają żadnych zmian konfiguracji ochronianych komputerów.

Stosowanie zapór firewall filtrujących pakiety ma jednak również kilka wad:

- ◆ Są najmniej bezpieczne z wszystkich typów zapór, ponieważ kontrolują nadchodzący ruch jedynie w minimalnym zakresie.
- ◆ Ignorują zawartość pakietów, przez co nie pozwalają na blokowanie dostępu użytkowników do nieautoryzowanych witryn WWW.
- ◆ Nie pozwalają na zaimplementowanie złożonej zapory firewall.

Zapory badające stan pakietów

Większość zapór decyduje o przepuszczeniu lub zatrzymaniu pakietu na podstawie zawartych w nim adresów: źródłowego i docelowego. Zapory takie jednakże nie usiłują „zrozumieć” danych zawartych w pakietach. *Zapora firewall badająca stan pakietów*

(*stateful packet inspection firewall*), jak nazwa wskazuje, przechwytuje nadchodzace pakiety i sprawdza stan polaczenia. Do sieci przepuszczane sa tylko te nadchodzace pakiety, które spełniaja wszystkie warunki zdefiniowane dla zapory. Zapory badajace stan pakietów stopniowo tworza dynamiczne tablice stanów, które sluzą im do sledzenia przepuszczanych polaczen. Dopuszczalne sa jedynie pakiety nalezace do poprawnych i nawiiazanych polaczen.



Mechanizm uzywany przez zapory firewall badajace stan pakietów stosuje sie do wszystkich protokolów.

Oprócz badania adresów IP i zawartosci pakietów, zapory te biora pod uwage dodatko-wo stan polaczen. Dzieki temu nadchodzacy pakiet mozna skojarzyc z wyslanym uprzed-nio zadaniem, zanim pakiet ten zostanie dopuszczony do sieci. Zapobiega to przedostaniu sie do sieci pakietów udajacych odpowiedz na nieistniejace zadanie. Zapory badajace stan pakietów stosuja również mechanizm *filtrowania sesji*, aby gromadzic informacje o sesji od jej poczatku az do konca. Informacje te sluzą razem z adresami IP pakietu i analiza zawartosci do podejmowania decyzji o filtrowaniu.

Korzystanie z zapór firewall badajacych stan pakietów ma kilka zalet:

- ◆ Zapewniaja wyzszy poziom bezpieczenstwa niz najprostsze zapory, poniewaz kojarza nadchodzace informacje z wysylanymi zadaniami.
- ◆ Udostepniaja szczegolowe rejestrowanie transakcji, co pomaga administratorom sieci łatwo lokalizowac źródła problemów w przypadku klopotów z funkcjonowaniem sieci.
- ◆ Zmniejsza naklady pracy administracyjnej, poniewaz nie sa wymagane zadne zmiany w konfiguracji komputerów w sieci prywatnej.

Zapory te maja tez kilka wad:

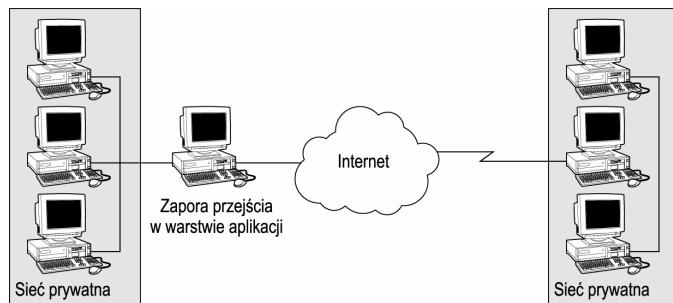
- ◆ Konfiguracja zapór badajacych stan pakietów jest skomplikowanym zadaniem.
- ◆ Nie zapewniaja uwierzytelnienia uzytkowników.
- ◆ Sa wolniejsze od zapór filtrujacych pakiety, poniewaz musza „pamietac” stany polaczen. Co za tym idzie, wymagaja wiekszych zasobów.
- ◆ Sa drozsze od zapór firewall filtrujacych pakiety

Zapory przejścia w warstwie aplikacji

Zapory firewall przejścia w warstwie aplikacji (application proxy firewall) wpuszcza ruch sieciowy z jednej strony i wypuszcza z drugiej po przejsciu pakietów przez oprogramowanie przejścia w warstwie aplikacji (*application proxy software*). Przejście to (*proxy*) analizuje wszystkie przechodzace przez nie dane i odrzuca nie autoryzowane i niebezpieczne pakiety danych. Gdy komputer spoza chronionej sieci komunikuje sie z hostem wewnatrz, proxy imituje tego hosta. Podobnie, gdy wewnętrzny host laczy sie z zewnętrzny klientem, proxy maskuje pochodzenie komputera, który zainicjalował polaczenie. W wyniku tego hosty w wewnętrznej sieci nie sa nigdy ujawniane na zewnatrz. Rysunek 11.3 przedstawia konfiguracje takiej zapory.

Rysunek 11.3.

*Konfiguracja
zapory przejścia
w warstwie aplikacji*



Zapory przejścia w warstwie aplikacji mają kilka zalet:

- ◆ Zapewniają najwyższy poziom bezpieczeństwa, ponieważ nie pozwala komputerom po obu końcach połączenia komunikować się ze sobą bezpośrednio.
- ◆ Zapewniają najlepsze zdolności filtrowania.
- ◆ Uwierzytelniają użytkowników i rejestrują zdarzenia w sposób pełny, zapewniając w ten sposób wysoki poziom bezpieczeństwa.
- ◆ Ich działanie opiera się na zasadach, które można łatwo konfigurować. Dzięki temu zapory te łatwiej jest konfigurować niż zapory badające stan pakietów, które korzystają z reguł filtrowania pakietów.

Korzystanie z zapór przejścia w warstwie aplikacyjnej ma też kilka wad:

- ◆ Są najwolniejszymi zaporami z wszystkich trzech typów.
- ◆ Dla każdego protokołu wymagają dodatkowego oprogramowania proxy.
- ◆ Opierają się na protokole TCP i nie obsługują UDP.
- ◆ Wymagają zmian konfiguracji wszystkich wewnętrznych hostów.
- ◆ Są najdroższym typem zapór firewall.



Granice podziału pomiędzy różnymi urządzeniami zapór praktycznie nie istnieją. Na przykład, możemy zakupić dla routera firmy Cisco dodatkowe oprogramowanie zapory firewall, uruchamiane w routerze, które pozwala routowi grać role zapory.

Najczęściej stosowane konfiguracje sieci z zaporami firewall

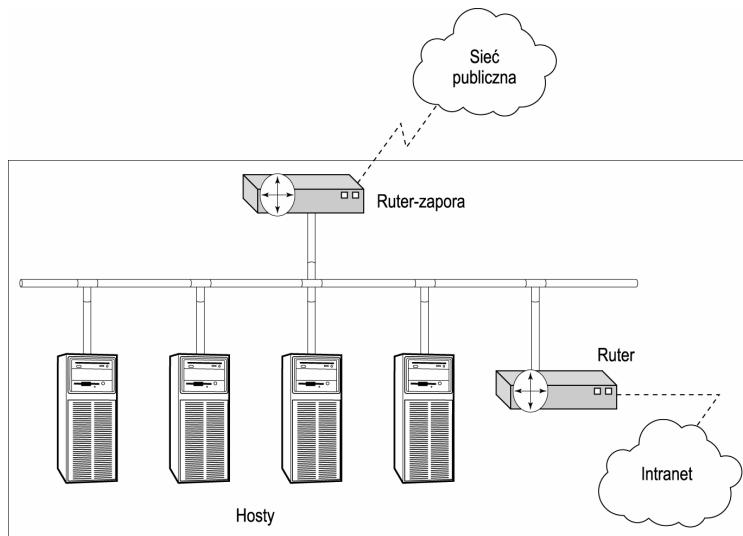
Istnieją dwie popularne konfiguracje sieci z zaporami firewall: *prosty system zapory*, używający routerów, oraz *trzyczęściowy system zapory*, składający się z trzech warstw (części).

Prosty system zapór

Rysunek 11.4 przedstawia prostą topologię zapory, skonstruowaną z wykorzystaniem routerów.

Rysunek 11.4.

*Prosta siec
z zapora firewall*



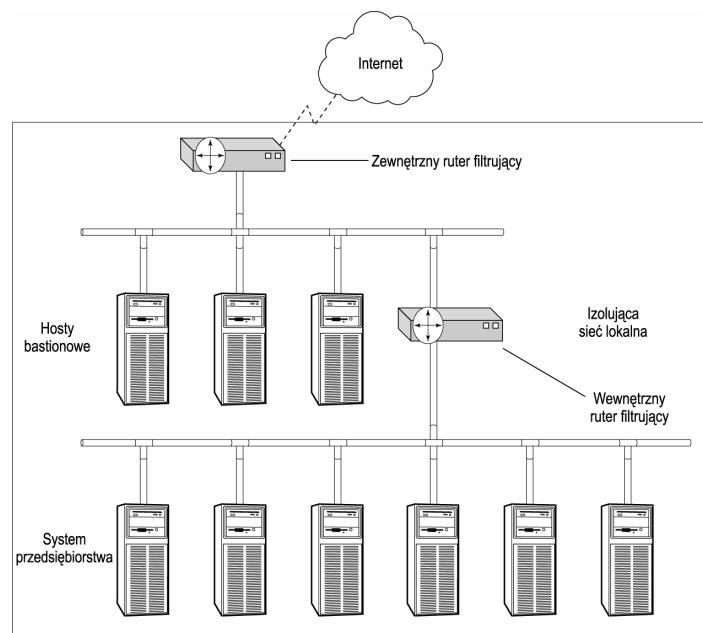
W tej konfiguracji ruter-zapora umieszczony jest na wyjściu prywatnej sieci, które łączy ją ze światem zewnętrznym. Każdy wchodzący i wychodzący pakiet danych musi przejść przez zapory na ruterze. Zabezpieczenie takie trzeba zastosować w każdym miejscu, gdzie sieć łączy się z innymi sieciami publicznymi i Internetem.

Trzyczesciowy system zapory

Rysunek 11.5 przedstawia konfigurację klasycznego systemu trzyczesciowej zapory firewall. System ten składa się z trzech wyspecjalizowanych warstw:

Rysunek 11.5.

*Klasyczna konfiguracja
trzyczesciowego
systemu
zapory firewall*



- ♦ *Izolująca sieć lokalna* — ta warstwa gra role bufora pomiędzy intranetem przedsiębiorstwa a sieciami nie zaufanymi. Izolująca sieć lokalna (LAN) otrzymuje unikatowy numer sieci, różny od numeru intranetu przedsiębiorstwa. Jedynie izolująca sieć lokalna jest widoczna dla sieci zewnętrznej.



Izolująca sieć lokalna nazywana jest *strefą zdemilitaryzowaną* (DMZ — *Demilitarized Zone*).

- ♦ *Wewnętrzny filtr pakietów* — ruter (lub zestaw ruterów), który filuluje pakiety przechodzące pomiędzy izolującą siecią lokalną i intranetem przedsiębiorstwa.
- ♦ *Zewnętrzny filtr pakietów* — ruter (lub zestaw ruterów), który filuluje pakiety przechodzące pomiędzy izolującą siecią LAN i światem zewnętrznym.



Jesli użytkownicy przedsiębiorstwa potrzebują dostępu do usług internetowych, należy zezwolić na wychodzący ruch TCP, lecz jedynie pod warunkiem, iż pakiety TCP będą odpowiedziami na wysłane wcześniej poprawne zadania. Nowy nadchodzący ruch TCP należy blokować, ponieważ może być inicjowany przez hakerów, chcących nawiązać sesje z jednym z hostów w sieci przedsiębiorstwa.

Podczas tworzenia trzyczesciowego systemu zapory należy postępować według następujących reguł:

- ♦ Wewnętrzny i zewnętrzny filtr pakietów powinny przepuszczać nadchodzące pakiety tylko wtedy, gdy należą do otwartej uprzednio sesji.
- ♦ Zewnętrzny filtr pakietów powinien przepuszczać pakiety kierowane do hostów bastionowych.
- ♦ Zewnętrzny filtr pakietów nie powinien posiadać niepotrzebnych usług i połączeń.
- ♦ Jesli to możliwe, na zewnętrznym ruterze filtrującym pakiety należy:
 - ♦ stosować tylko trasy statyczne,
 - ♦ całkowicie zablokować usługę TFTP,
 - ♦ wyłączyć usługi finger, proxy ARP, przekierowanie IP, buforowanie tras IP oraz telnet,
 - ♦ stosować szyfrowanie hasel,
 - ♦ unikać stosowania w roli zewnętrznego filtru pakietów serwera MacIP, który udostępnia połączenia IP przez protokół AppleTalk.
- ♦ Ruch z ruterów zapory firewall do przedsiębiorstwa powinien być blokowany, ponieważ te ruterzy mogą stać się ofiarą ataku hakerów. Jesli zablokujemy cały ruch z zapory do sieci, prawdopodobieństwo ataku na sieć stanie się niskie.
- ♦ Ruter zapory i hosty bastionowe powinny zawierać jak najmniej oprogramowania, które na dodatek nie powinno być złozone. Skomplikowane aplikacje zwykle zawierają wiele błędów, tworzących luki w zabezpieczeniach.

Zapory są rozwiązaniem skutecznym — lecz kosztownym. Ponadto wymagają obsługi ekspertów, ponieważ ich implementacja i utrzymanie są trudne. Z tych powodów zapo-

ry firewall często sa poza zasięgiem małych firm i użytkowników laczacych sie z Internetem z domu. W przeciwnieństwie do nich *tlumaczenie adresów sieciowych* (NAT — *Network Address Translation*) daje tania ochronę, która nie wymaga złożonej instalacji. NAT jest potężnym mechanizmem, pomagającym oszczędzać będące już na wyczerpaniu zarezerwowane adresy IP dla dużych sieci oraz upraszczającym zarządzanie adresowaniem IP.

Stosowanie NAT

Aby laczyc sie z Internetem, kazdy komputer potrzebuje unikatowego adresu IP. Jednakże liczba hostów przylaczonych do Internetu wciąż rośnie wykładniczo, co oznacza, że niedobory adresów IP są już blisko. Rozwiązaniem tego problemu jest IPv6, lecz zaimplementowanie tego protokołu zajmie kilka lat, ponieważ wymaga modyfikacji całej istniejącej infrastruktury Internetu. Tymczasem niezbedne jest rozwiązanie zastępcze.

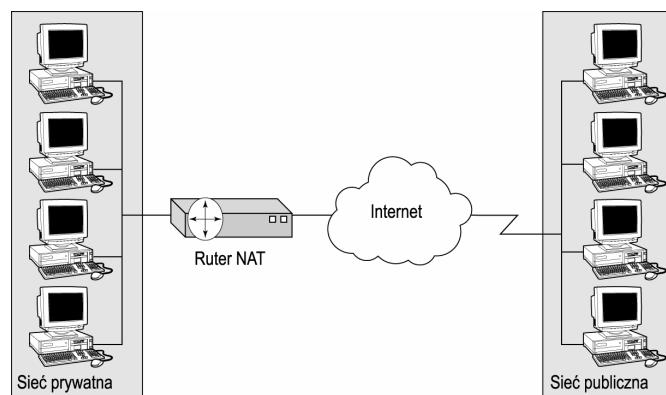
Mechanizm NAT został opracowany przez firmę Cisco — początkowo jako mechanizm trasowania, oszczędzający zarejestrowane adresy IP w dużych sieciach. Z czasem stał się również skuteczna metoda ochrony sieci prywatnych przed nieupoważnionym dostępem z zewnątrz. NAT pozwala nałączenie z Internetem i innymi publicznymi sieciami prywatnych sieci, które stosują adresy nie zarejestrowane w InterNIC lub innej agencji rejestrującej, co odbywa się za pomocą tłumaczenia adresów prywatnych na adresy IP zarejestrowane globalnie.



Mechanizm NAT jest dobrym sposobem rozwiązania problemów z wyczerpaniem adresów i skalowaniem, ponieważ jego implementacja wymaga bardzo niewielu zmian i może odbywać się etapami. Mechanizm ten posiada jednak wiele cech negatywnych, przez które nie nadaje się do roli rozwiązania długoterminowego.

NAT musi być zainstalowany na styku pomiędzy siecią prywatną i resztą świata. Takim punktem styku jest router. Kazde urządzenie zdolne do obsługi NAT posiada tablice translacji, która służy do tłumaczenia prywatnych adresów IP na adresy IP unikatowe globalnie. Jeżeli sieć ma kilka wyjść, bardzo ważne jest, aby wszystkie routery obsługujące NAT posiadały identyczne tablice translacji. Rysunek 11.6 przedstawia konfigurację NAT.

Rysunek 11.6.
Konfiguracja NAT





Mechanizm NAT jest zasadniczo używany przez routery, lecz może też być stosowany przez zapory firewall. Routery używające NAT nazywane są czasami *ruterami NAT* lub *translatorami adresów sieciowych*.

Przez zaimplementowanie NAT automatycznie tworzymy zapory firewall pomiędzy siecią prywatną i światem zewnętrznym. Urządzenie używające mechanizmu NAT gra rolę agenta pomiędzy siecią prywatną i resztą świata. Inaczej mówiąc, pojedynczy adres IP może posłużyć do reprezentowania całej sieci, co dodatkowo zwiększa jej bezpieczeństwo przez ukrycie wewnętrznych adresów IP przed światem zewnętrznym.

W NAT jedynie połączenia zainicjowane wewnętrznie sieci mają prawo przejść przez ruter. Dzięki temu komputer wewnętrzny może z powodzeniem łączyć się z komputerem położonym na zewnątrz swojej sieci. Jednakże zewnętrzny komputer nie będzie w stanie połączyć się z wewnętrzny, ponieważ musiałby zainicjować połączenie (na co NAT nie pozwala). W wyniku tego użytkownicy należący do sieci mogą przeglądać zasoby internetowe, łącząc się z komputerami należącymi do innych sieci publicznych, a nawet pobierając pliki, lecz kontakt z zewnątrz z wewnętrznymi hostami za pomocą ich adresów IP jest niemożliwy.



Mozemy porównać NAT do sekretarki, która przekazuje tylko połączenia telefoniczne, których sobie życzymy, natomiast niepożądane połączenia odrzuca.

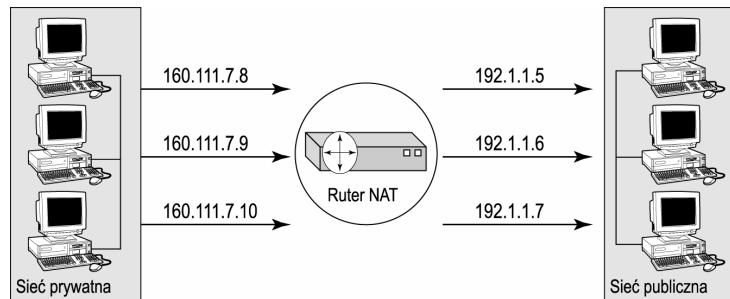
W razie potrzeby administrator sieci prywatnej może, w sposób kontrolowany, zezwolić klientom z zewnątrz na dostęp do usług typu FTP lub WWW w określonych hostach wewnętrznych. Robi się to za pomocą przypisania dobrze znanych portów TCP do adresów wewnętrznych — proces taki nosi nazwę *odwzorowania wejściowego (inbound mapping)*.

Istnieją dwie formy mechanizmu NAT: *statyczny* i *dynamiczny* oraz dwa typy dynamycznego NAT: *przeciążony (overloaded)* i *nakładany (overlapped)*.

- ◆ *Statyczny NAT* — nie zarejestrowane adresy IP są kojarzone z zarejestrowanymi jednym do jednego, jak na rysunku 11.7. Na przykład, adres 160.111.7.8 będzie zawsze tłumaczony na 192.1.1.5. Ta forma jest stosowana, gdy trzeba łączyć się z wewnętrzny hostem spoza danej sieci.

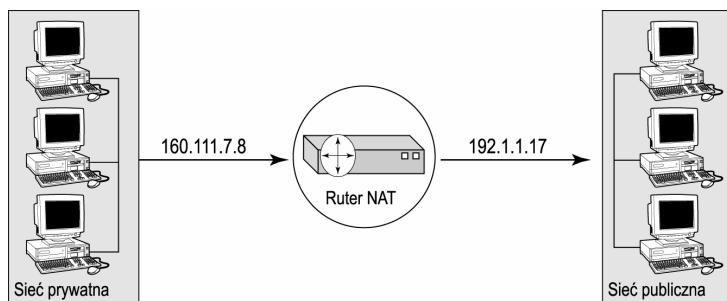
Rysunek 11.7.

Statyczny NAT



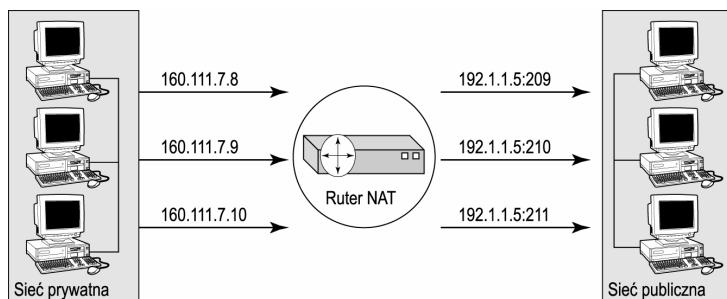
- ◆ *Dynamiczny NAT* — nie zarejestrowany adres IP może zostać odwzorowany na dowolny pierwszy dostepny adres z bloku zarejestrowanych adresów IP, jak na rysunku 11.8. Na przykład, 160.111.7.8 będzie tłumaczony na pierwszy dostepny adres z zakresu od 192.1.1.1 do 192.1.1.120. Ta forma NAT jest stosowana, gdy wewnętrzny host musi mieć dostęp do zewnętrznego.

Rysunek 11.8.
Dynamiczny NAT

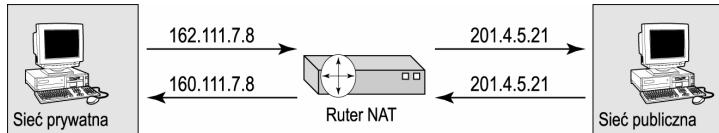


- ◆ *Przeciążony NAT dynamiczny* — wiele nie zarejestrowanych adresów IP odwzorowuje się na pojedynczy zarejestrowany adres IP za pomocą różnych portów, jak na rysunku 11.9. Na przykład, adres 160.111.7.8 będzie tłumaczony na 192.1.1.5:209, a 160.111.7.10 na 192.1.1.5:210. Przeciążony NAT dynamiczny nazywany jest również *tłumaczaniem adresów portów* (PAT — *Port Address Translation*), *NAT multipleksowanym na poziomie portów* (*port-level multiplexed NAT*) oraz *NAT jednoadresowy* (*single address NAT*).

Rysunek 11.9.
Przeciążony NAT dynamiczny



- ◆ *Nakładany NAT dynamiczny* — jeśli w prywatnej sieci używany jest blok adresów IP legalnie zarejestrowanych dla innej sieci, routery NAT muszą przechwycić je i zastąpić zarejestrowanymi adresami unikatowymi. W przeciwnym razie pakiety danych mogłyby zostać utracone z powodu obecności w Internecie dwóch hostów o identycznym adresie IP. Zalóżmy na przykład, że sieć prywatna używa bloku adresów 162.111.xxx.xxx. Ten sam blok został jednak przydzielony przez internetową agencję rejestrującą inną sieci. Aby uniknąć możliwości kolizji adresów, ruter NAT powinien przetłumaczyć nie zarejestrowany adres IP na zarejestrowany, gdy wewnętrzny host będzie musiał komunikować się z hostem należącym do innej sieci. Analogicznie, ruter NAT musi przetłumaczyć tez zarejestrowane globalnie adresy IP na nie zarejestrowane adresy IP używane w sieci prywatnej, gdy zewnętrzny host wysła informacje do wewnętrznego. Nakładany NAT dynamiczny został przedstawiony na rysunku 11.10.

Rysunek 11.10.*Nakładany NAT dynamiczny*

Korzyści ze stosowania NAT

NAT znaczaco upraszcza zadania zwiastane z zarządzaniem adresami IP. W sieci TCP/IP kazdy komputer musi miec odrebowe skonfigurowane: poprawny adres IP, maski podsieci, nazwe domeny, ruter domyslny i adres serwera DNS. Im wieksza jest siec, tym trudniej skoordynowac dystrybucje adresów. Co wiecej, niewlasciwa konfiguracja chocby pojedynczego komputera moze prowadzic do zatrzymania funkcjonowania cześci sieci. NAT moze znaczaco zmniejszyc nakladы pracy administratorów. Inne korzyści ze stosowania tego mechanizmu sa nastepujace:

- ◆ NAT moze posluzyc do podzialu duzej sieci na wiele mniejszych. Komputery mozna dodawac, usuwac i zmieniac im adresy bez wpływu na inne jednostki.
- ◆ NAT jest zadaniem ruterów, wobec czego jedynie rutery w punktach wyjsciowych sieci wymagaja modyfikacji.
- ◆ NAT moze obslugiwac protokół DHCP, który pozwala na automatyczna aktualizacje informacji zwiastanych z TCP/IP we wszystkich komputerach. Dzieki temu administrator nie musi modyfikowac recznie wspomnianych informacji w kazdym komputerze w sieci.
- ◆ NAT pozwala administratorom zabranic dostepu do lokalizacji potencjalnie niebezpiecznych lub zawierajacych wątpliwe tresci.
- ◆ NAT udostepnia funkcjonalosc rejestrowania ruchu, co pomaga administratorom w wyszukiwaniu uzytkowników, lokacji i polaczen sieciowych powodujacych problemy.
- ◆ NAT przewaznie nie wykorzystuje stoso protokolów komputera, wobec czego jest mniej podatny na ataki przez protokoly niskiego poziomu, np. „SYN Flood” i podobne.
- ◆ NAT obsluguje filtrowanie i trasowanie na poziomie pakietów, zapewniajac wysoki poziom bezpieczenstwa hostów wewnętrznych.

Oprócz wszystkich tych korzyści NAT posiada pewne negatywne cechy:

- ◆ W przypadku globalnych sieci duzych przedsiębiorstw korzystanie z NAT nie jest zalecane, poniewaz duza liczba hostów moze chciec komunikowac sie ze soba, co zwiększa rozmiary tablic translacji.
- ◆ Pakiety danych aplikacji zawierajace adresy IP nie beda dzialac z NAT, o ile NAT nie bedzie rozpoznawac takich przypadkow i posiadac odpowiedniego mechanizmu dokonujacego odpowiednich translacji.
- ◆ W prawdziwe NAT obsluguje szyfrowanie na poziomie aplikacji, lecz nie wspiera szyfrowania sumy kontrolnej nagłówka, co zmniejsza liczbe dostepnych opcji zabezpieczeń.

- ◆ NAT moze utrudnic wykrycie naruszenia bezpieczenstwa. Na przyklad, jesli wewnętrzny lub zewnętrzny host zaatakuje innego hosta lub wysle duza liczbe bezwartosciowych informacji, namierzenie źródła kłopotów moze byc trudne, poniewaz adres IP hosta jest ukryty.

W przeciwnieństwie do NAT, z reguły implementowanego w ruterach, *serwer proxy* nie wymaga trasowania, czyli inaczej mówiąc, nie wymaga określonego sprzętu. Moze byc zainstalowany na dowolnym komputerze spełniającym odpowiednie wymagania. Serwer proxy daje administratorom sieci mozliwosc „ukrycia” sieci przed resztą świata i ochronę cennych danych przesyłanych przez te sieci.



Wiecej informacji o NAT mozna znalezc w RFC 1631 i 2663.

Przezroczysty czy nieprzezroczysty

Poniewaz działanie NAT i działanie proxy warstwy aplikacji sa do siebie bardzo podobne, jedno jest często mylone z drugim. Sa jednak różnice:

- ◆ NAT jest przezroczysty dla punktów koncowych komunikacji (wewnętrznych i zewnętrznych). Oznacza to, ze podczas transakcji ani hosty wewnętrzne, ani zewnętrzne nie „wiedzą” o istnieniu posrednika. Serwer proxy natomiast nie jest dla wewnętrznych hostów przezroczysty. Wprawdzie wewnętrzne hosty „wiedzą” o istnieniu oprogramowania proxy, lecz zewnętrzne nie — poniewaz oprogramowanie proxy nasładuje wewnętrznego hosta.
- ◆ Jesli mechanizm NAT funkcjonuje w ruterze granicznym lub zaporze firewall, wewnętrznych hostów nie trzeba konfigurować do korzystania z niego. Z drugiej strony, wszystkie hosty wewnętrzne muszą posiadać skonfigurowane informacje związane z proxy.
- ◆ NAT działa w warstwie 3. modelu odniesienia OSI, zaś serwery proxy w warstwie 4. i wyższych.
- ◆ NAT, jako protokół niskiego poziomu, jest o wiele szerszy od części serwerów proxy.



Proszę nie mylić NAT z serwerami proxy.

Wykorzystanie serwera proxy

Serwer proxy pełni funkcje bramy pomiędzy siecią prywatną i sieciami publicznymi, włącznie z Internetem. *Brama (gateway)* jest programem (aplikacją) lub komputerem z uruchomionym specjalnym oprogramowaniem, który gra role bariery pomiędzy dwiema sieciami, a jednocześnie umożliwia komunikację pomiędzy nimi.

Serwer proxy jest aplikacja sieciowa, skonfigurowana tak, by działała w imieniu wyznaczonej sieci. Gdy aplikacja uruchomiona w wewnętrzny hostie wysyła zadanie danych na zewnątrz sieci, serwer proxy przechwytuje zadanie, tłumaczy je i przesyła do sieci docelowej. Gdy zewnętrzny host musi połączyć się z hostem wewnętrzny, serwer proxy ponownie przechwytuje zadanie, sprawdza czy zawarte w nim dane są bezpieczne, a następnie przekazuje pakiet danych do docelowego hosta wewnętrznego. Dla hostów zewnętrznych zawsze wygląda to tak, jakby zadania i odpowiedzi pochodząły od serwera proxy. W ten sposób wewnętrzny host jest zawsze ukryty przed światem zewnętrzny.



Uslugi proxy mogą, lecz nie muszą, dokonywać tłumaczenia adresów sieciowych (NAT).

Serwer proxy dodatkowo utrzymuje pamięć podręczną najnowszych zadań. Gdy host — zewnętrzny lub wewnętrzny — zada informacji, które były ostatnio pobierane, serwer proxy spełnia zadanie korzystając z pamięci podręcznej, zamiast ponawiając zadanie wysypane do hosta docelowego. Chroni to dodatkowo sieć i przyspiesza transakcje.

Do wad serwera proxy można zaliczyć jego nieprzezroczystość dla użytkowników i konieczność konfigurowania wszystkich wewnętrznych hostów, by mogły z niego korzystać. Zwiększa to znacznie nakłady pracy na administrowanie. Z drugiej strony, serwer proxy ma kilka zalet:

- ◆ Konfiguując hosty wewnętrzne do korzystania z serwera proxy, możemy dostarczyć aplikacje internetowe do wszystkich komputerów w sieci.
- ◆ Wykorzystanie pamięci podręcznej serwera proxy może znacznie poprawić wydajność i bezpieczeństwo sieci prywatnej.
- ◆ Konfiguując w serwerze proxy dla wewnętrznych użytkowników zezwolenia i odmowy dostępu do zasobów na zewnątrz, porty lub domeny, możemy zapewnić wewnętrznym użytkownikom bezpieczeństwo dostępu. „Niebezpieczne” lokalizacje i witryny WWW możemy z łatwością zablokować.



Dodatkowe informacje o serwerach proxy można znaleźć w dokumentach RFC 1445, 1906, 2607, 2616 i 2843.

Aby sprostać konkurencyjności na dzisiejszym rynku, firmy muszą być połączone do Internetu. Jednakże dla wielu osób i małych firm utrzymywanie wielu połączeń internetowych może być kosztowne. Rozwiązaniem może być współdzielenie pojedynczego łączna internetowego przez sieć domową, biuro domowe lub małe firmy.

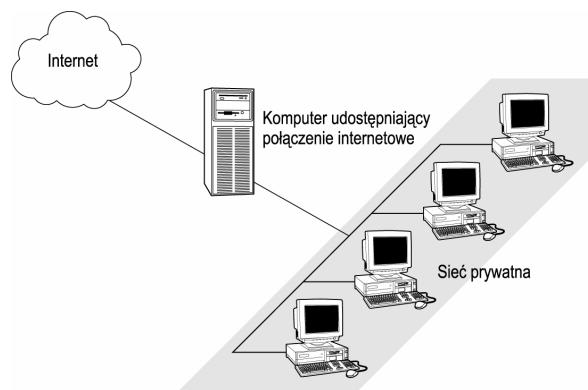
Udostępnianie połączenia internetowego Microsoftu

Microsoft dołączył usługę *Udostępnianie połączenia internetowego* (ICS — *Internet Connection Sharing*) do drugiej edycji Windows 98. Usługa ta pozwala kilku użytkownikom wspólnie korzystać z jednego połączenia z Internetem w obrębie sieci domowej lub malego biura. Po zainstalowaniu ICS w komputerze, zostaje on komputerem udostępniającym połączenie internetowe. Komputer ten musi posiadać połączenie z Internetem.

tem. Po zainstalowaniu ICS komputer potrafi udostepniać prywatne adresy IP oraz usługi rozwijania nazw reszcie komputerów w sieci. Rysunek 1.11 przedstawia laczosc, która daje komputer ICS reszcie komputerów w malej sieci.

Rysunek 1.11.

Typowa konfiguracja ICS



Gdy komputer z sieci laczy sie z Internetem, jego adres IP jest przekazywany do komputera ICS. Ten tłumaczy otrzymany prywatny adres IP na własny, globalnie unikatowy adres IP. Zadanie zostaje następnie przeslane do Internetu. Po otrzymaniu odpowiedzi na wyslane zadanie, komputer ICS tłumaczy adres IP z powrotem na adres prywatny oryginalnego autora zadania i przesyła dane do niego.

Poza udostepnieniem laczosci z Internetem kilku komputerom, ICS ukrywa reszte sieci przed Internetem i innymi sieciami publicznymi. Jedynym komputerem widocznym z zewnatrz jest komputer ICS. Zaden z pozostałych komputerów w sieci nie ma bezpośredniego polaczenia ze swiatem zewnetrznym. Inaczej mówiac, ICS jest skutecznym i tanim rozwiazaniem zabezpieczajacym male sieci.

Wprawdzie na rynku dostepnych jest obecnie wiele produktów innych firm, pozwalajacych na wspólne uzytkowanie polaczenia internetowego, lecz ICS Microsoftu z kilku powodów ma zdecydowaną przewagę:

- ◆ ICS Microsoftu jest wbudowany w systemy operacyjne Windows 98 i nowsze, wobec tego do polaczenia z Internetem nie musimy kupowac dodatkowego oprogramowania.
- ◆ ICS jest przyjazny dla uzytkownika. Inaczej mówiac, aby skonfigurowac i uruchomic polaczenie, nie jest wymagana pomoc eksperta.

Zagrozenia zwiazane z laczeniem sie z Internetem spowodowaly pojawienie sie licznych metod i mechanizmów zabezpieczajacych, na przyklad zapór firewall, serwerów proxy i NAT. Najnowszym produktem z dziedziny bezpieczenstwa sieciowego sa *wirtualne sieci prywatne*.

Wirtualne sieci prywatne

Prywatne sieci na potrzeby bezpiecznej laczosci pomiedzy rozrzuconymi lokacjami sieci przedsiębiorstwa uzywaja wydzielonych linii dzierzawionych. W przeciwienstwie do tego mechanizmu, technologia wirtualnych sieci prywatnych (VPN — Virtual

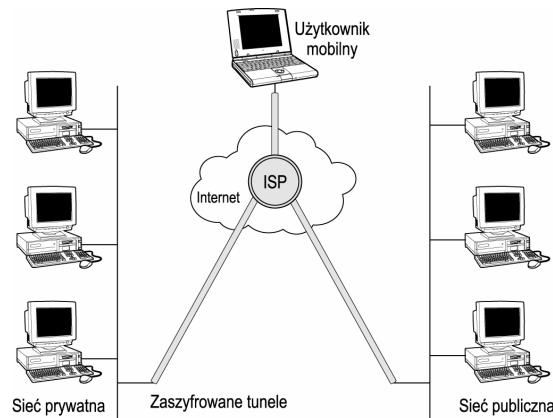
Private Networks) zapewnia bezpieczne połaczenie pomiędzy rozproszonymi jednostkami sieci przedsiębiorstwa za pomocą sieci publicznych, w tym Internetu, nie wymagając zarazem stosowania kosztownych linii dzierżawionych lub trwałych obwodów wirtualnych PVC (*Permanent Virtual Circuit*). Zamiast nich sieci VPN stosują na swoje potrzeby otwartą, rozproszoną infrastrukturę Internetu. Wirtualne sieci prywatne opierają się na protokole IP i stosują szyfrowanie i tunelowanie, aby udostępniać:

- ◆ *bezpieczny dostęp zdalny* w ramach sieci przedsiębiorstwa,
- ◆ *dostęp typu intranetowego*, łącząc sieci oddziałów lokalnych w sieci przedsiębiorstwa,
- ◆ *dostęp typu ekstranetowego*, umożliwiając korzystanie z zasobów intranetowych firmy partnerom, klientom i dostawcom.

Inaczej mówiąc, VPN zachowuje się z punktu widzenia bezpieczeństwa i funkcjonowania jak sieć prywatna. Wirtualne sieci prywatne zwiększą jednak zasięg sieci prywatnej za pomocą relacji zaufania, bez ryzykowania zabezpieczeń przy połaczeniach zdalnych. Rysunek 11.12 przedstawia typowa konfigurację VPN.

Rysunek 11.12.

Wirtualna sieć prywatna (VPN)



Aby stworzyć VPN, organizacja musi podłączyć się do lokalnych punktów połączenia — tzw. *punktów obecności* (POP — *Point-of-Presence*) w sieci swojego ISP. Za szczegóły połączenia i przesyłanie danych przez Internet do odpowiedniego miejsca przeznaczenia odpowiada dostawca usług internetowych. Ponieważ dane muszą podróżować do miejsca przeznaczenia przez Internet, administrator VPN musi zastosować odpowiednie środki szyfrowania danych przesyłanych pomiędzy dwiema sieciami. Zabezpieczy to dane przed podsłuchiwaniem i manipulacjami ze strony nieupoważnionych użytkowników.

Ponieważ wirtualne sieci prywatne nie używają wydzielonych linii dzierżawionych ani łączy WAN, zas dane firmy muszą być przesyłane przez Internet, sieci te zapewniają bezpieczeństwo danych za pomocą czterech krytycznych funkcji:

- ◆ *Uwierzytelnianie* — po otrzymaniu pakietu VPN weryfikuje, czy dane pochodzą z zaufanego źródła.
- ◆ *Poufność* — sieci VPN stosują różne metody szyfrowania, na przykład *kryptografię z kluczem publicznym i prywatnym*, aby zapobiec odczytowi i kopiowaniu danych podczas transmisji. W kryptografii klucza prywatnego nadawca szyfruje komunikat

za pomoca swojego klucza prywatnego i klucza publicznego odbiorcy. Po odebraniu zaszyfrowanego komunikatu odbiorca odszyfrowuje go za pomoca własnego klucza prywatnego i klucza publicznego nadawcy.

- ♦ *Integralnosc* — VPN zapewniają dodatkowo, iż dane nie ulegną modyfikacji podczas transmisji. W tym celu wirtualne sieci prywatne stosują *funkcje mieszanina, kody uwierzytelnienia wiadomości* i podpisy cyfrowe.
- ♦ *Funkcje mieszanina (hash functions)* — funkcja mieszanina generuje wartość mieszaną (*hash value*) pliku przed wysłaniem. Wartość ta utrudnia utworzenie pliku, który odpowiadalby wartości mieszanej dostarczanej z nadsyłanymi pakietami. Po odebraniu pakietu danych lub pliku odbiorca oblicza wartość mieszaną i porównuje ją z wartością nadawaną przez nadawcę. Jeśli obie wartości nie są identyczne, to dane uznaje się za uszkodzone i odrzuca. Do przykładów algorytmów z mieszaniem należą MD5, RIPE-MD-160 oraz SHA-1.
- ♦ *Kody uwierzytelniania wiadomości (MAC — Message Authentication Code)* — dodaje klucz do funkcji mieszaniny. Po wygenerowaniu wartości mieszanej obliczany jest MAC i dodawany do danych. Odbiorca oblicza własny MAC i porównuje z wysłana wartością. Gdy wartości te nie są identyczne, pakiet zostaje odrzucony.
- ♦ *Podpisy cyfrowe* — nadawca „podpisuje” pakiet własnym kluczem prywatnym. Po odebraniu pakietu odbiorca weryfikuje popis za pomocą klucza publicznego nadawcy, wysłanego przez niego razem z komunikatem.
- ♦ *Kontrola dostępu* — VPN wymaga procesu logowania, aby zapobiec dostępowi do sieci ze strony nieautoryzowanych użytkowników. Do uwierzytelniania użytkowników i kontroli dostępu do zasobów sieciowych mogą posłużyć protokoły CHAP (*Challenge Handshake Authentication Protocol*), RADIUS (*Remote Authentication Dial-In User Service*) oraz zetony generowane sprzętowo.

Zaimplementowanie VPN może przynieść organizacji wiele korzyści. Do najważniejszych z nich należą:

- ♦ *Oszczędności* — wirtualne sieci prywatne mogą korzystać z różnych technologii i usług oferowanych przez dostawców usług sieciowych. Dzięki temu firmy nie muszą stosować kosztownych łącz dzierżawionych, banków modemów lub technologii *frame relay* (przekazywania ramki), aby połączyć oddalone części intranetu z główną siecią przedsiębiorstwa. Zmniejsza to wewnętrzne zapotrzebowanie na zasoby i personel obsługi technicznej i równoczesnie redukuje koszty sieci.
- ♦ *Optymalne wykorzystanie przepustowości* — sieci VPN nie utrzymują stałych łącz pomiędzy punktami końcowymi komunikacji. Połączenie tworzone jest po zgłoszeniu autentycznego zadania, dlatego sieci noszą nazwę *wirtualnych* sieci prywatnych. Po zakończeniu transakcji połączenie jest usuwane. W wyniku zasoby sieciowe zostają zwolnione dla innych połączeń i przepustowość sieci jest wykorzystywana do maksimum.
- ♦ *Bezpieczeństwo* — każda VPN do ochrony przesyłanych danych przed manipulacją stosuje zaawansowane metody szyfrowania, na przykład kriptografię z kluczem publicznym i prywatnym. Inne metody, na przykład funkcje mieszaniny, kody

uwierzytelniania wiadomości i podpisy cyfrowe, służą do zapewnienia integralności danych wysyłanych i odbieranych za pomocą sieci publicznych. VPN stosują dodatkowo protokły uwierzytelniające, aby chronić sieć przed nieupoważnionym dostępem.

- ◆ *Skalowalność* — przedsiębiorstwa skutecznie współpracują z dostawcami usług internetowych, dzięki czemu są w stanie korzystać z ogromnej ilości zasobów komputerowych bez konieczności inwestowania w infrastrukturę. Zmniejsza to nakłady pracy na tworzenie i zarządzanie połączeniami WAN.
- ◆ *Obsługa użytkowników mobilnych* — zamiast polegać na kosztownych i niezbyt niezawodnych zasobach, takich jak banki modemów i serwery dostępu zdalnego, co może prowadzić do olbrzymich rachunków za połączenia telefoniczne, mobilni sprzedawcy i osoby pracujące zdalnie (z domu) mogą łączyć się z intranetem firmy za pomocą szybkich łączysDSL i kablowych. Pozwala to zwiększyć elastyczność i wydajność sieci.

Dodatkowe informacje o wirtualnych sieciach prywatnych zawiera RFC 2547, 2637 i 2684.



Oprócz zaawansowanych metod szyfrowania, sieci VPN stosują do ochrony pakietów przed ciekawością osób z zewnątrz *tunelowanie* — dwupunktowe połączenia intranetowe, przechodzące przez Internet i inne sieci zewnętrzne. Tunelowanie pozwala nadawcy opakować pakiety danych w taki sposób, że informacje o trasie i przekształcaniu ukryte są przed odbiorcą. Do najlepiej znanych protokołów tunelowania, używanych do tworzenia VPN, należą PPTP (*Point-to-Point Tunneling Protocol*) Microsoftu oraz L2TP (*Layer-2 Tunneling Protocol*) firmy Cisco.

Innymi znanyimi protokołami VPN są L2F (*Layer-2 Forwarding*) oraz IPSec (*IP Security*), oba opracowane przez Cisco.



PPTP

Protokół tunelowania dwupunktowego (*Point-to-Point Tunneling Protocol*) Microsoftu był jednym z pierwszych protokołów udostępniających w telefonicznych VPN dostęp zdalny, który można tunelować przez Internet do lokacji docelowej. PPTP jest implementowany w systemach RAS (serwerach zdalnego dostępu), dzięki czemu RAS może w razie konieczności umożliwić sterowanie i zarządzanie połączeniami. Pozwala to na kontrolę zdalnych połączeń telefonicznych. Połączenia te mogą pochodzić zarówno z publicznej komutowanej sieci telefonicznej (PSTN — *Public Service Telephone Network*), jak i z sieci ISDN (*Integrated Services Digital Network* — sieć cyfrowa z integracją usług). PPTP może również inicjować połączenia wychodzące.

PPTP został opracowany na podstawie protokołu PPP (*Point-to-Point Protocol*), który jest najczęściej stosowanym protokołem dostępu zdalnego. Zamiast zmieniać całą strukturę PPP, wprowadzono w jego podstawowej strukturze kilka uzupełnień i zmian, aby dostosować protokół do wirtualnych sieci prywatnych.

PPTP składa się z dwóch elementów: *sterowania połączeniem* i *tunelu IP* pomiędzy VPN i twórcą połączenia:

- ◆ *Sterowanie połączeniem* — standardowa sesja TCP, która trzeba nawiązać pomiędzy hostem docelowym VPN i komputerem wywołującym połączenie, zanim będzie można utworzyć pomiędzy nimi tunel. W tej sesji przekazywane są dane sterowania i zarządzania połączeniem.
- ◆ *Tunel IP* — po pomyslnym nawiązaniu połączenia tworzony jest *tunel* (kanal wirtualny) pomiędzy nadawcą i odbiorcą. Utworzenie tunelu jest konieczne, aby zapewnić powodzenie sesji wymiany danych, ponieważ właśnie przez tunel będą przesyłane pakiety PPP.

Ponieważ bezpieczny dostęp zdalny w PPTP został przejęty z PPP, metody uwierzytelniania w obu protokołach są identyczne: CHAP, MS-CHAP i PAP (*Password Authentication Protocol*). Ostatnio dodano jednak do PPTP bardziej skuteczną metodę szyfrowania — MPPE (*Microsoft Point-to-Point Encryption* — szyfrowanie dwupunktowe Microsoftu).

Pakiety PPTP przypominają pakiety PPP, z tą różnicą, że opakowane zostają za pomocą zmodyfikowanej wersji protokołu GRE (*Generic Routing Encapsulation*). Wykorzystanie GRE pozwala PPTP funkcjonować w warstwie 2. modelu odniesienia OSI (warstwie łącza danych). Dzięki temu PPTP może obsługiwać oprócz IP inne protokoły, na przykład IPX (*Internet Packet Exchange*), AppleTalk i NetBEUI (*NetBIOS Extended User Interface*). GRE zapewnia również usługę kontroli przepływu i zatorów opakowanych datagramów dla transportu pakietów PPP. Dzięki temu protokół PPTP jest bardziej elastyczny od swojego poprzednika — PPP.

PPTP posiada kilka zalet:

- ◆ umożliwia elastyczne zarządzanie adresami IP,
- ◆ obsługuje inne protokoły poza IP: m.in. AppleTalk i IPX.

Protokół ten posiada również kilka wad:

- ◆ nie obsługuje silnych metod szyfrowania dla ochrony danych,
- ◆ nie obsługuje metod uwierzytelniania użytkownika opartych na zetonach — metody te są skuteczniejsze od PAP i CHAP.

Gdy Microsoft zaproponował PPTP, firma Cisco — lider na rynku produktów sieciowych — opracowała protokół L2TP (*Layer-2 Tunneling Protocol* — protokół tunelowania w warstwie 2.), który został wypuszczony na rynek jako ulepszenie protokołu L2F (*Layer-2 Forwarding*), które miało zaraździć niedociagnięciom protokołu PPTP. L2TP został zaakceptowany przez IETF jako protokół standardowy i przechodzi dalsze modyfikacje jako następca PPTP.

Dodatkowe informacje o PPTP zawiera RFC 2637.



Uwaga

Layer-2 Tunneling Protocol

L2TP jest standardowym protokołem VPN, który łączy funkcjonalność PPTP i L2F. Podobnie jak PPTP, L2TP jako podstawa używa PPP, aby udostępniać telefoniczny dostęp zdalny, który można tunelować przez Internet do sieci docelowej. Różnica polega na tym, że L2TP stosuje własny protokół tunelowania. Ponieważ tunelowanie L2TP jest niezależne od IP, protokół ten może współpracować z różnymi nosikami, na przykład frame relay, ATM, X.25 i tak dalej.

Tak jak PPTP, L2TP używa PPP dla połączeń telefonicznych, wobec czego stosuje metody uwierzytelniania PPP — PAP i CHAP. Oprócz tego stosuje rozszerzalne metody uwierzytelniania udostępnione przez PPP, na przykład RADIUS. Uwierzytelnianie użytkownika jest jednak dwuprzestrzenne. Najpierw użytkownik jest uwierzytelniany przez ISP przed utworzeniem tunelu, a następnie po nawiązaniu połączenia uwierzytelniany jest ponownie w bramie przedsiębiorstwa.

Tunel L2TP opakuje ramki L2TP w pakiet UDP. Ten z kolei zostaje opakowany w pakiet IP. Zgodnie z zaleceniami Cisco, jeśli L2TP stosowany jest razem z IPSec, zwiększa się sila szyfrowania i poziom bezpieczeństwa z wykorzystaniem kluczy kryptograficznych w środowisku IP, dzięki czemu dane przesyłane tunelem L2TP są skutecznie chronione. Podstawowa różnica pomiędzy L2TP i PPTP jest definiowanie przez L2TP połączeń w tunelu, co pozwala na równoczesne używanie wielu połączeń w pojedynczym tunelu.

Wady L2TP to brak własnego silnego mechanizmu opakowywania — wymagany jest inny protokół, IPSec. W połączeniu protokoły te zapewniają mocną podstawę zabezpieczeń. Za to korzyści ze stosowania L2TP są duże, ponieważ:

- ◆ Protokół ten jest standardem, wobec czego dostawcy usług internetowych, klienci i administratorzy sieci nie muszą polegać na produktach pojedynczego dostawcy. Możliwe jest stosowanie szerokiego zakresu usług różnorodnych producentów.
- ◆ L2TP zapewnia większe bezpieczeństwo niż PPTP, ponieważ użytkownicy uwierzytelniani są na dwóch poziomach.
- ◆ L2TP może z powodzeniem działać w środowiskach nie korzystających z protokołu IP — AppleTalk, NetBEUI, IPX i innych.



IPSec jest protokołem firmy Cisco, który został po pewnym czasie zaakceptowany przez IETF jako standard. IPSec zapewnia wysoki poziom bezpieczeństwa VPN podczas przesyłania danych przez sieci publiczne, ponieważ obsługuje kriptografię z kluczem publicznym i uwierzytelnianie oparte na zetonach. Protokół IPSec został jednak zaprojektowany tylko dla pakietów IP, wobec czego nie nadaje się do innych środowisk. Dodatkowe informacje o IPSec zawierają RFC 2407 i 2409. L2TP został opisany w RFC 2261.

Rozdział 12.

Narzędzia

do obsługi plików

W tym rozdziale:

- ◆ Serwery NFS: przegląd, wersje i konfiguracja
- ◆ DFS: przegląd i konfiguracja w Windows 2000
- ◆ Narzędzia do przesyłania plików i protokół FTP

Podstawowym celem, dla którego łączymy komputery w sieci, jest udostępnianie zasobów — zwłaszcza plików. TCP/IP oferuje liczne narzędzia, których możemy w sieci używać do zarządzania plikami: NFS (*Network File System* — sieciowy system plików), DFS (*Distributed File System* — rozproszony system plików) oraz narzędzia do przesyłania plików: FTP (*File Transfer Protocol*), TFTP (*Trivial File Transport Protocol*) oraz rcp (*Remote Control Protocol*). Niniejszy rozdział oprócz omówienia tych narzędzi przedstawia również konfiguracje serwera NFS i systemu DFS w Windows 2000.

NFS

Sieciowy system plików NFS jest protokołem, który daje użytkownikom za pomocą protokołu TCP/IP przezroczysty dostęp do zasobów plikowych udostępnionych w sieci. Przezroczystość dostępu do zasobów plikowych oznacza, że użytkownicy mogą korzystać ze zdalnych plików i katalogów tak, jakby mieściły się w lokalnym systemie — bez konieczności logowania do odległego systemu.

Wprowadzenie do NFS

NFS zajmuje się łączeniem systemów plików w różnych komputerach w sieci, aby umożliwić użytkownikom przezroczysty dostęp do nich. Przykład pozwoli lepiej zrozumieć przezroczysty dostęp do plików: weźmy pod uwagę firmę, która posiada komputer o nazwie *stock1*. Komputer ten przechowuje raporty o zapasach magazynowych w katalogu *stocks/reports*. Dział sprzedawy posiada komputer o nazwie *sale1*, używany podczas sprzedaży. Dział ten potrzebuje regularnego dostępu do raportów o stanie magazynów z komputera *sale1*, aby sprawdzać, czy towar jest dostępny. Jeśli sieć oparta jest na TCP/IP, zas komputery mają zainstalowaną i uruchomioną usługę NFS, uzy-

kownicy w dziale sprzedazy beda mogli korzystac z raportow magazynowych bez koniecznosci logowania sie do komputera *stock1* lub kopiowania plikow do komputera *sale1*.

Aby to jednak umozliwic, trzeba w obu komputerach (*stock1* i *sale1*) wykonac odpowiednie czynnosci:

- ◆ W komputerze *stock1* trzeba udostepnic sprzedawcom folder */stocks/reports*, który zawiera raporty magazynowe. Udostepniajac katalog, administrator moze ograniczyc dostep dla wyznaczonych komputerow oraz umozliwic dostep z prawami zapisu i odczytu lub tylko odczytu. Komputer *stock1* pelni funkcje serwera NFS. Serwer jest komputerem, który udostepnia uslugi i zasoby.
- ◆ W komputerze *sale1* nalezy nawiiazac polaczenie pomiedzy udostepnionym katalogiem */stocks/reports* z komputera *stock1* a katalogiem, na przyklad */all_reports/stocks*, w lokalnym komputerze *sale1*. W srodowisku uniksowym proces ten nosi nazwe *montowania (mounting)*. Komputer *sale1* gra role klienta NFS. Klient jest komputerem, który wymaga dostepu do zasobow lub uslug innego procesu w innym komputerze w sieci. Katalog podlaczony do katalogu w serwerze NFS nazywany jest *katalogiem zamontowanym*.

Po zamontowaniu katalogu uzytkownicy w dziale sprzedazy otrzymaja dostep do raportow magazynowych ze swojego komputera *sale1*. Wystarczy, ze zmienia katalog roboczy na zamontowany katalog */all_reports/stocks* w lokalnym komputerze *sale1*, a beda mogli korzystac z plikow raportow, jakby miescily sie w komputerze lokalnym.

Serwery NFS sa *bezstanowe* — nie zachowuja zadnych informacji o stanie protokolu dla klientow NFS. Ta cecha serwera NFS jest korzystna w przypadku awarii serwera. Klient nie zdaje sobie sprawy, iz serwer nie dziala; jedynie ponawia probu polaczenia az do uzyskania odpowiedzi od serwera. Dzieki temu nie istnieja problemy z odzyskaniem stanu po stronie klientow.

NFS implementuje przezroczysty dostep do plikow za pomoca protokolu RPC (*Remote Procedure Call* — zdalne wywolanie procedur) przez standard XDR (*eXternal Data Representation* — zewnetrzna reprezentacja danych).

Protokol RPC

RPC jest protokołem uzywanym do komunikacji pomiedzy procesami w różnych komputerach sieciowych, poprzez system logicznej komunikacji klienta z serwerem. RPC korzysta z procedur zdalnych — programów, wymagajacych do komunikacji adresu hosta, numeru programu i numeru procedury.

Model RPC jest podobny do modelu uzywanego do komunikacji pomiedzy różnymi procesami w jednym komputerze. Proces RPC w komputerze klienta zada określonej usługi od innego komputera w sieci. Taki komunikat zadajacy usługi nazywany jest *komunikatem wywołania (call message)*. Serwer uwierzytelnia zadanie, a następnie sam udostępnia usługę lub uruchamia w tym celu inny proces. Na koniec serwer zwraca odpowiedź do klienta — *komunikat odpowiedzi (reply message)*. Kazdy komunikat wywołania jest parowany z komunikatem odpowiedzi.



Dodatkowe informacje o RPC zawiera rozdział 7.

RPC udostępnia logiczną komunikację klient-serwer za pomocą protokołu transportowego, na przykład TCP lub UDP, który przenosi dane komunikatu pomiędzy komunikującymi się programami. RPC jest jednakże niezależny od protokołów transportowych; metoda przesyłania komunikatów pomiędzy procesami nie ma dla RPC znaczenia. Wobec tego RPC nie kojarzy ze zdalnymi procedurami zadnej określonej semantyki podczas implementacji logicznej łączności pomiędzy procesami. Semantyka jest wnioskowana z położonych ponizej protokołów transportowych. Ponadto RPC nie zapewnia żadnego mechanizmu wiarygodności, polegając zamiast tego na stosowanych protokołach transportowych. Oznacza to, że aplikacje muszą znać typ używanego przez RPC protokołu transportowego. Sama aplikacja musi zapewnić wiarygodność transmisji, jeśli stosowany jest protokół transportowy bez gwarancji dostawy, na przykład UDP.

Zewnętrzna reprezentacja danych

W implementacji NFS dostęp do zasobów plikowych odbywa się poprzez sieć. Reprezentacja danych w każdym komputerze zależy od modelu urządzenia i używanego systemu operacyjnego (jedno i drugie mogą być różne). Na przykład, kod ASCII (*American Standard Code for Information Interchange*) przedstawia znaki w kodzie 7-bitowym, zasada EBCDIC (*Extended Binary Coded Decimal Interchange Code*) używa kodu 8-bitowego. Znak „A” w ASCII reprezentowany jest przez kod 65, natomiast w EBCDIC przez kod 193. Te sposoby przedstawiania danych są zależne od komputera. Aby więc komunikacja odbywała się pomyslnie, musi istnieć standardowy sposób reprezentacji danych, niezależny od architektury komputera i od systemu operacyjnego. Standard XDR (*EXternal Data Implementation* — zewnętrzna reprezentacja danych) jest znormalizowanym sposobem prezentacji danych w sieci. XDR używa języka opisu danych do kodowania formatów danych, dzięki czemu dane można przesyłać przez sieć do dowolnego komputera, niezależnie od modelu urządzenia i systemu operacyjnego. W ten sposób standard XDR umożliwia komunikację pomiędzy różnymi systemami operacyjnymi i architekturami komputerów.

Uslugi NFS

Aby system NFS mógł funkcjonować, w serwerze NFS muszą być uruchomione określone usługi NFS, udostępniane przez następujące programy usługowe (demony):

- ◆ *nfsd* — obsługuje tworzenie, przeszukiwanie, odczyt i zapis plików. Za każdym razem, gdy klient zada komunikację, uruchamiany jest odseparowany demon *nfsd* służący do jej obsługi. Klient korzysta następnie z usług tego programu.
- ◆ *mountd* — kontroluje listę systemów plików udostępnianych w systemie lokalnym serwera i nasłuchiwa zadań montowania plików i katalogów ze stroną innych komputerów. Jeśli zadane pliki lub katalogi są dostępne, demon *mountd* pozwala klientom zamontować je. Ten program śledzi również zasoby montowane przez odległe systemy.

- ◆ *pcnfsd* — uzywany przez klienty do pobierania z serwera informacji zwiazanych z uwierzytelnianiem, na potrzeby dalszych transakcji. Klient inicjujacy laczosc wysyla nazwe uzytkownika i haslo, kontrolowane w serwerze.
- ◆ *portmapper* — udostepnia numery portów, z którymi klienty moga sie powiazac. Wobec tego kazdy klient przed wyslaniem pakietu do serwera musi poprosic program portmapper o identyfikacje portu, na który pakiety musza zostac wyslane.
- ◆ *statd* — monitoruje status serwera i klientów, na przyklad wystepujace restarty komputerów, aby przywróć funkcjonowanie po blokadach NFS.
- ◆ *lockd* — zarzadza blokowaniem systemów, aby uniknac równoczesnej modyfikacji danych przez kilka klientów. Demony statd i lockd współpracuja przy odzyskiwaniu zablokowanych systemów. W przypadku blokady systemu demon statd czeka na ponowna próbe dostepu systemu do zablokowanych plików. Jesli jednak zaden z zablokowanych systemów nie odpowiada, demon statd wysyla do administratora systemu komunikaty alarmów. Ponadto lockd przetwarza zadania blokad.

Zagadnienia bezpieczeństwa w NFS

NFS zapewnia latwy dostep do plików w sieci. Jednakze kilka cech zabezpieczeń powoduje, iz system NFS moze nie byc bezpieczny:

- ◆ W NFS tylko komputery, których nazwy lub adresy IP wymienione sa w specjalnym pliku w serwerze maja prawo montowac zasoby. Jednakze osoba z zewnatrz moze przejac zaufany adres i uzyskac dostep do zamontowanych zasobów.
- ◆ Po zakonczeniu procedury montowania system plików kontroluje dostep uzytkownika do plików. Zalozmy sytuacje, w której uzytkownik posiadajacy określony identyfikator jako jedyny ma pelne prawa do pliku, do którego inni uzytkownicy maja jedynie prawo do odczytu. Jesli ktos inny przypisze sobie ten identyfikator, uzyska pelne prawa do pliku.

Mozemy rozwiazac problemy zwiazane z bezpieczeństwem NFS, instalujac najnowsze dostepne laty zabezpieczajace NFS. Ponadto systemy plików nalezy eksportowac z uprawnieniami tylko do odczytu. Jesli nie jest to jednak mozliwe, powinnismy eksportowac systemy plików tylko do ograniczonej liczby hostów.



Dodatkowe informacje o bezpieczeństwie NFS mozna znalezc w witrynie Webopaedia (www.pcwebopaedia.com) oraz witrynie WWW firmy Sun Microsystems (www.sun.com).

Wersje NFS

NFS pozwala uzytkownikom zarzadzac plikami w kilku komputerach w sieci tak, jakby miescily sie na lokalnym dysku twardym. Usluga NFS zostala opracowana przez Sun Microsystems w polowie lat 80., jednakze pierwsza wersja NFS nie zostala nigdy opublikowana. Poprawiona, druga wersja zostala opublikowana i zaimplementowana w systemie operacyjnym SunOS 2.0. Od tego czasu firma Sun Microsystems wlacza NFS do systemów SunOS. Poniewaz NFS jest niezalezny od architektury komputera, systemu operacyjnego, sieci i protokolu transportowego, został zaimplementowany na różnych

platformach — Unix, MS Windows, NetWare i OS/2. Jednakże NFS w wersji 2. ma kilka ograniczeń, a od jej premiery zaproponowano przynajmniej osiem nowych wersji NFS, aby uporać się z tymi ograniczeniami. Nie wszystkie zaproponowane wersje zostały zaimplementowane.

NFS w wersji 3. nadal jest bezstanowy. Podobnie jak w wersji 2., każde zadanie jest kompletne, wystarczające i jego przetworzenie nie zależy od innych zadań. Serwer nie musi utrzymywać żadnych informacji o stanie, zas przywrócenie stanu sprzed awarii systemu jest proste — wystarczy, że klient będzie ponawiać zadania, dopóki nie otrzyma odpowiedzi od serwera. Wersja 2. do transportu używa protokołu UDP, natomiast wersja 3. TCP (TCP jest bardziej niezawodny, lecz daje gorszą wydajność). Do różnic pomiędzy wersjami 2. i 3. należały:

- ◆ *Rozmiar identyfikatora pliku* — identyfikator pliku (*file handle*) jest wskaznikiem do pliku udostępnionego w serwerze NFS. Wersja 2. używa identyfikatorów o stałej długości (32 bajty). W wersji 3. rozmiar identyfikatora pliku może być zmieniający i został zwiększyony do maksimum 64 bajtów.
- ◆ *Maksymalne rozmiary danych* — wersja 2. nakładała na procedury zapisu i odczytu (READ i WRITE) ograniczenie do 8 kB, co pogarszało wydajność klientów. Procedura READ używana jest we wszystkich zadaniach odczytu ze strony klientów. Wersja 3. nie nakłada ograniczeń rozmiarów danych.
- ◆ *Niezawodny asynchroniczny zapis pliku* — w wersji 2. serwer zapisuje dane na trwałym nośniku — na przykład dysk twardy — synchronicznie, a następnie potwierdza zadanie klienta WRITE. Procedura taka pogarszała wydajność systemu. W wersji 3. serwer wysyła do klienta potwierdzenie zapisu natychmiast po otrzymaniu zadania zapisu asynchronicznego. Następnie, aby sprawdzić, czy dane zostały zapisane na nośniku trwałym, klient wysyła do serwera zadanie COMMIT. Serwer odpowiada na to zadanie dopiero po bezpiecznym zapisaniu danych.
- ◆ *Zgodność pamięci podręcznej* — w wersji 2. klienci przechowywali w pamięci podręcznej pliki i katalogi, aby poprawić wydajność. Aby sprawdzić poprawność danych w pamięci podręcznej, klienci porównywały datę i czas modyfikacji pliku lub katalogu w serwerze i w pamięci podręcznej. Jeśli wartości te są identyczne, to klient zakłada, że dane w pamięci podręcznej są aktualne. Jednakże gdy sam klient modyfikuje dane w pamięci podręcznej, dane czasowe nie mogą zostać użyte do kontroli ważności danych. Jak widać, metoda ta jest nieskuteczna. W wersji 3. zgodność pamięci podręcznej jest utrzymywana za pomocą dwóch wersji atrybutów pliku w serwerze — przed operacją i po operacji. Jeśli czas modyfikacji dwóch operacji zgadza się, pamięć podręczna zawiera poprawne dane.

Wprawdzie NTFS w wersji 3. rozwiązuje większość problemów z wersji 2., lecz w połowie roku 2000 wydana została nowa, czwarta wersja NTFS, która zapewnia poprawione:

- ◆ zgodność danych i pamięci podręcznej,
- ◆ opcje bezpieczeństwa,
- ◆ współpracywność pomiędzy różnymi platformami.

Konfiguracja serwera NFS

Aby zaimplementowac protokol NFS w celu przezroczystego dostepu do plikow udostepnianych w sieci, musimy skonfigurowac serwer NFS. Mozna to zrobic w roznych systemach operacyjnych, zarówno Windows, jak i uniksowych. Biezacy punkt opisuje konfiguracje serwera NFS pod Linuksem i Windows 2000.

Konfiguracja serwera NFS w systemie Linux

Konfiguracja serwera NFS w systemie Linux obejmuje edycje kilku plikow konfiguracyjnych. W pierwszej kolejnosci musimy sie jednakze upewnic, czy protokol TCP/IP jest skonfigurowany i funkcjonuje poprawnie, oraz czy zainstalowany jest odpowiedni pakiet nfs-utils. Pakiet ten sklada sie z kolekcji programow uslugowych (demonow) wymaganych do funkcjonowania NFS i miesci sie w katalogu */sbin* lub */usr/sbin*. Musimy dokonac edycji trzech plikow konfiguracyjnych: */etc(exports*, */etc/hosts.allow*, oraz */etc/hosts.deny*.

Plik */etc(exports* zawiera wpisy dla wszystkich katalogow udostepnionych w sieci. Wpisy te zawieraja szczegolowe informacje o dostepie do katalogow. Na przyklad, czesc katalogow pozwala jedynie na odczyt, zas inne na odczyt i modyfikacje. Do tego pliku musimy dodac wpisy dla wszystkich udostepnionych katalogow. Typowy wpis wyglada nastepujaco:

```
katalog komputer1(opcja11,opcja12) komputer2(opcja21,opcja22)
```

W tej skladni:

- ◆ *katalog* oznacza udostepniony katalog. Wszystkie jego podkatalogi zostaja udostepnione automatycznie.
- ◆ *komputer1 i komputer2* oznaczaja komputery majace prawo dostepu do katalogu. Dopuszczalne jest okreslenie komputera zarówno za pomoca adresu IP, jak i nazwy DNS, jednakze bezpieczniejsza jest uzywanie adresow IP.
- ◆ *opcja* okresla typ dostepu, jaki komputer otrzyma do udostepnionego katalogu. Do opcji naleza:
 - ◆ *ro* — dostep tylko do odczytu; jest to opcja domyslna.
 - ◆ *rw* — prawo do odczytu i zapisu w udostepnionym katalogu.
 - ◆ *no_root_squash* — wskazuje, ze uzytkownik root z komputera klienta bedzie mial taki sam poziom dostepu do plikow, jak uzytkownik root z serwera. Opcja ta zwykle nie jest uzywana z uwagi na bezpieczenstwo, lecz moze byc potrzebna, jesli bedziemy musieli wykonac jakies zadania administracyjne w kliencie.
 - ◆ *no_subtree_check* — oznacza, ze za kazdym razem, gdy klient zada pliku, serwer weryfikuje (za pomoca procedury zwanej sprawdzaniem poddrzewa), czy plik mieści sie w odpowiedniej czesci partycji, jesli tylko czesc tej partycji zostala wyeksportowana. Jesli jednak wyeksportowana zostala cala partycja, mozemy nie uzywac tej opcji, aby przyspieszyc transfer.

Wezmy pod uwagę sytuację, gdy trzeba udostępnic przez sieć komputerom *komputer1* i *komputer2* dwa katalogi w serwerze: */usr/sales* i */usr/reports*. Adresy wymienionych komputerów to odpowiednio 192.17.0.1 i 192.17.0.2. Jeśli oba komputery potrzebują jedynie dostępu z prawami do odczytu, to plik */etc/exports* będzie zawierał następujące wpisy:

```
/usr/sales 192.17.0.1(ro) 192.17.0.2(ro)  
/usr/reports 192.17.0.1(ro) 192.17.0.2(ro)
```

Gdy liczba komputerów, które wymagają dostępu do komputera jest duża, możemy stosować zakresy adresów zamiast poszczególnych nazw komputerów. W tym przypadku, aby wyszczególnić komputery w podsieci, możemy podać adres sieci i maskę podsieci.

Pliki */etc/hosts.allow* i */etc/hosts.deny* zawierają wpisy dla wszystkich usług w serwerze, z których mogą korzystać inne komputery w sieci. Wpisy w tym pliku wymieniają usługi i zbiory komputerów w sposób następujący:

```
usługa: komputer1, komputer2
```

W tych wpisach możemy stosować zarówno nazwy komputerów, jak i adresy IP.

Za każdym razem, gdy klient wysyła zadanie do serwera, ten wykonuje następujące czynności:

- ◆ Serwer sprawdza, czy komputer, który wysłał zadanie, posiada w pliku *hosts.allow* wpis dla zadanej usługi. Jeśli tak, serwer zezwala na dostęp.
- ◆ Jeśli komputer nie posiada wpisu dla zadanej usługi w pliku *hosts.allow*, to serwer szuka tej samej informacji w pliku *hosts.deny*. Gdy zostanie znaleziony wpis tej usługi dla danego komputera, klient nie otrzymuje dostępu.
- ◆ Jeśli serwer nie znajdzie wpisu komputera dla danej usługi w żadnym z tych plików, to zezwala na dostęp.

Po zmodyfikowaniu tych trzech plików konfiguracyjnych wystarczy zrestartować komputer, aby uruchomić NFS. W trakcie restartu serwera skrypty uruchomieniowe automatycznie wykrywają ustawienia */etc/exports* i uruchamiają NFS. By zweryfikować, czy usługa NFS jest uruchomiona, możemy użyć polecenia *rpcinfo -p*. Czasami niezbędne jest dodanie wpisów w pliku inicjalizacyjnym, aby uruchomić usługi lub demony NFS. Na przykład, aby uruchomić demony *mountd* i *nfsd*, należy dodać do skryptów uruchomieniowych następujące wiersze:

```
rpc.mountd  
rpc.nfsd
```

Konfiguracja serwera NFS w Windows NT Server

Podczas instalacji systemu Windows NT Server 4.0 oprogramowanie serwera NTFS nie zostaje automatycznie zainstalowane. W tym celu potrzebny jest dodatkowy pakiet Microsoft Windows NT Services for Unix (SFU). Pakiet SFU pozwala wzajemnie udostępniać zasoby sieciowe systemom Windows NT, Windows 2000 i odmianom Uniksa. Po zainstalowaniu SFU możemy skonfigurować serwer NFS. Zadanie to jest bardzo podobne w systemach Windows 2000 Server i Windows NT Server. Aby skonfigurować NFS w systemie Windows NT Server, trzeba utworzyć folder przeznaczony do udo-

stepnienia, dodac grupy klientów NFS, udostepnic katalog, z którego klienci maja korzystac, oraz przydzielic uprawnienia uzytkownikom. Pomoze w tym Czytelnikowi poniwsza procedura:

1. Kliknij prawym przyciskiem myszy folder przeznaczony do udostepnienia i wybierz *Wlasciwosci* z menu podrecznego.
2. Wybierz zakladke *Zabezpieczenia*, kliknij *Uprawnienia* i zaznacz pole wyboru *Zastap uprawnienia dla podkatalogów*.
3. Zaznacz pole wyboru *Zastap uprawnienia dla istniejacych plikow*.
4. Pod polem *Nazwa* wybieraj pojedynczo po kolei uprawnienia i za kazdym razem kliknij *Usun*, aby je skasowac.
5. Kliknij *Dodaj*, a nastepnie *Pokaz uzytkownikow*.
6. Wybierz *Wszyscy*, grupe administratorow i uzytkownika Administrator. Kliknij *Dodaj*.
7. Zmien typ dostepu na *Pelna kontrola* i zamknij okno dialogowe.
8. Wybierz *Start/Programy/Windows NT Services for Unix/Server for NFS*.
9. Wybierz zakladke *NFS Client Groups*. Nacisnij *Alt+G*, wpisz nazwe grupy i kliknij *OK*, aby dodac grupy klientow. Nacisnij *Alt+M*, wpisz nazwe hosta lub adres IP klienta NFS, kliknij *OK*, a nastepnie *Apply* (Dodaj), aby dodac czlonka do grupy klientow.



Grupa klientów (*client group*) oznacza grupę komputerów mających dostęp do folderów udostepnionych przez NFS.

10. Wybierz zakladke *Share Options* (Opcje udostepniania). Wprowadz pełna ścieżkę udostepnianego folderu i zmień dostęp na *No Access* (brak dostępu). Nacisnij *Alt+A*, wybierz utworzoną przed chwilą grupę klientów i kliknij *Add*. Przydziel uprawnienia administracyjne jako *Root* albo *Anonymous*. Aby przyznać uprawnienia administracyjne jako *Root*, nacisnij *Alt+A* i wybierz typ dostępu *Root*. Aby przyznać je jako *Anonymous*, nacisnij *Alt+A* i wybierz typ dostępu *Read-Write* (zapis i odczyt). Nacisnij *Alt+Y* i kliknij *Apply*, aby zastosować ustawienia opcji udostepniania.
11. Skonfiguruj odwzorowania użytkowników i grup systemu Windows na użytkowników i grupy systemu Unix. Nacisnij *Alt+O*, a następnie *Alt+E*, aby zmodyfikować plik hasel i wprowadzić nową nazwę i identyfikator użytkownika oraz identyfikator grupy. Nacisnij *Alt+A*, a następnie *Alt+O*. Wybierz użytkowników w *NFS Users* i *Windows Users*, a następnie nacisnij *Alt+D*, aby dodać odwzorowanie. Nacisnij *Alt+A*, aby zastosować zmiany.

Po skonfigurowaniu serwera NFS, uniksowe klienci NFS będą mogli korzystać z udostepnionych plików przez proste zamontowanie udziału serwera NFS.

DFS

Systemy plików dostępne w różnych systemach operacyjnych — na przykład FAT lub NTFS — jedynie zarządzają organizacją danych na lokalnym nośniku fizycznym. Użytkownicy jednakże muszą często znajdować informacje w sieci, zwykle rozproszone po różnych serwerach. Wobec tego użytkownicy mają kłopoty z wyszukiwaniem informacji, gdy przeglądają udostępnione foldery różnych komputerów w sieci. Administratorzy sieciowi mogą zaradzić tym problemom, stosując rozproszony system plików DFS (*Distributed File System*). DFS oszczędza czasu i zyskuje na wyszukiwaniu pliku w całej sieci.

Wprowadzenie do DFS

W wyciągu systemów operacyjnych do pracy z Internetem Microsoft udostępnił DFS, który wyewoluował z NetBIOS-u — składnika pakietu protokołów TCP/IP. DFS (*Distributed File System* — rozproszony system plików) pozwala użytkownikom korzystać z udostępnionych plików i katalogów, fizycznie rozrzuconych po sieci, bez konieczności podawania ich położenia fizycznego. DFS można skonfigurować w dowolnym serwerze sieciowym Windows 2000. Usługa ta gromadzi razem wszystkie udostępnione pliki i katalogi, fizycznie rozrzucone po sieci, i udostępnia je wszystkim użytkownikom tak, jakby mieściły się w pojedynczym serwerze, w którym skonfigurowany został DFS. Na przykład, jeśli dane sprzedawane są po wielu serwerach w sieci, możemy skorzystać z systemu DFS, aby uzyskać taki dostęp do danych, jakby wszystkie były położone w pojedynczym serwerze.



W strukturze serwerów sieciowych Windows 2000 serwer DFS można również umieścić w systemie Windows NT Server 4.0.

DFS składa się z katalogu głównego (korzenia) DFS, jednego lub wielu linków DFS i jednego lub wielu udostępnionych folderów DFS, na które wskazują poszczególne linki. Wszystkie te składniki tworzą razem topologię DFS. Serwer, w którym znajduje się katalog główny DFS, nosi nazwę serwera macierzystego. Katalog główny DFS oznacza udostępniony katalog w tym serwerze, który gra rolę punktu wyjściowego i hosta dla innych udziałów. W jego obrębie możemy za pomocą linków DFS, wskazujących fizyczną ścieżkę do udostępnionego folderu w sieci, tworzyć udostępnione foldery DFS. W ten sposób DFS udostępnia ścieżki logiczne do wszystkich udostępnionych plików i katalogów (folderów) w sieci. Użytkownicy uzyskują dostęp do plików i katalogów udostępnionych w sieci, po prostu montując DFS. Ponieważ użytkownicy nie muszą znać nazw serwerów i udziałów, mogą korzystać z plików i katalogów spod stałych udostępnionych folderów DFS, nawet gdy ich fizyczne położenie ulegnie zmianie. Serwer DFS daje szereg różnych korzyści:

- ◆ Pojedyncza struktura hierarchiczna służy do przeglądania wszystkich udostępnionych folderów w sieci i zachowuje się jak pojedynczy dysk twardy o dużej pojemności. Taka struktura ułatwia użytkownikom dostęp do zasobów sieciowych.

- ◆ Uzytkownicy nie musza znac fizycznego polozenia plików i folderów. Administratorzy moga przenosic udostepnione foldery i nie ma to wpływu na dostep uzytkowników do danych. Dzieki temu DFS zapewnia elastyczne zarzadzanie danymi.
- ◆ Pojedynczy katalog główny DFS moze laczyc wiele udostepnionych folderów DFS, rozrzuconych fizycznie po sieci. Dzieki temu, jesli plik na jednym serwerze jest intensywnie uzywany, uzytkownicy nie musza korzystac z plików na tym samym serwerze, co zmniejsza jego obciążenie — dostep uzytkowników do plików jest rozlozony na wiele serwerów. Jednak ze punktu widzenia uzytkowników wyglada, jakby plik znajdował sie tylko w jednym miejscu sieci.
- ◆ DFS potrafi współpracowac z innymi systemami sieciowymi, na przyklad Microsoft Windows 95, Windows 98 i Windows NT 4.0.
- ◆ DFS pozwala klientom buforowac informacje o udostepnionych folderach w serwerze macierzystym. Lokalna pamiec podreczna klientów minimalizuje ruch w sieci i czas odpowiedzi dla uzytkowników.
- ◆ DFS zapewnia integracje zabezpieczeń bez dodatkowych nakladów pracy na ich implementacje. Uzytkownik, który laczy sie z katalogiem głównym DFS, ma dostep tylko do tych plików, do których posiada odpowiednie uprawnienia.

Katalogi główne DFS: autonomiczny i domeny

DFS mozemy zaimplementowac tworzac autonomiczny katalog główny DFS lub katalog główny DFS domeny. W przypadku autonomicznego DFS domena zawiera tylko jeden serwer macierzysty — katalog główny DFS mieści sie tylko w jednym serwerze DFS. Natomiast w przypadku DFS domeny moze istniec wiele serwerów macierzystych. Poniewaz autonomiczna implementacja DFS stosuje tylko jeden serwer macierzysty, jesli z jakiegos powodu bedzie on niedostepny (na przyklad, podczas konserwacji systemu), uzytkownicy nie beda mogli korzystac ze swoich plików. Poniewaz implementacja DFS w domenie moze obejmowac wiele serwerów macierzystych, zapewnia ona uzytkownikom wysoka dostepnosć plików; jest ona również wynikiem dwóch faktów:

- ◆ Windows 2000 Server automatycznie publikuje topologie DFS w Active Directory. Active Directory jest usluga katalogowa, zawarta w systemie Windows 2000 Server, która sklada się informacje o obiektach sieciowych i udostepnia je uzytkownikom i administratorom sieci. Dzieki temu DFS domeny zapewnia widoczność topologii DFS dla wszystkich uzytkowników we wszystkich serwerach w domenie.
- ◆ Implementacja DFS w domenie pozwala na automatyczna replikacje katalogu głównego DFS i udostepnionych folderów DFS do wiekszej liczby serwerów. Replikacja oznacza kopianie katalogu głównego i folderów DFS do serwerów domeny. Dzieki temu, gdy jeden serwer jest niedostepny, uzytkownicy nadal mogą korzystac ze swoich plików.

Konfiguracja DFS w Windows 2000

DFS mozna skonfigurowac zarówno na partycji FAT, jak i NTFS. Poniewaz jednak NTFS zapewnia wiele funkcji zabezpieczeń od systemu plików FAT, zaleca sie konfigurowac DFS na partycji NTFS. Konfigurowanie DFS obejmuje utworzenie korzenia, laczy i udostepnionych folderów DFS.

Tworzenie katalogu głównego DFS

Aby utworzyć korzeń DFS:

1. Wybierz *Start/Programy/Narzędzia administracyjne/Rozproszony system plików.*
2. Wybierz *Nowy katalog główny rozproszonego systemu plików DFS* z menu *Akcja*, aby uruchomic kreatora.
3. Kliknij *Dalej*, aby otworzyć okno typu katalogu głównego plików DFS.
Domyslnie wybrana jest opcja *Utwórz katalog główny systemu plików DFS domeny*. Aby założyć autonomiczny system plików DFS, trzeba wybrać *Utwórz autonomiczny katalog główny systemu plików DFS*. Kliknij *Dalej*, aby przejść do następnego kroku kreatora.
4. Jeśli wybrany został katalog główny DFS oparty na domenie, to kreator zaproś o nazwę domeny, w której ma zostać utworzony katalog główny DFS. Wpisz nazwę domeny i kliknij *Dalej*.
5. Podaj nazwę serwera macierzystego dla katalogu głównego DFS i kliknij *Dalej*, aby otworzyć okno *Podaj udział dla katalogu głównego systemu plików DFS*.
 Zamiast wpisywać ręcznie nazwę serwera, można kliknąć *Przeglądaj* i wybrać nazwę serwera z listy.
6. W oknie tym można wpisać udostępniony folder. Podaj ścieżkę do istniejącego udostępnionego folderu lub nazwę nowego, do utworzenia. Kliknij *Dalej*.
7. Wyświetlona zostanie nazwa domyślna dla katalogu głównego DFS, zamiast której możesz podać nową. Kliknij *Dalej*.
8. Kliknij *Zakoncz*. Nowy katalog główny DFS został utworzony.

Po założeniu katalogu głównego DFS należy ponownie uruchomić serwer, aby nowa implementacja DFS została aktywowana.

Tworzenie lacza DFS lub udostępnionych folderów

Lacze DFS może zawierać wiele udostępnionych folderów. Jednakże podczas tworzenia lacza DFS automatycznie dodawany jest pierwszy udostępniony folder. Aby utworzyć lacze DFS:

1. Wybierz *Start/Programy/Narzędzia administracyjne/Rozproszony system plików.*
2. Kliknij prawym przyciskiem myszy katalog główny DFS, do którego chcesz przydzielić udostępniony folder, a następnie wybierz *Nowe lacze DFS* z menu podręcznego.
3. Podaj nazwę foldera w polu *Nazwa lacza*.
4. Utwórz lacze do udostępnionego foldera, wpisując ścieżkę do niego w polu *Wyslij użytkownika do tego udostępnionego foldera*.



Nazwa foldera wpisana do pola Nazwa lacza bedzie widoczna dla wszystkich uzytkownikow sieci.

Narzedzia do przesyłania plików

Bywa, ze chcemy udostepnic pliki innemu komputerowi lub przeslac pliki z jednego komputera do innego. Proces przenoszenia pliku z jednego komputera do drugiego to transfer plików. Mozemy przesyłac pliki przez Siec za pomoca odpowiednich narzedzi. Jednym z podstawowych zakresów wykorzystania protokolów TCP/IP jest transfer plików. TCP/IP udostepnia protokoly, na przyklad FTP (*File Transfer Protocol*), TFTP (*Trivial File Transfer Protocol*) oraz rcp (*Remote Copy*), ktore pomagaja przesyłac pliki przez Siec i zarzadzac nimi. W niniejszym podrozdziale omówimy szczegółowo te narzedzia.

FTP

Jednym z najpowszechniej spotykanych zastosowan TCP/IP jest transfer plików.



Transfer plików oznacza kopiowanie ich z jednego komputera do drugiego, bez naruszania kopii zródłowej.

Protokół transferu plików FTP jest jednym z protokolów pakietu TCP/IP. FTP jest zbiorem reguł, które zajmuja sie przesyaniem plików z jednego komputera do drugiego. Dodatkowo, zapewnia on przesyłanie danych w sposób wiarygodny i wydajny. FTP zawiera szereg różnych polecen, które pomagaja przesyłac pliki, tworzyc katalogi i zarzadzac nimi. FTP różni sie od innych protokolów z pakietu TCP/IP tym, ze uzywa dwóch portów TCP/IP — 20. i 21. Porty te nosza odpowiednio nazwy procesu transferu danych (DTP — *Data Transfer Process*) oraz interpretera protokolu (PI — *Protocol Interpreter*). Port 20. sluzy do przesyłania informacji o katalogach i plikach, natomiast 21. do przesyłania polecen.

FTP opiera sie na architekturze klient-serwer. Klient laczy sie z serwerem na porcie 21., podczas gdy serwer uzywa portu 20., aby polaczyc sie z powrotem z klientem i przesyłac dane. Klient rozpoczyna sesje, wysylajac na port 21. zadanie polaczenia z serwerem FTP. Jest to tzw. polaczenie kanalu sterujacego (*Control Channel*). Klient wysyla polecenie PORT, zawierajace numer portu, z którym serwer musi sie polaczyc w celu wymiany danych pomiedzy dwoma portami. Nastepnie, serwer FTP przesyła dane ze swojego portu 20. na port wyszczególniony przez klienta w poleceniu PORT. Poniewaz ten transfer danych inicjowany jest przez serwer i nie jest kontrolowany przez klienta, zapora firewall po stronie klienta nie jest w stanie wykryc zródła danych, co moze prowadzic do problemów z bezpieczenstwem — poniewaz niechciane dane moga udawac transfer danych FTP i dotrzec do klienta. Problem ten mozna rozwiązacz za pomoca pasywnego FTP, w którym klient zamiast polecenia PORT wysyla polecenie PASV. Polecenie to zada od serwera numer portu klienta, który bedzie uzywany do przesyłania danych. Serwer wysyla numer portu, który nastepnie zostaje uzyty przez klienta do inicjacji

wymiany danych. Ponieważ w tym przypadku serwer odpowiada na zadanie inicjowane przez klienta, zapora jest w stanie wykryć źródło, z którego nadchodzi dane. Większość klientów stosuje pasywny FTP.

Przy stosowaniu FTP do transferu plików program FTP w komputerze użytkownika (hoscie lokalnym) komunikuje się z programem FTP w odległym komputerze. Jednakże te dwa hosty niekoniecznie muszą posiadać taki sam system operacyjny. Aby przesłać pliki, wymieniany jest ciąg poleceń pomiędzy hostami. Po przesłaniu plików połączenie może zostać przerwane przez komputer lokalny.

Dostępnych jest wiele różnych programów FTP, zarówno z tekstowymi, jak i graficznymi interfejsami użytkownika. Interfejsy znakowe i programy uruchamiane z wiersza poleceń dostępne są dla tekstowych systemów operacyjnych — na przykład, DOS-u i Unixa, natomiast programy z graficznym interfejsem użytkownika są dostępne dla systemów operacyjnych typu Windows. W przypadku programów uruchamianych z wiersza poleceń użytkownik musi wpisywać polecenia, aby przesyłać pliki lub nimi zarządzać. W programach FTP z interfejsem graficznym możemy wykorzystać pomocą przycisków i ikon.

FTP przesyła pliki w dwóch formatach: binarnym i ASCII. Format binarny służy do przesyłania plików z danymi binarnymi i plików wykonywalnych. Format ASCII służy do przesyłania plików tekstowych. Domyslnie pliki przesyłane są w formacie ASCII, ponieważ systemy Unix i Windows stosują w plikach tekstowych odmienne zakończenia wierszy. Systemy uniksowe stosują do tego celu znak przesuwu o wiersz (*line feed*), zaś Windows kończy wiersze znakami powrotu karetki (*carriage return*) i przesuwu o wiersz. Stosowanie formatu ASCII zapewnia poprawne tłumaczenie zakończeń wierszy przy przesyłaniu plików pomiędzy różnymi środowiskami hostów.



Należy zawsze pamiętać, by podać właściwy format podczas przesyłania plików. Jeśli przesyłamy plik binarny w formacie ASCII, to wersja, która dotrze do komputera docelowego będzie uszkodzona. Plik ASCII przesłany jako binarny zostanie skopiowany bez żadnych zmian.

Jak więc działa FTP? W prawdziwe poszczególne programy FTP funkcjonują nieco odmiennie, lecz podstawowy mechanizm pozostaje niezmieniony. Podstawowa procedura działania programu FTP wygląda następująco:

1. Komputer macierzysty nawiązuje połączenie z komputerem zdalnym.
2. Użytkownik komputera macierzystego loguje się do zdalnego hosta.
3. Użytkownik tworzy lub znajduje katalog w komputerze zdalnym. Katalog ten zawiera potrzebny plik lub przeznaczony jest do odbioru pliku.
4. Użytkownik używa odpowiedniego polecenia lub menu i przycisków (zależnie od typu używanego programu), aby przesłać pliki.
5. Po przesłaniu wszystkich plików użytkownik wychodzi z sesji ftp przez wylogowanie ze zdalnego komputera.

Programy znakowe FTP zazwyczaj wymagają przy uruchomieniu podania nazwy użytkownika i hasła. Graficzne programy FTP zwykle wyświetlają okno, w którym możemy

wybrac z listy nazwe lub adres IP zdalnego systemu. Po nawiazaniu polaczenia uzytkownik musi podac nazwe uzytkownika i haslo, by zalogowac sie do zdalnego komputera.

W Internecie FTP moze posluzyc do pobierania darmowego oprogramowania i plików z publicznych bibliotek mieszczacych sie na *anonimowych serwerach FTP*. Osrodkie takie pozwalaja laczac sie z serwerem w sposob anonimowy, stad nazwa uzytkownika Anonymous. Na zapytanie o haslo, mozemy wprowadzic swoja tozsamosc sieciowa. Witryny tego typu daja anonimowym uzytkownikom jedynie prawo do odczytu, aby nie mogli oni wprowadzac zmian w archiwach.

TFTP

TFTP (*Trivial File Transfer Protocol* — prosty protokol przesyłania plików) jest protokołem TCP/IP sluzacym do transferu plików, tekstowych i binarnych, z jednego komputera do drugiego. TFTP uzywa do przesyłania plików portu 69. UDP i jest z założenia prosty i łatwy w użyciu. Wobec tego, w przeciwienstwie do FTP, TFTP nie pozwala wykonywac operacji na plikach i katalogach — na przykład, listowac zawartosci katalogu lub zarządzac plikami, a poza tym nie zapewnia uwierzytelnienia uzytkownika. TFTP jest implementowany na podstawie protokołu UDP i sluzy przede wszystkim do uruchamiania ruterów i bezdyskowych stacji roboczych. Poniewaz TFTP nie uwierzytelnia uzytkownika, komputer macierzysty nie musi podawac nazwy uzytkownika ani hasla. Z tego powodu protokol TFTP jest uzywany jedynie z zaufanymi klientami.

TFTP dziala w sposob nastepujacy:

1. Komputer macierzysty wysyla zadanie transferu plików do komputera zdalnego.
2. Po przyjeciu zadania przez komputer zdalny, zamówiony plik zostaje wyslany w postaci pakietów o stalej dlugosci 512 bajtów. Kazdy wyslany pakiet jest numerowany.
3. Komputer odbierajacy potwierdza odbiór kazdego pakietu, odsylajac potwierdzenie z numerem bloku otrzymanego pakietu.
4. Po wyslaniu pakietu o dlugosci mniejszej od 512 bajtów transfer pakietów pomiędzy odbiorca i nadawca zostaje zakonczony.

Jesli pakiet (lub potwierdzenie) zostanie utracony podczas transmisi, to nalezy wyslac go ponownie. Gdy do przesyłania plików przez siec sluzy TFTP, oba komputery pełnia funkcje nadawców i odbiorców. Komputer macierzysty odbiera dane i wysyla potwierdzenia, zas komputer zdalny wysyla pakiety i odbiera potwierdzenia.

Remote Copy Protocol

Protokol *rcp* (*Remote Copy Protocol* — protokol zdalnego kopiowania) jest protokołem TCP/IP, nalezacym do kategorii tzw. *r-narzedzi (r-utility)* i sluzy do przesyłania plików do i ze zdalnego komputera. Protokol ten został zaimplementowany w postaci polecenia *rcp*. Przed użyciem polecenia *rcp* trzeba w komputerze zdalnym utworzyc plik *.rhosts*, zawierajacy nazwy systemów, którym ten zdalny komputer może zaufać. Jesli jednak do dostępu do zdalnego komputera uzywane są różne nazwy uzytkowników, w pliku *.rhosts* musimy podać te nazwy po nazwach systemów. Stosując protokol *rcp*, uzytkow-

nik musi podać nazwę systemu, która jest następnie porównywana z zapisaną w pliku *.rhosts*. Pliki można kopiować tylko wtedy, gdy nazwy: podana jako parametr polecenia *rcp* i zawarta w pliku *.rhosts* pasują do siebie. Jeśli nazwy użytkowników w komputerach lokalnym i zdalnym są różne, musimy w poleceniu *rcp* podać nazwę użytkownika przed nazwą systemu. Gdy stosuje się polecenie *rcp*, nie trzeba podawać hasła. Weźmy pod uwagę dwa komputery, zdalny *komputer1* i lokalny *komputer2*. Aby komputer 2 mógł uzyskać dostęp do plików w komputerze 1, w pliku *.rhosts* komputera 1 należy umieścić następujący wiersz:

```
komputer2.<nazwa domeny>
```

Polecenie skopiowania przez *rcp* pliku o nazwie *plik1* z komputera 1 do komputera 2 wyglądać będzie następująco:

```
komputer1$ rcp komputer2: plik1 plik1
```

Jeśli nazwa użytkownika w komputerze 1 brzmi *użytkownik1*, zaś w drugim *użytkownik2*, to w pliku *.rhosts* musimy podać nazwę użytkownika po nazwie systemu:

```
komputer2.<nazwa domeny> użytkownik2
```

W takim przypadku polecenie skopiowania przez *rcp* pliku o nazwie *plik1* z komputera 1 do komputera 2 wyglądać będzie następująco:

```
komputer1$ rcp użytkownik2@komputer2: plik1 plik1
```

Polecenie *rcp* może przyjmować różne opcje, opisane w tabeli 12.1:

Tabela 12.1. Opcje polecenia *rcp*

Opcja	Znaczenie
-a	Wybiera tryb przesyłu ASCII, domyślny dla <i>rcp</i> .
-b	Wybiera binarny tryb przesyłu, którego trzeba używać do przesyłania plików binarnych i wykonywalnych.
-h	Sluży do przesyłania razem z innymi plikami plików ukrytych.
-r	Sluży do kopiowania zawartości podkatalogów.
Host	Sluży do określenia hosta lokalnego lub zdalnego.
Użytkownik	Sluży do określenia nazwy użytkownika. Opcji tej należy używać, gdy zdalny użytkownik jest inny od bieżącego lokalnego.
Zródło	Sluży do wyszczególnienia plików, które trzeba skopiować.
Sciezka\przeznaczenie	Sluży do określenia wzglednej sciezki do katalogu logowania w zdalnym komputerze.

Rozdział 13.

Narzędzia zdalnego wykonywania poleceń

W tym rozdziale:

- ◆ Przegląd narzędzi zdalnego wykonywania poleceń
- ◆ Opis serwerów terminali

Zyjemy w czasach, gdy przedsiębiorstwa nie uznaja granic krajów i kontynentów: sieci firm mogą mieć zasięg globalny. W ciągu ostatniej dekady ogromnie zwiększyły się szeregi pracowników mobilnych, co oznacza konieczność opracowania dla nich nowych metod dostępu do zasobów poza ich fizycznym zasięgiem. W jaki sposób, na przykład, podróżujący biznesmen może korzystać z komputera mieszkającego się w jego domowym biurze? Jak administrator może zdalnie rozwiązywać problemy ze stacjami roboczymi?

Odpowiedź na powyższe pytania brzmi: za pomocą *dostępu zdalnego* — mechanizmu dostępu do zasobów i usług fizycznie od nas oddalonych. Dostępnych jest wiele programów użytkowych, które pomagają korzystać z zasobów zdalnych; narzędzia te udostępniają interaktywne połączenia z komputerami zdalnymi i możliwość wydawania w nich interaktywnych poleceń. Podczas pracy z systemem zdalnym lokalny system staje się przezroczysty. Polecenia w nim wydawane są bezpośrednio przesyłane do odległego komputera, a odpowiedzi tego odległego komputera są wyświetlane na monitorze użytkownika.

W niniejszym rozdziale Czytelnik zapozna się z popularnymi narzędziami zdalnego wykonywania poleceń, takimi jak Telnet, Remote login (rlogin), Remote shell (rsh), Secure shell (ssh) i Remote execute (rexec), które pozwalają na dostęp do zdalnych zasobów i usług. Pokażemy też, jak serwery terminali, takie jak Sun Ray, Serwer usług terminalowych Microsoftu i Citrix, pomagają w dostępie do zdalnych usług i zasobów.

Przegląd narzędzi zdalnego wykonywania poleceń

Większość z tych narzędzi została opracowana w University of California w Berkeley (UCB) w ramach prac rozwojowych nad TCP/IP. Ponieważ nazwy większości tych narzędzi zaczynają się na literę „r”, są nazywane potocznie *r-narzędziami (r-utilities)*. Litera „r” pochodzi od *remote* — zdalne. Początkowo r-narzędzia stanowiły część składową

systemu operacyjnego Unix, wobec czego sa w duzym stopniu zalezne od platformy uniksowej. Z czasem jednak narzędzia te zaczely byc przenoszone na inne platformy i srodowiska, na przyklad Windows. Do popularnych r-narzędzi naleza rlogin, rsh i rexec. Wprawdzie ssh udostepnia uslugi zdalne podobne do r-narzędzi, lecz do nich nie nalezy. Jest to odrebny protokół, przypominajacy raczej Telnet.



r-narzędzia czasami nazywane sa *r-narzędziami Berkeley*.

Wprawdzie r-narzędzia sa nadal popularne w systemach operacyjnych Unix i Linux, lecz bardzo szybko zastępuja je standardowe usługi TCP/IP, takie jak FTP i Telnet, poniewaz:

- ◆ r-narzędzia przeznaczone sa jedynie do użytku wewnętrznego w *zaufanych sieciach*, na przykład położonych za zaporami firewall.
- ◆ Większość r-narzędzi udostępnia znakowy interfejs użytkownika (CUI — *Character User Interface*), który nie jest zbyt łatwy w użyciu, ponieważ użytkownik musi pamiętać związane z literami polecenia.

Obecnie r-narzędzia są bardzo rzadko implementowane w komercyjnych pakietach TCP/IP. Telnet oferuje dobrze znany i łatwy w obsłudze interfejs, dzięki czemu jest jedna z najbardziej popularnych usług protokołu TCP/IP.



Dodatkowe informacje o zaporach firewall można znaleźć w rozdziale 11.

Telnet

Początkowo uzyskanie dostępu do zdalnego komputera było operacją pracochlonną, wymagającą zmian w systemie operacyjnym komputera zadającego dostępu. Ponadto, z uwagi na niejednorodny charakter środowiska sieciowego, nie można było ustalić, jak naciśnięcia klawiszy będą interpretowane po drugiej stronie. Na przykład, kombinacja klawiszy *Ctrl+D*, służąca do zakończenia sesji w systemie lokalnym, niekoniecznie musiała zamknieć sesję w systemie zdalnym.

Krok po kroku, programiści systemów opracowali narzędzie pozwalające użytkownikom współdziałając ze zdalnym systemem tak, jakby pracowali z systemem lokalnym. Narzędzie to otrzymało nazwę *Telnet* od *TELecommunication NETwork*. Usługa Telnet ma wyższy priorytet od lokalnej interpretacji wszelkich naciśnięć klawiszy. Inaczej mówiąc, Telnet został opracowany jako usługa pozwalająca użytkownikom logować się do zdalnego komputera i wykonywać w nim polecenia tak, jakby siedzieli przy jego konsoli.

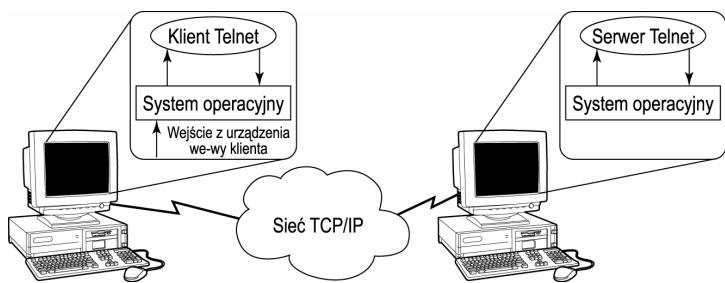


Telnet jako protokół jest starszy od reszty protokołów TCP/IP. Był on oryginalnym protokołem, na podstawie którego stworzono pakiet TCP/IP. Telnet jest też uważany za protokół „uniwersalny”, ponieważ może posłużyć do „recznego połączenia” z niemal wszystkimi innymi protokołami. Usługa Telnet jest zorientowana na połączenia, wobec czego opiera się na protokole TCP. Usługa Telnet korzysta z portu 23. TCP.

Telnet opiera się na trzech elementach: na *wirtualnym terminalu sieciowym* (NVT — *Network Virtual Terminal*), na *zasadzie negocjacji* i na *symetrycznym widoku terminali i procesów*.

- ♦ *Wirtualny terminal sieciowy (NVT)* — aby móc działać w niejednorodnym środowisku (współpracując z różnymi platformami i systemami), Telnet wykorzystuje NVT, który jest standardem reprezentowania danych i sekwencji sterujących. NVT jest implementacją architektury klient-serwer, w której oba punkty końcowe połączenia traktowane są jak wirtualne terminale (logiczne urządzenia wejścia-wyjścia). Logiczne urządzenie wejściowe — klawiatura użytkownika — generuje dane wychodzące, natomiast logiczne urządzenie wyjściowe (monitor) reaguje na nadchodzące dane i inne sygnały ze zdalnego systemu. Instrukcje wydawane na dowolnym z terminali wirtualnych są tłumaczone na odpowiednie polecenia dla urządzenia fizycznego. Inaczej mówiąc, program Telnet po stronie klienta (czyli stronie użytkownika, który zainicjalował zadanie w usłudze Telnet) odwzorowuje kody otrzymane od serwera na kody zrozumiałe dla klienta. Równoczesnie, kody generowane po stronie klienta zostają odwzorowane na kody NVT zrozumiałe dla serwera, które może je przetwarzac. Rysunek 13.1 przedstawia działanie usługi Telnet korzystającej z NVT.

Rysunek 13.1.
Komunikacja w usłudze
Telnet korzystającej z
NVT



NVT jest mechanizmem półdupleksem, pozwalającym wydawać w każdej chwili polecenia tylko po jednej stronie.

- ♦ *Zasada negocjacji* — niektóre systemy mogą świadczyć dodatkowe usługi, poza dostępnymi w NVT. W związku z tym, systemy korzystające z minimalnego zestawu usług nie są w stanie poprawnie komunikować się z drugim urządzeniem. Wobec tego, gdy dwa systemy komunikują się ze sobą za pomocą protokołu Telnet, parametry połączenia i terminali zostają ustalone podczas procesu łączenia, zasady i procesy, których dowolny z komputerów nie jest w stanie obsługiwać, są ignorowane. Eliminuje to potrzebe interpretowania informacji wymienianych pomiędzy komputerami po obu końcach połączenia. Na przykład, użytkownik może negocjować ustawienia opcji echa i zdecydować, czy echo powinno funkcjonować lokalnie, czy zdalnie.

Każda ze stron może, jeśli tego potrzebuje, zainicjalować podczas sesji dodatkowe negocjacje uzupełniające.



- ◆ *Symetryczny widok terminali i procesów* — składnia negocjacji jest symetryczna, co pozwala zarówno klientowi, jak i serwerowi zadać określonych opcji. Ten symetryczny widok terminali i procesów optymalizuje usługi świadczone przez drugą stronę połączenia. Telnet nie tylko pozwala na interakcje terminala ze zdalnymi aplikacjami, lecz pozwala również na interakcje pomiędzy dwoma procesami i pomiędzy dwoma terminalami.

Użytkownicy za pomocą Telnetu mogą:

- ◆ łączyć się z dostępna online baza danych, aby korzystać z zawartych w niej informacji,
- ◆ łączyć się z bazami wiedzy dostępnymi online, na przykład bibliotekami, i szukać w nich informacji,
- ◆ łączyć się z zdalnym systemem, aby korzystać z aplikacji, na przykład z poczty elektronicznej.

Proces połączenia w protokole Telnet

Połączenie telnetowe jest nawiązywane pomiędzy portami użytkownika i serwera. Jeden i drugi może „słuchać” wszelkich zadań związanych z usługą Telnet na porcie 23.



Serwer może obsługiwać jednocześnie wiele połączeń.

1. Aby wywołać sesję Telnet, użytkownik musi podać adres IP komputera docelowego (w przykładzie poniżej 132.45.78.44) lub nazwę skojarzoną z tym adresem (w przykładzie — lperry). Składnia polecenia może wyglądać następująco:

```
telnet 132.45.78.44
telnet lperry
```

Ponieważ Telnet akceptuje adresy IP, usługa ta może być stosowana nawet wtedy, gdy rozwiązanie nazwy na adres nie jest możliwe. Jeśli jednak nie podamy adresu IP lub nazwy komputera, to Telnet przejdzie w tryb zleceń, czekając na następne polecenia.

2. Teraz usługa zada podanie ID użytkownika i hasła. Aby zalogować się do zdalnego systemu, użytkownik potrzebuje poprawnego identyfikatora. Jeśli jednak komputer, z którego użytkownik łączy się ze zdalnym, jest hostem zaufanym, to hasło nie jest wymagane. Ekran logowania w usłudze Telnet jest przedstawiony na rysunku 13.2.
3. Jeśli identyfikator użytkownika i hasło są poprawnie zatwierdzone, połączenie Telnet zostaje nawiązane i komputer lokalny (przy którym użytkownik pracuje) zaczyna zachowywać się jak komputer zdalny.



Aby wyjść z sesji Telnet, trzeba użyć polecenia po stronie odbiorcy (zazwyczaj *Ctrl+D*). W systemach Windows zamknięcie okna Telnet kończy sesję.

Rysunek 13.2.

Telnet — ekran logowania

```

Telnet - 157.158.1.3
Pokaż Edycja Terminal Pomoc
Red Hat Linux release 7.0 (Guinness)
Kernel 2.2.16-22 on an i686
login: john
Password:
[john@server1 john]$ ls-1
bash: ls-1: command not found
[john@server1 john]$ ls -l
total 4
drwxr-xr-x    5 john      john        4096 May  8 18:42 Desktop
[john@server1 john]$ █

```

Najczesciej stosowane polecenia usługi Telnet

Telnet obsługuje szereg polecen, służących do sterowania procesem interakcji klient-serwer i szczegółami związanymi z tym procesem. Polecenia wysyłane są jako element danych, wymienianych przez oba komputery. Niemal wszystkie polecenia usługi Telnet składają się z przynajmniej dwóch bajtów. Pierwszy bajt zawiera znak ucieczki IAC (*Interpret As Command* — interpretuj jako polecenie), który służy do wprowadzenia następującego po nim polecenia. Następny bajt zawiera kod polecenia, które należy wykonac. Najczęściej stosowane polecenia Telnetu zostały przedstawione w tabeli 13.1. Ich składnia wygląda następująco:

IAC <kod_polecenia>



Powyzsze kody polecen mają znaczenie tylko wtedy, gdy poprzedza je znak ucieczki IAC.

Oprócz powyższych polecen, sterujących interakcją pomiędzy klientem i serwerem, dostępne są różnorodne opcje, które można negocjować pomiędzy dwoma punktami końcowymi połączenia w dowolnej chwili. Opcje te gwarantują, że oba systemy „zrozumieją” dodatkowe parametry wymiany danych. Polecenia związane z negocjowaniem opcji składają się z trzech bajtów. Pierwsze dwa są identyczne z bajtami w poleceniach ogólnych, trzeci stanowi kod opcji, której dotyczy polecenie. Tabela 13.2 wymienia opcje i ich kody polecen. Format polecen negocjujących opcje jest następujący:

IAC <kod_polecenia> <kod opcji>

Instalacja usługi Telnet

Ponieważ Telnet to jedna z najpopularniejszych usług stosu protokołów TCP/IP, zwykle jest ona z góry instalowana z systemem operacyjnym. Użytkownicy systemów Windows 9x, Windows NT, Windows 2000 nie muszą tej usługi instalować. Telnet jest też instalowany w uniksowych systemach operacyjnych.

Tabela 13.1. Polecenia w usłudze Telnet

Wartosc polecenia	Polecenie	Opis
240	SE	<i>Sub-negotiation End</i> — oznacza koniec fazy podnegocjacji.
241	NOP	<i>No Operation</i> — „nie rób nic”.
242	Data Mark	Poręca danych synchronizacji.
243	BRK	<i>BReaK</i> — polecenie przerwania.
244	IP	<i>Interrupt Process</i> (przerwij proces) — przerywa, porzuca lub konczy proces.
245	AO	<i>Abort Output</i> (porzuc wyjście) — wykonuje proces do konca, lecz nie wysyła wyników do klienta.
246	AYT	<i>Are You There</i> (jestes tam?) — odpytuje drugi koniec połączenia, by upewnić się, czy proces działa.
247	EC	<i>Erase Character</i> (usun znak) — usuwa znak z potoku wyjściowego.
248	EL	<i>Erase Line</i> (usun wiersz) — usuwa wiersz z potoku wyjściowego.
249	GA	<i>Go Ahead</i> (gotów) — zezwolenie na kontynuowanie komunikacji półdupleksowej.
250	SB	<i>Sub-Negotiation</i> (podnegocjacje) — inicjuje podnegocjacje, zazadane przez klienta.
251	Will	Sygnalizuje, aby drugi koniec połączenia emulował koniec przeciwny.
252	Won't	Odmowa emulacji.
253	Do	Potwierdzenie wykonania czynności.
254	Don't	Informacja o niewykonaniu czynności.
255	IAC	<i>Interpret As Command</i> (interpretuj jako polecenie) — następny lancuch należy traktować jak polecenie.



Jesli Telnet nie jest zainstalowany w komputerze, można pobrać odpowiednie oprogramowanie za darmo z Internetu. Jednym z najpopularniejszych miejsc w Internecie, w których dostępne są programy telnetowe, jest win3x.tucows.com/softterm.html.

Instalacja Telnetu na platformach Windows

W prawdziwej Telnet jest instalowany domyślnie w systemach operacyjnych Windows, lecz Czytelnik może zechcieć zainstallować solidniejszą wersję tej aplikacji, udostępniającą lepsze usługi od domyślnej wersji Telnetu. Do popularnych odmian programów telnetowych używanych na platformach Windows należą EWAN i QVTTerm. W ich przypadku instalacja przebiega następująco:

Tabela 13.2. Kody polecen negocjacji opcji

Wartosc opcji	Opcja	Opis
1	Echo	Powtarza zwrotnie znaki odebrane z drugiego konca polaczenia.
5	Status	Inicjuje wymiane biezacego statusu opcji uslugi Telnet.
24	Typ terminala	Inicjuje wymiane dostepnych typow terminala i wybiera najstosowniejszy.
31	Rozmiar okna	Inicjuje negocjacje rozmiaru okna dla potoku danych.
32	Predkosc terminala	Inicjuje negocjacje predkosci wymiany danych.
33	Zdalne sterowanie przeplywem	Inicjuje negocjacje, czy zezwolic podczas wymiany danych na sterowanie przeplywem, czy nie.
34	Tryb lacza	Inicjuje negocjacje, czy znaki w terminalu maja byc interpretowane po stronie klienta, czy po stronie serwera.

1. Rozpakuj plik instalacyjny, jesli ma postac spakowanego archiwum.
2. Kliknij dwukrotnie plik instalacyjny, aby rozpoczac proces instalacji. Przeprowadz ten proces zgodnie ze wskazówkami w pliku README, dolaczonym do aplikacji Telnetu.
3. Po zakonczaniu instalacji proces instalacji powinien utworzyc skrot w menu Start. Uruchom aplikacje.



Jesli w trakcie procesu instalacji program pyta o miejsce docelowe, w którym ma utworzyc folder aplikacji, to zaakceptuj domyslny folder instalacyjny.

Instalacja uslugi Telnet w systemie Macintosh

Aby zainstalowac Telnet na platformie Macintosh:

1. Skopuj plik instalacyjny do folderu docelowego.



Jednym z najczesciej uzywanych programow jest Mac Telnet 2.6 NCSA. Do innych popularnych aplikacji telnetowych naleza NiftyTelnet, DataComet i BetterTelnet.

2. Kliknij dwukrotnie plik, aby rozpakowac jego zawartosc.

3. Zaladuj aplikacje i wybierz *File/Open Connection*, aby wyswietlic okno dialogowe *Open Connection* (otwórz polaczenie). To okno dialogowe moze posluzyc do nawiazania polaczenia z odpowiednim hostem.



Aby zainstalowac aplikacje Telnet w systemie Macintosh, w komputerze musi byc zainstalowane odpowiednie oprogramowanie TCP/IP. Wersje systemu 8.0 i wyzsze wymagaja zainstalowania Open TransportPPP. Dla wersji starszych niz 8.0 odpowiednim oprogramowaniem TCP jest MAC TCP.

Instalacja uslugi Telnet na platformach uniksowych

Podobnie jak w innych popularnych systemach operacyjnych, na przyklad Windows, w systemie Unix Telnet jest zainstalowany domyslnie. Aby jednak zainstalowac solidniejsza wersje tej aplikacji, nalezy:

1. Pobrac wymagany plik z Internetu lub skoplowac z CD.
2. Rozpakowac plik za pomoca odpowiedniego narzedzia, jesli ma postac skompresowana. Jesli nie jest spakowany, ten krok jest zbedny.
3. Skoplowac plik instalacyjny do wybranego katalogu.
4. Przeczytac uwaznie plik z instrukcjami (*README* lub *INSTALL*) i zainstalowac oprogramowanie zgodnie z opisana procedura.



Poniewaz Telnet jest jednym z najpopularniejszych narzedzi, poswiecono mu wiele dokumentow RFC. Aby dowiedziec sie wiecej o roznych aspektach protokolu Telnet, mozesz skorzystac z RFC o numerach 854, 855, 856, 857, 858, 859, 860, 861, 927, 933, 1041, 1073, 1079, 1096, 1116, 1143, 1184, 1205, 1372, 1408, 1571, 1411, 1416, 1572, 2066 i 2217. Dokument RFC 854 jest juz nieaktualny, lecz zawiera szczegolowy opis protokolu.

Remote login

Remote Login (*rlogin*), dosl. *zdalne logowanie*, jest poleceniem uniksowym, ktore pozwala uzytkownikowi polaczyc sie ze zdalnym komputerem i zalogowac do niego. Usluga rlogin ma funkcjonalnosc podobna do Telnetu. Różnice pomiedzy interfejsami uzytkownika uslug rlogin i telnet w wiekszosci przypadkow sa niewidoczne. Istnieja jednak w sposobie utrzymywania komunikacji dwupunktowej i w charakterystyce sesji.

Po stronie nadawcy usługa rlogin wywolywana jest przez polecenie *rlogin*. Podobnie jak w usłudze Telnet, polecenie *rlogin*, aby zidentyfikowac odbiorca, przyjmuje w roli parametru adres IP lub nazwe docelowego hosta. Demon (watek w serwerze) o nazwie *rlogind* steruje usługa rlogin po stronie odbiorcy. Po pomyslnym nawiiazaniu polaczenia uzytkownik nie jest pytany o nazwe, lecz tylko o haslo. Usługa rlogin nie pozwala uzytkownikowi logowac sie do zdalnej usługi pod inną nazwą. Dozwolona jest jedynie zarejestrowana nazwa uzytkownika, stosowana przez komputer-odbiorce. Jest to podstawowa różnica pomiedzy usługami rlogin i Telnet (pozwalajaca logowac sie do systemu pod dowolna poprawna nazwa uzytkownika). Rysunek 13.3 przedstawia ekran rlogin.

Komputer docelowy zezwala odbiorcy na dostep jedynie wtedy, jesli spełnione sa nastepujace warunki:

- ♦ Plik */etc/hosts.equiv* w komputerze zdalnym zawiera wpis dla komputera odbiorcy.
- ♦ Plik *\$HOME/.rhosts* w komputerze zdalnym zawiera wpis dla komputera i nazwy uzytkownika, który zglosil zadanie polaczenia.



Aby zapobiec atakom z zewnatrz, tylko wlasiciel powinien miec prawa odczytu i zapisu w pliku *\$HOME/.rhosts*.

Rysunek 13.3.*Ekran rlogin*

```
[root@localhost /root]# rlogin 172.17.55.135
Password:
Last login: Wed May  9 01:22:03 from :
You have new mail.
[root@localhost /root]#
```

Proces połaczenia rlogin

Połaczenie z innym komputerem za pomocą usługi rlogin przebiega następująco:

1. Po stronie nadawcy zostaje wywołane zadanie połaczenia rlogin z adresem IP (np. 132.45.78.44) lub nazwa hosta odbiorcy (lperry). Składnia polecenia jest następująca:

```
rlogin lperry
rlogin 132.45.78.44
```

Do odbiorcy (serwera) zostają wysłane trzy lancuchy znakowe rozdzielone zerami. Pierwszy lancuch zawiera ID logowania użyty po stronie nadawcy. Drugi lancuch zawiera ID użytkownika, który posłuży do logowania do zdalnego systemu. Ten identyfikator jest identyczny z ID użytkownika używanym po stronie odbiorcy. Ostatni lancuch zawiera dodatkowe dane identyfikacyjne użytkownika (opcjonalnie) oraz predkosc transmisji, jaką stosować będzie nadawca.

2. Odbiorca po otrzymaniu lancuchów przekształca je na zmienne środowiskowe, które sterują metodą i różnymi szczegółami interakcji pomiędzy klientem i serwerem. Uzgodnienie odebranych parametrów (szczególnie predkosci transmisji) kończy proces logowania. Od tej chwili każdy znak wpisany po stronie klienta jest przesyłany do odbiorcy i vice versa.



Aby wyjść z sesji rlogin, należy nacisnąć kombinację klawiszy *Ctrl+D* lub w nowym wierszu wpisać znak ucieczki. Domyslnym znakiem ucieczki jest tylde (~), aczkolwiek niektóre wersje usługi rlogin wymagają kombinacji ~!.

Instalacja usługi rlogin

Narzędzie rlogin jest wbudowane w system operacyjny Unix i instalowane automatycznie razem z tym systemem. Jednakże w innych systemach operacyjnych, na przykład Windows i Macintosh, gdzie rlogin jest używany bardzo rzadko, niezbędne będzie narzędzie innych producentów. Do takich narzędzi należą SAMBA i PCNFS. Można pobrać je z Internetu, lecz mogą nie być darmowe. Aby zainstalować usługę, należy rozpakować archiwum (gdy jest taka potrzeba), a następnie postępować zgodnie ze wskazówkami zawartymi w dołączonym pliku z instrukcjami.



Dodatkowe informacje o rlogin mozna znalezc w RFC 1258.

Remote shell (rsh)

Narzedzie rsh sluzy do wykonywania polecen w systemie zdalnym, przy czym uzytkownik nie musi logowac sie do systemu, aby te polecenia wykonac. Pierwszym parametrem polecenia rsh jest adres IP lub nazwa zdalnego komputera. Drugim parametrem jest polecenie, ktore nalezy wykonac w zdalnym komputerze. Po stronie odbiorcy (serwera) polecenia, wydane po stronie klienta, sa wykonywane przez proces drugoplanowy o nazwie rshd.



Narzedzia rsh *nie wolno* mylic z powlokami uniksowymi, takimi jak powloka C (csh) lub Bourne'a.

Skladnia polecenia wyglada nastepujaco:

```
rsh <adres_serwera> <polecenie_zdalne>
```

Przykladami polecen rsh sa:

```
rsh lperry ls
rsh 132.45.78.44 ls
```

W pierwszym przykladzie lperry jest nazwa zdalnego komputera, w którym chcemy wykonac polecenie ls. Polecenie rsh nie zostanie wykonane pomyslnie, jesli w plikach *hosts.equiv* i *rhosts* nie istnieja odpowiednie wpisy oraz jesli pliki te sa nieobecne lub uszkodzone. Pliki powyzsze zawieraja informacje zwiazane z logowaniem, potrzebne dla komputerow, ktore beda zdalnie wykonywac polecenia.

W srodowiskach uniksowych nie trzeba instalowac narzedzia rsh, poniewaz jest wbudowane w system. Aby zainstalowac rsh w srodowisku Windows, na potrzeby lacznosci systemow Windows i Unix, nalezy:

1. Skopiwac pliki *rshsetup.exe*, *rshsvc.dll* i *rshsvc.exe* do folderu *System32*. Folder ten w systemach Windows 9x mieści sie w folderze *Windows*. W przypadku Windows NT 4.0 mieści sie on w folderze *WINNT (%SystemRoot%)*.
2. Dwukrotnie kliknac program *rshsetup.exe*, aby go uruchomic. Powinien pojawić sie komunikat, iż usługa Remote Shell została pomyslnie zainstalowana.
3. W wierszu poleceń wpisać *net start rshsvc*, aby uruchomic polecenie. Jesli pojawi się komunikat mówiący, że usługa remote shell została pomyslnie zainstalowana, oznacza to powodzenie instalacji rsh. Po uruchomieniu usługi należy skonfigurować plik *rhosts*, aby umożliwić dostęp klientom uniksowym.



Dodatkowe informacje o rsh mozna znalezc w RFC 1282 i RFC 1258.

Secure shell (ssh)

Narzędzie rsh nie jest uznawane za bezpieczne. Kazdy użytkownik, mający dostęp do komputerów w sieci jako administrator (root) lub mający dostęp do kanalu łączności, może uzyskać nieautoryzowany dostęp do systemu. Osoba taka może rejestrować cały ruch sieciowy wchodzący i wychodzący z systemu, łącznie z hasłami. Stanowi to poważne zagrożenie dla integralności poufnych danych przesyłanych przez sieć.

Tzw. *bezpieczna powłoka ssh (Secure Shell)* została opracowana przez Fina Tatu Ylonena, aby ominąć luki w bezpieczeństwie systemów, powodowane przez rsh i inne narzędzia. Narzędzie to szybko zyskało na popularności i z czasem stało się usługą, z której korzystają ponad 2 miliony użytkowników na całym świecie. Wprawdzie narzędzie ssh było na początku przeznaczone dla platform uniksowych, lecz jego popularność wpłynęła na przeniesienie go na inne platformy. Dzisiaj istnieją różne implementacje ssh:

- ♦ **SSH1** — pierwsza implementacja ssh, przeznaczona dla platform uniksowych. Był to jeden z pierwszych protokołów dostępnych dla użytkowników za darmo.
- ♦ **SSH2** — ta wersja zawiera wiele zmian w porównaniu z poprzednią i może być stosowana w systemach Unix, Macintosh oraz Windows. SSH1 i SSH2 różnią się szyfrowaniem pakietów. Ponadto SSH1 do uwierzytelniania używa kluczy serwera i hosta. W przeciwnieństwie do tej wersji, SSH2 używa jedynie klucza hosta. Wersja SSH2 jest również dostępna jako freeware, lecz z ograniczeniami w licencjonowaniu.



Organizacja IETF (*Internet Engineering Task Force*) podjęła dalsze prace rozwojowe nad SSH2, jednakże zmiany, których celem było wzmacnianie SSH2, spowodowały niezgodność tej implementacji z poprzednią.

- ♦ **LSH** — implementacja opracowywana jako darmowa wersja SSH2.
- ♦ **FreeSSH** — ta wersja nie wywodzi się z oryginalnego opracowania ssh autorstwa Tatu Ylonena. FreeSSH działa jedynie na platformach uniksowych i jest wciąż w fazie rozwoju.
- ♦ **sftp** — aplikacja FTP działająca przez tunel SSH. Narzędzie to jest przeznaczone jedynie dla systemów Unix i Linux.
- ♦ **MindTerm SSH** — darmowy klient ssh, napisany w języku Java, który może funkcjonować z wykorzystaniem graficznego interfejsu użytkownika (GUI) lub bez niego.
- ♦ **Klienty SSH dla Windows** — istnieją różnorodne klienty ssh dla systemów Windows:
 - ♦ **TTSSH** — ta wersja pod Windows jest darmowa aplikacja emulatora terminala.
 - ♦ **Putty** — darmowy klient Win32/ssh.
 - ♦ **WinSCP** — narzędzie dla systemu Windows, oferujące wyjątkowo łatwy w użyciu interfejs użytkownika.
- ♦ **OpenSSH** — najnowsza propozycja na rynku. Wersja ta obsługuje systemy Linux, FreeBSD, Unix, Solaris, AIX, IRIX oraz HP/UX.



Istnieja tez inne narzędzia ssh dla systemu Windows, dostepne jako freeware: iXplorer lub FiSSH. Programy F-secure SSH i VanDyke SSH sa komercyjnymi narzędziami SSH dla Windows.

Narzędzie ssh pozwala użytkownikowi logować się przez sieć do zdalnego komputera, wykonywać w nim polecenia i przenosić pliki z jednego komputera do drugiego. W porównaniu z r-narzędziami, ssh udostępnia silny mechanizm uwierzytelniania i bezpieczna komunikację przez nie zabezpieczone kanaly i podatne na ataki systemy operacyjne. Protokół ssh pozwala skutecznie tunelować ruch sieciowy dla klientów X Window. Oznacza to, że klient X Window może łączyć się z hostem X Window, a następnie, po uwierzytelnieniu, bezpośrednio korzystać z aplikacji X Window. Za pomocą ssh można w sposób przejrzysty dla użytkownika nawiązywać bezpieczne sesje zdalne. Ponadto dostęp do zdalnych klientów poprzez ssh jest dla użytkowników wygodny, ponieważ usługa ta dalej korzysta ze starych plików *.hosts* i */etc/hosts.equiv*, stosowanych przez rsh.



Dostępny jest mechanizm wycofania do rsh na wypadek, gdyby zdalny komputer nie obsługiwał ssh.

W transakcjach opartych na ssh ważne dane, jak np. hasła, są wysyłane w postaci zaszyfrowanej. Chroni to systemy przed uzyskaniem nieupoważnionego dostępu przez osoby ze złymi zamiarami. Ponadto ssh jest bardzo odporny na podszywanie się (*spoofing*), ponieważ stosuje złożone metody uwierzytelniania i bezpieczne metody komunikacji do przesyłania danych przez sieci. W atakach przez podszywanie się, zdalny host, który nie jest autoryzowanym członkiem danej sieci, wysyła pakiety udające, że pochodzą od zaufanego hosta w sieci. Gdy pakiety te uzyskają już dostęp do sieci, mogą służyć do podsłuchiwania poufnych informacji lub do włamania. Ataki przez podszywanie się mogą odbywać się lokalnie lub zdalnie.



Usluga ssh, gdy wykorzystuje TCP/IP, jest związana z portem 22.

Polaczenie ssh

Proces połączenia ssh wygląda następująco:

- ◆ Komputer standardowo oczekuje zadań ssh, wysyłanych przez inne węzły, na porcie 22. Polecenia ssh mają składnię:

```
ssh lperry who
ssh 132.45.78.44 who
```

- ◆ Po przechwyceniu zadania oba punkty końcowe połączenia wymieniają ze sobą lancuch identyfikacyjny, zakończony znakiem nowego wiersza (/n). Maksymalna długość lancucha (razem ze znakiem /n) wynosi 255 znaków. Zazwyczaj wymiana kluczy rozpoczyna się natychmiast, bez czekania na identyfikator drugiej strony.



Zainstalowanie ssh nie wymusza żadnych metod szyfrowania, kompresji ani kodów uwierzytelniania wiadomości (MAC). Parametry te są dobierane dynamicznie podczas wymiany kluczy.

- ♦ Dane podczas transmisji podlegają kompresji. Jeśli jednak nie została uzgodniona kompresja danych, to zaczyna się *wymiana kluczy*. Wymiana kluczy jest technika stosowana do generowania losowych kodów zabezpieczających (na przykład haseł) za zgódą obu stron zaangażowanych w transakcję. Kazda ze stron stosuje preferowany algorytm i zakłada, że druga strona również używa tego samego algorytmu. Nadawca może nawet zgodnie z algorytmem wysłać wstępny pakiet wymiany kluczy. Jeśli jednak odbiorca nie używa tego samego algorytmu, to nadawca i odbiorca ignorują pierwsze dane odebrane od drugiej strony, ustalają wspólny algorytm i wstępny pakiet wymiany kluczy zostaje wysłany ponownie.



Stosowane są dwa typy wymiany kluczy: RSA i Diffiego-Hellmana. W wymianie RSA do szyfrowania i odszyfrowania informacji stosowane są dwa odrebre klucze. Klucz szyfrujący jest publicznie udostępniany urządzeniom sieciowym, aby mogły za jego pomocą zaszyfrować swoje dane, lecz klucz deszyfrujący każdego urządzenia jest prywatny. Za pomocą tego prywatnego klucza deszyfrującego i dostępnego publicznie klucza szyfrującego każde urządzenie może bezpiecznie odszyfrować komunikat odebrany podczas transmisji. W wymianie Diffiego-Hellmana strony zaangażowane w transmisję uzyskują wspólny klucz tajny przez wymiane komunikatów i uwierzytelnianie drugiej strony za pomocą podpisu, który jest unikatowy dla każdego urządzenia sieciowego.

- ♦ Następnie klient wysyła własny komunikat uwierzytelniający hosta. Jeśli ten komunikat nie zostanie wysłany, serwer uzna na potrzeby uwierzytelnienia, że klient nie ma nazwy. Wiele serwerów nie „rozmawia” z klientami, które nie zostały uwierzytelnione.
- ♦ Po uwierzytelnieniu klienta i hosta zostaje wysłane zadanie usługi. Format tego zadania jest następujący:

"ssh <adres_IP/nazwa_hosta> <polecenie>"

Proces instalacji ssh

Aby zainstalować ssh, należy:

1. Pobrać oprogramowanie ssh spod jednego z wielu dostępnych adresów internetowych. Oficjalny punkt dystrybucji ssh to <ftp://ftp.cs.hut.fi/pub/ssh>.
2. W razie potrzeby rozpakować plik.
3. Przeczytać plik z instrukcjami (README) i wykonac następujące polecenia, aby zainstalować narzędzie:

```
./configure
make
make install
```

Narzędzie ssh zostanie zainstalowane z domyslną konfiguracją, która w zupełności wystarcza do używania programu. Użytkownicy zainteresowani dostosowaniem konfiguracji do własnych potrzeb znajdą dodatkowe informacje w pliku README.



Dodatkowe informacje o ssh zawiera RFC 793.

Remote execute (rexec)

Narzedzie rexec, podobnie jak ssh, pozwala wykonywac polecenia w zdalnym komputerze. Narzedzie to pojawiło się po raz pierwszy we wczesniejszych wersjach systemu Unix. Komputer, w którym polecenia będą wykonywane, do ich uruchomienia używa drugoplanowego procesu rexd. rexec działa podobnie do rsh, z dwiema różnicami:

- ◆ Hasło wysyłane z zadaniem jest szyfrowane, co utrudnia osobom niepowolanym przechwytcie hasel.
- ◆ Stosowany jest pełny proces logowania.

Podobnie jak rsh, rexec przyjmuje dwa parametry. Pierwszym z nich jest nazwa lub adres IP zdalnego komputera, natomiast drugi to polecenie, które należy wykonać po stronie zdalnego komputera. Przykładowy format polecenia jest następujący:

```
rexec lperry ls  
rexec 132.45.78.44 ls
```

Jesli plik `$HOME/.netrc` nie zawiera odpowiedniego wpisu dla komputera zdalnego, użytkownik jest pytany o ID logowania i hasło. Po podaniu informacji wymaganych do zalogowania, wynik wydanego polecenia jest wyświetlany po stronie klienta.

Podobnie jak inne narzędzia, rexec jest wbudowany w system Unix. Nie jest natomiast obsługiwany przez systemy Windows, dlatego niezbędne jest zainstalowanie oprogramowania innego producenta. Jednym z popularnych narzędzi, które możemy wykorzystać, jest Ataman TCP Remote Logon Service (ATRLS), udostępniający oprócz narzędzia rexec usługi rsh, rlogin i Telnet.



Narzędzie rexec jest obecnie używane bardzo rzadko, ponieważ rsh jest szybszy i wygodniejszy.

Oprócz narzędzi i standardowych usług typu ssh, FTP i Telnet, TCP/IP obsługuje narzędzie (usługa) oparte na interfejsie graficznym, noszące nazwę *serwera terminali* (*Terminal server*). Serwer terminali umożliwia wysoce bezpieczny dostęp przez sieć do zdalnych usług. Usługa ta ma duże możliwości, ponieważ może bez trudu obsługiwać równoczesnie wiele sesji. Ponadto instalacja sieciowa korzystająca z serwera terminali może znaczco obniżyć koszty eksploatacji sieci, ponieważ stacje robocze nie wymagają zbyt wiele obsługi (serwer terminali pełni funkcje magazynu danych i aplikacji). Co więcej, zastosowanie serwera terminali może ogromnie zmniejszyć poważność sieci na ataki z zewnątrz. Jesli jednak serwer terminali nie zostanie poprawnie zabezpieczony, jego awaria lub udany atak hakerów spowoduje wyłączenie całej sieci.

Serwery terminali

Serwer terminali jest potężnym narzędziem, służącym do dystrybucji i obsługi aplikacji oraz zarządzania nimi z jednego, centralnego miejsca. Serwery terminali udostępniają wieloużytkownikowe środowiska typu Unix, definiowane jako architektura *thin client* („lekkich klientów”). W takiej architekturze wszystkie aplikacje i przetwarzanie procesów

działają centralnie w serwerze terminali. *Thin client* jest bezdyskowym „gluchym terminaliem”, o ograniczonych zdolnościach do przetwarzania — może wystarczyć nawet sam monitor, klawiatura iłączność z siecią. Na rynku dostępne jest obecnie różnorodne oprogramowanie serwerów terminali, dające dostęp do różnych platform: Macintosh, Unix, Windows i Solaris. W czołówce konkurencyjnych na tym rynku produktów znajdują się Microsoft Terminal Server, serwer terminali Sun Ray firmy Sun oraz Citrix MetaFrame.



Gdy nazywamy komputer serwerem terminali, oznacza to, iż jest w nim zainstalowane oprogramowanie serwera terminali. Jednakże są też dostępne autonomiczne serwery terminali, posiadające odpowiednie oprogramowanie osadzone w systemie.

Serwer terminali składa się z trzech składników: wieloużytkownika rdzenia serwera, oprogramowania klienta serwera terminali i protokołu, stosowanego do komunikacji pomiędzy klientem i serwerem.

- ♦ *Wieloużytkownikowy rdzeń serwera* — udostępnia podstawowe zdolności do obsługi wielu równoczesnych sesji klientów oraz zawiera narzędzia administracyjne, służące do zarządzania serwerem i sesjami klientów.
- ♦ *Oprogramowanie klienta serwera terminali* — należy je zainstallować we wszystkich węzłach, które korzystają z dostępu do różnorodnych usług i aplikacji serwera terminali. Korzystanie z oprogramowania klienckiego może być również proste, jak praca z narzędziem Telnet.
- Klientem może być zarówno terminal „gluchy” (*dumb*), jak i „inteligentny”, posiadający własną moc obliczeniową.
- ♦ *Protokół* — służy do komunikacji pomiędzy serwerem terminali i różnymi klientami. Jednym z najlepiej znanych protokołów dla serwerów terminali jest RDP (*Remote Desktop* — pulpit zdalny), z którego korzysta Microsoft Terminal Server.

Sun Ray

Bedącą własnością firmy Sun oprogramowanie serwera terminali — Sun Ray — udostępnia wysoce skoncentrowane i bezpieczne zarządzanie i administrowanie systemem. W skład tej funkcjonalności wchodzą: uwierzytelnianie użytkowników, zarządzanie grupami serwerów oraz przekierowanie wejścia i wyjścia do *urządzeń klienckich Sun Ray* (*Sun Ray appliance*). Oprogramowanie serwera Sun Ray obejmuje również funkcje administracyjne, w tym zarządzanie zasadami uwierzytelniania. Całosc oprogramowania Sun Ray mieści się w serwerach terminali — w urządzeniach klienckich Sun Ray nic nie jest składowane.



Oprogramowanie serwera Sun Ray może być instalowane w serwerach SPARC, działających w środowisku operacyjnym (*Operating Environment*) Solaris 2.6, Solaris 7 lub Solaris 8.

Urządzenia klienckie Sun Ray są bezstanowymi komputerami thin client. Nazwa *urządzenia bezstanowe* oznacza urządzenie posiadające jedynie podstawowe składniki wejścia-wyjścia (np. klawiaturę, mysz i monitor). Urządzenia klienckie Sun Ray nie posiadają

w ogóle systemu operacyjnego, jedynie 8 MB pamięci RAM i 512 kB pamięci Flash EPROM. Urządzenia te posiadają specjalną opcję o nazwie *Hot desk* („aktywne biurko”), która pozwala użytkownikom łączyć się ze swoim pulpitem z dowolnego urządzenia klienckiego Sun Ray z wykorzystaniem osobistej karty intelligentnej lub poprzez zalogowanie się z odpowiednia nazwa użytkownika.



512 kB pamięci Flash zawiera oprogramowanie sprzętowe systemu, zajmujące się automatycznym testowaniem urządzenia po złączeniu (tzw. POST — Power-On Self Test), komunikacją ze wspólnym serwerem Sun Ray, uwierzytelnianiem, sterownikami urządzeń lokalnych i informacjami wyświetlonymi na ekranie.

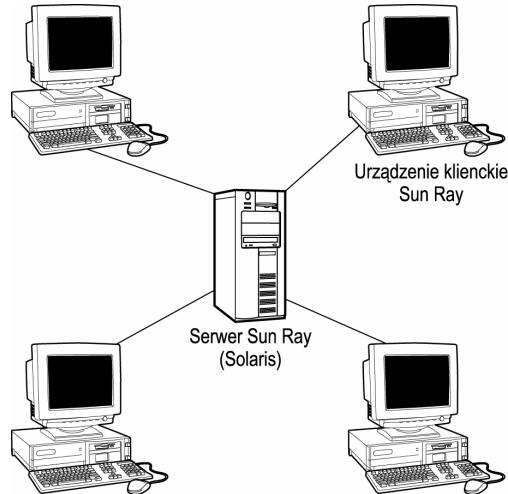
Oprogramowanie serwera Sun Ray zapewnia w pełni skonsolidowane sterowanie i bezpieczeństwo. Ponieważ urządzenia klienckie Sun Ray nie obsługują stacji dyskietek, indywidualnych ustawień zabezpieczeń ani opcji otwierania plików, uznanych przez system za niebezpieczne, odpowiednio skonfigurowany serwer jest bezpieczny.



Największa wada konfiguracji skoncentrowanej na serwerze jest znaczny ruch sieciowy.

Uruchomienie systemu terminali Sun Ray jest dosyć proste. Aby skonfigurować system, należy na serwerze zainstalować oprogramowanie Sun Ray Server i aplikacje użytkowników. Przez szybkie łącze Ethernet 10/100 Mb/s można do serwera terminali podłączyć do 30 węzłów. Kazdy węzeł składa się z monitora, myszy i klawiatury, połączonych z wezłem portami USB (*Universal Serial Bus* — standard uniwersalnej magistrali szeregowej). Dostęp do urządzeń periferyjnych (np. drukarek i skanerów) z zasadą możliwy jest tylko z serwera, a nie z poszczególnych węzłów. Konfiguracja systemu przedstawia rysunek 13.4.

Rysunek 13.4.
Instalacja Sun Ray



Oprogramowanie serwera Sun Ray zawiera kilka funkcji, służących do utrzymania sieci urządzeń klienckich Sun Ray i zarządzania nimi. Należą do nich: zarządzanie uwierzytelnianiem, zarządzanie sesjami, zarządzanie grupami, obsługa sterowników urządzeń wirtualnych oraz różne narzędzia administracyjne.

- ♦ *Menedżer uwierzytelniania* — dokonuje identyfikacji i uwierzytelniania klientów i użytkowników. Do tego celu domyślnie służy adres sprzetowy (Ethernet) klienta, opcjonalnie można zamiast niego zastosować typ i identyfikator karty inteligentnej (jeśli są dostępne). Zarejestrowani użytkownicy są akceptowani tylko wtedy, gdy zostali przed uwierzytelnieniem zarejestrowani w serwerze.



Karta inteligentna (*Smart Card*) jest mała, przenosna karta z tworzywa sztucznego, w której znajduje się mikroprocesor i pamięć. Karta inteligentna zawiera informacje wymagane do zalogowania użytkownika, dlatego może być wykorzystywana jako urządzenie służące do kontroli dostępu. Zastosowanie kart inteligentnych pozwala również udostępniać dane osobiste i handlowe jedynie odpowiednim użytkownikom. Karty te funkcjonują podobnie do kart płatniczych (np. kredytowych lub bankomatowych).

- ♦ *Menedżer sesji* — przypisuje sesje użytkownika w serwerze do fizycznego urządzenia klienckiego Sun Ray i wiąże oraz usuwa powiązania odpowiednich usług z określonymi urządzeniami klienckimi Sun Ray.
- ♦ *Menedżer grup* — śledzi przynależność do grup serwerów, a poza tym dokonuje statycznego rozkładu obciążenia oraz wyboru i przekierowania do serwerów.
- ♦ *Sterowniki urządzeń wirtualnych* — obsługują całość wejść-wyjść urządzeń klienckich Sun Ray.
- ♦ *Obsługa urządzeń periferyjnych* — zarządza urządzeniami przyłączonymi bezpośrednio do serwera Sun Ray. Dla urządzeń klienckich Sun Ray te urządzenia są typu zdalnego.
- ♦ *Narzędzia administracyjne* — różnorodne narzędzia, służące do zarządzania użytkownikami i monitorowania wykorzystania serwera.

Microsoft Terminal Server

Microsoft Terminal Server stanowi rozszerzenie systemu Windows NT Server 4.0, które udostępnia obsługę terminali dla rodziny systemów operacyjnych Windows (9.x oraz NT) oraz środowisko ultralekkich klientów (*super-thin client*), które pozwala wielu klientom zdalnie uruchamiać różne 16- i 32-bitowe aplikacje w centralnym serwerze. Pojęcie *ultralekkiego klienta* związane jest z systemem Microsoft Terminal Server, ponieważ dostęp do niego możliwy jest z różnych platform — biurkowych i nie tylko — do których należą, na przykład, Unix, Macintosh, terminale X Window, MS-DOS, komputery sieciowe itp.

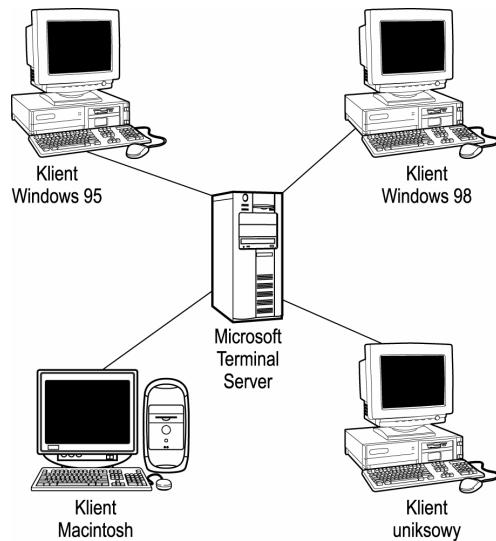


Microsoft Terminal Server nosi jeszcze jedną nazwę: Windows NT Server 4.0, Terminal Server Edition (TSE). Wraz z premierą nowego systemu operacyjnego Windows 2000, usługi terminalowe zostały zintegrowane w samym Windows 2000 Server i rozbudowane o nowe funkcje; są obecnie łatwiejsze do zainstalowania i jeszcze łatwiejsze w zarządzaniu. Na ich potrzeby nie trzeba nabywać dodatkowego oprogramowania serwera terminali.

Instalacja Microsoft Terminal Server składa się z wydajnego komputera, w którym znajdują się: oprogramowanie serwera terminali oraz różne aplikacje użytkowników. TSE może obsługiwać do 250 klientów. Klient może posiadać lokalny dysk twardy, może też nie mieć żadnego. Rysunek 13.5 przedstawia konfigurację usługi Microsoft Terminal Server.

Rysunek 13.5.

Instalacja Microsoft Terminal Server



Microsoft Terminal Server Edition składa się z trzech składników: *serwera terminali, ultralekkiego klienta i protokołu RDP*.

- ◆ *Serwer terminali (Terminal Server)* — udostępnia zdolność do równoczesnej obsługi wielu kompatybilnych klientów działających na różnych platformach, Windows i innych (odmiennych zarówno pod względem oprogramowania, jak i sprzętu). W architekturze serwera terminali procesem sterującym jest usługa Terminal Server (termsrv.exe). Odpowiada ona za inicjacje i konczenie sesji użytkowników, zarządzanie nimi oraz powiadomienia o zdarzeniach związanych z sesjami.
- ◆ *Ultralekki klient (super-thin client)* — wyświetla interfejs użytkownika 32-bitowego systemu Windows na różnorodnych platformach systemów operacyjnych, Windows i innych.
- ◆ Protokół RDP (Remote Desktop Protocol — protokół zdalnego pulpitu) — pozwala klientom lączyć się z serwerem terminali. Protokół ten jest kluczowym składnikiem usługi Microsoft Terminal Server. RDP opiera się na pakiecie standardowych protokołów komunikacyjnych ITU T.120 (International Telecommunications Union). Klient RDP może zostać zainstalowany w dowolnym kliencie Windows lub innym systemie operacyjnym.

Uslugi terminalowe są całkowicie niezależne od protokołu. Mogą korzystać z RDP lub protokołu innego producenta.



Klient łączy się z serwerem terminali i funkcjonuje w sposób następujący:

- ◆ Klient inicjuje połączenie z serwerem terminali przez port TCP. Na tym etapie, zanim klient będzie mógł zalogować się do serwera, pomiędzy klientem i serwerem negocjowane są szczegóły licencjonowania. W przypadku klientów Windows licencja weryfikowana jest w komputerze, który zada połączenia. W pozostałych przypadkach wydawana jest licencja na połączenie, aby klient mógł połączyć się z serwerem terminali.

- ♦ Po ustaleniu szczegółów sesji użytkownikowi zostaje wyświetlony standardowy ekran logowania Windows NT. Po wpisaniu nazwy użytkownika i hasła odbywa się uwierzytelnienie konta, aby sprawdzić, czy użytkownik ma prawo do zalogowania się. Jeśli klient jest zarejestrowany w serwerze terminali, zostaje użytkownikowi wyświetlony pulpit serwera terminali.
- ♦ Gdy użytkownik wybiera aplikacje, polecenia zostają przekazane do serwera terminali, który uruchamia aplikacje. Jeśli użytkownik rozłączy sesję przez pomyłkę (bez wylogowania), procesy i pamięć zajmowana przez sesję nie są zwalniane. Przy ponownym przyłączeniu się użytkownika, istniejąca sesja zostaje ponownie załadowana, jakby nic się nie wydarzyło. Jeśli jednak użytkownik wyloguje się z sesji, wszystkie związane z nią procesy zostają zakończone, a pamięć przydzielona sesji zostaje zwolniona.



Jeśli dla danego klienta skonfigurowane jest logowanie automatyczne, ekran logowania nie zostaje wyświetlony. W tym przypadku, w celu zalogowania, do serwera terminali zostają przesłane zaszyfrowane hasło i nazwa użytkownika.

Citrix

Citrix MetaFrame jest bazującym na serwerze oprogramowaniem thin client. Stanowi ono rozszerzenie opartej na systemie Windows usługi Terminal Services Microsofta oraz zdalnych usług dla klientów uniksowych. Zapewnia ono kompletne rozwiązanie serwerowe, rozszerzając funkcjonalność klienta i serwera. Obejmuje obsługę środowisk niejednorodnych, zarządzanie na skali przedsiębiorstwa i integracje bez „szwów”.

Citrix wykorzystuje protokół ICA (*Independent Computing Architecture*), który rozszerza funkcjonalność Microsoft Terminal Services po stronie klienta i serwera, pozwalając na obsługę różnych klientów, oraz dostęp do klientów uniksowych, przypominający serwer terminali. Rozwiązanie to pozwala na ekonomiczną instalację i dostęp do aplikacji oraz zarządzanie nimi poprzez sieć, niezależnie od platformy klienta i typu łącznia sieciowego.



Aktualna wersja Microsoft Terminal Server jest podziobrem usług terminalowych oferowanych przez Citrix MetaFrame. Microsoft opracował swój serwer terminali we współpracy z firmą Citrix (która włożyła większość pracy). Rozwiązanie Citrix jest bardziej wszechstronne w wyniku umowy wypracowanej pomiędzy obydwoma firmami.

Protokół ICA, stosowany przez Citrix MetaFrame, obsługuje szeroki zakres platform klienckich, w tym DOS, Windows, OS/2, Unix i Linux, a poza tym pozwala uruchamiać sesje w serwerze MetaFrame aplikacjom w różnych urządzeniach podłączonych do internetowych. Dobrym rozwiązaniem jest instalacja MetaFrame w serwerze Solaris — zwłaszcza na potrzeby rosnącej regularnie grupy klientów mobilnych, na przykład sprzedawców — z uwagi na łatwość instalacji i konfiguracji.

Dodatkowe możliwości po stronie klienta i serwera, które daje Citrix w systemie Microsoft Terminal Server obejmują: zarządzanie serwerami i klientami typu thin client, obsługę mieszanych klientów, sieci i protokołów oraz integracje pulpitu „bez szwów” z aplikacjami uruchamianymi zdalnie lub lokalnie.

- ♦ Zarządzanie serwerami i klientami typu thin client — Citrix rozszerza zestaw narzędzi administracyjnych, dostępnych w Microsoft Terminal Server. Na przykład, udostępnia dodatkowe narzędzia do zarządzania użytkownikami, systemami i aplikacjami na skali przedsiębiorstwa, mogącą pomóc administratorom

kontrolowac wersje oprogramowania, obslugiwac zdalnych uzytkowników, rozwiazywac problemy z konfiguracja i wykorzenic powielanie danych z różnych galezi sieci przedsiebiorstwa.



W przypadku zwiększych potrzeb uzytkowników, serwery bazujace na usludze Citrix MetaFrame moga obslugiwac ogromne liczby uzytkowników przez dodawanie kolejnych serwerów. Organizacje, które buduja duze sieci na skale przedsiebiorstwa moga, za pomoca serwera terminali opartego na Windows, nadzorowac, skalowac „farmy serwerów” i zarzadzac nimi z jednego miejsca, redukujac w ten sposob calkowity koszt posiadania.

- ◆ *Obsluga srodowisk mieszanych* — umozliwia dostep do szerokiej gamy aplikacji w niejednorodnych srodowiskach, zlozonych z różnych komputerów biurkowych, typów sieci i systemów operacyjnych. Mozliwa jest obsluga praktycznie dowolnego typu sprzetu (PC, komputery sieciowe, urzadzenia bezprzewodowe i tak dalej) oraz systemów operacyjnych (MS-DOS, Windows 3.x, Windows 9.x, Windows NT, Unix, OS/2, Mac OS, Java Virtual Machine itd.). Citrix MetaFrame moze korzystac z dowolnych polaczen sieciowych: LAN, WAN, telefonicznych, Internetu i intranetów. Ponadto Citrix obsluguje różnorodne protokoly, w tym TCP/IP, IPX/SPX, SLIP, PPP i NetBIOS.
- ◆ *Integracja pulpitu* — umozliwia przezroczysty dostep do szerokiej gamy aplikacji, opartych np. na Windows, jezyku Java lub przegladarce WWW. Chociaz aplikacje sa wykonywane w serwerze terminali, to zachowuja sie tak, jakby byly uruchomione w systemie uzytkownika.

Citrix funkcjonuje w sposob bardzo podobny do Microsoft Terminal Server. Citrix udostepnia uslugi terminalowe zarówno dla klientow Windows, jak i klientow uniksowych. Dowolny komputer z uruchomionym klientem MetaFrame for Unix 1.0 moze otworzyc sesje z serwera terminali opartym na hostie MetaFrame. Zgodnie ze specyfikacija firmy Citrix, w roli serwera terminali musi zostac wykorzystany system Sun Solaris 2.6 lub 2.7 ze Sparc lub Intel MetaFrame, poniewaz udostepnia skalowalnosc systemu Unix i pozwala na latwe uruchamianie starych aplikacji dla klientow mobilnych i typu thin client.



Aplikacje stare (typu *legacy*) oznaczaja aplikacje uruchomione na platformie Unix.

Citrix MetaFrame posiada szereg zalet w porównaniu z innymi serwerami terminali, miedzy innymi:

- ◆ Obsluge zarówno Microsoft Terminal Server, jak i serwerów uniksowych.
- ◆ Przejrzysta integracje z olbrzymia liczba typów klientow.
- ◆ Bezproblemowa współpraca z laczami o przepustowosciami nawet 15 kb/s w przypadku szkieletu uniksowego. Prowadzi to do szybszych czasów reakcji i redukcji ogólnego obciążenia lacz WAN.
- ◆ Mozliwosc monitorowania sesji uzytkowników i zdolnosc do przejecia w razie potrzeby sesji przez administratora. Opcja ta jest bardzo przydatna przy rozwiazywaniu problemów.

- ♦ Latwosc instalacji oraz niewielkie dodatkowe koszty utrzymania.

Rozdział 14.

Drukowanie przez siec

W tym rozdziale:

- ◆ Drukowanie plików w srodowisku uniksowym
- ◆ Drukowanie plików w systemach Microsoftu
- ◆ Laczenie z lokalna drukarka
- ◆ Laczenie z drukarka sieciowa
- ◆ Wprowadzenie do protokolu IPP Microsoftu

Niniejszy rozdział omawia drukowanie w srodowiskach uniksowych i linuksowych oraz w systemach Microsoftu. Zostały w nim opisane procedury drukowania w systemach operacyjnych Microsoftu i systemach Unix/Linux, konfiguracja drukarek na obu platformach, procedury konfiguracji serwera lpd oraz Microsoft Internet Printing Protocol.

Jedna z podstawowych zalet korzystania z TCP/IP jest zdolosc do drukowania przez siec, zarówno w systemach Windows, jak i z komputerów uniksowych i linuksowych. Z tego powodu biezacy rozdzial omawia drukowanie na drukarkach lokalnych i sieciowych.

Wprowadzenie do drukowania

W biznesowym srodowisku sieciowym drukowanie jest jedna z najczesciej wykonywanych codziennych czynosci. Wyslanie zadania do wydrukowania nie wyglada na trudne, lecz konfigurowanie drukarki moze sprawic klopoty. Drukarki mozna ogólnie podzielic na dwie kategorie: lokalne i sieciowe. Drukarka lokalna jest przylaczona tylko do jednego komputera, wobec czego tylko uzytkownik tego komputera moze wydawac polecenia drukowania. Drukarki lokalne sa przydatne dla uzytkowników pracujacych w domu przy komputerze osobistym. W przeciwienstwie do nich, drukarki sieciowe mozemy najczesciej spotkac w duzych organizacjach, gdzie przydzielanie osobnej drukarki dla kazdego pracownika byloby nieopłacalne i niepraktyczne. W tym przypadku drukarka sieciowa moze zostac skonfigurowana przy serwerze, gdzie wszyscy uzytkownicy sieci beda mogli z niej korzystac. Wówczas uzytkownicy serwera moga wysylac zadania do drukarki przylaczonej do serwera. Drukarki lokalne i sieciowe mozemy przylaczac do komputerów uzywajacych zarówno systemów Linux, jak i Windows.

Drukowanie w srodowisku linuksowym

Uslugi drukowania sa dostepne we wszystkich dystrybucjach Linuksa oferowanych przez różnych producentów, aczkolwiek pomiedzy poszczególnymi wersjami moga wstepowac drobne różnice. W tym podrozdziale Czytelnik dowie sie, jak zarzadzac zadaniami drukowania w jednej z najpopularniejszych wersji systemu operacyjnego Linux — dystrybucji Red Hat. Red Hat Linux zawiera szereg programów, plików i katalogów, pomagajacych w procesie drukowania.

W systemie Red Hat Linux użytkownicy moga drukowac pliki na centralnej drukarce. Mozna skonfigurowac komputer linuksowy tak, by drukował na drukarce podlaczonej do serwera linuksowego lub do serwera Windows NT/2000. W tym systemie za buforowanie w serwerze plików przeznaczonych do drukowania odpowiada program usługowy drukarki wierszowej — *line printer daemon (lpd)*, który przyjmuje pliki od klientów i składuje je w serwerze, dopóki drukarka nie bedzie gotowa do drukowania.

Program lpd musi byc uruchomiony w systemie *zawsze*, gdy chcemy korzystac z uslug drukowania. Mozemy sprawdzic, czy lpd jest uruchomiony za pomoca polecenia statusu *lpc* z wiersza polecen. Jesli chcemy, by lpd uruchamial sie podczas uruchamiania systemu, mozemy to skonfigurowac za pomoca polecenia *ntsysv*.



Buforowanie drukowania (*spooling*) jest procesem, w którym demon drukowania zapisuje dokumenty na dysku, a nastepnie wysyla je do drukarki.

Aby zrozumiec dokladnie proces drukowania, Czytelnik musi poznac pojecia zwiazane ze składnikami uslug drukowania w systemie Linux: plik urzadzenia drukarki, buforowanie drukowania, kolejki wydrukowów oraz plik */etc/printcap*. Składniki te zostana omówione w ponizszych podpunktach.

Plik urzadzenia drukarki

Urzadzenia sa w systemie reprezentowane w postaci plików urzadzen (*device file*). Drukarki sa urzadzeniami dzialajacymi w trybie znakowym i mieszcza sie w katalogu */dev*. Plik urzadzenia drukarki reprezentuje drukarke przylaczona do portu równoleglego (*parallel line printer*).

Pliki drukarek wygladaja nastepujaco:

```
# ls -l /dev/lp*
crw-rw---- 1 root lp 6, 0 Aug 24 2000 /dev/lp0
crw-rw---- 1 root lp 6, 1 Aug 24 2000 /dev/lp1
crw-rw---- 1 root lp 6, 2 Aug 24 2000 /dev/lp2
```



Port oznacza interfejs, za pomoca którego mozna przylaczyc urzadzenie sprzętowe do komputera. Port moze byc wewnetrzny lub zewnetrzny. Porty wewnetrzne sluzą do przylaczania takich urzadzen, jak dyski twarde. Porty zewnetrzne sluzą do przylaczania modemów, myszy, drukarek itp.

Buforowanie drukowania (spooling)

Angielski termin *spool* jest skrótem od *Simultaneous Peripheral Operations On Line* (równoczesne operacje na dolaczonych urzadzeniach peryferyjnych). *Buforowanie drukowania* jest procesem, w którym zadania drukowania sa zapisywane na dysku w plikach zrozumialych dla drukarki, a nastepnie wysylane do wydrukowania. Buforowanie tymczasowo przydziela pamiec dla wydanego zadania drukowania. Jesli wiec drukarka jest zajeta, moze ukonczyc biezace zadanie, a nastepnie rozpoczac kolejne. Procesem drukowania zajmuje sie demon drukarki wierszowej (*line printer daemon*).

Kolejka do drukowania

Kolejka do drukowania (print queue) oznacza liste zadan wysłanych do drukarki w celu drukowania, lecz jeszcze nie wydrukowanych. Inaczej mówiac, kolejka do drukowania sklada sie z zadan oczekujacych w potoku do drukowania. Dokumenty znajdujace sie w kolejce mozna przejrzec za pomoca polecenia `lpq`.

Plik printcap

Wszystkie niezbedne informacje o skonfigurowanych drukarkach sa obecne w pliku `/etc/printcap`. Demon drukarki wierszowej uzywa tych informacji do zarzadzania buforowaniem drukowania. Plik `printcap` zawiera również dane atrybutów — wymagane czcionki, marginesy, odstepy na papierze oraz protokół sluzacy do komunikacji z drukarka.

Ponizej przedstawiony zostal przykładowy plik `printcap`:

```
# /etc/printcap
#
# Please don't edit this file directly unless you know what you are
doing!
# Be warned that the control-panel printtool requires a very strict
format!
# Look at the printcap(5) man page for more info.
#
# This file can be edited with the printtool in the control-panel.

##PRINTTOOL3##    SMB
lp:\                :sd=/var/spool/lpd/lp:\\
:mx#0:\             :sh:\\
:af=/var/spool/lpd/lp/acct:\\
:lp=/dev/null:\\
:if=/usr/lib/rhs/rhs-printfilters//smbprint:
```

W tym przykladzie dodana zostala tylko jedna drukarka, podlaczona do urzadzenia SMB (*Server Message Block*). Tabela 14.1 objasnia znaczenie wpisów w pliku.

Uslugi drukowania korzystaja z określonych plików systemowych, wymienionych w tabeli 14.2.

Tabela 14.1. Wpisy w pliku printcap

Wpis	Znaczenie
:sd=/var/spool/lpd/lp:\`	Polozenie katalogu sluzacego do buforowania drukowania.
:mx#0:\`	Maksymalny rozmiar pliku. Zero oznacza brak ograniczen dla rozmiaru pliku.
:sh:\`	Ten wpis oznacza, ze drukarka nie powinna drukowac nagłówków stron.
:af=/var/spool/lpd/lp/a cct:\`	Nazwa pliku ewidencyjnego.
:lp=/dev/null:\`	Nazwa urzadzenia.
:if=/usr/lib/rhs/ rhs- printfilters//smbprint:	Nazwa filtra wejsciowego, który jest równoczesnie odpowiedzialny za rozliczanie uslugi.

Tabela 14.2. Pliki uzywane przez uslugi drukowania w systemie Linux

Nazwa pliku	Opis
/etc/passwd	Ten plik sluzy do identyfikacji uzytkownika w celu kontroli praw dostepu do drukarki.
/etc/printcap	Plik zawierajacy baze danych mozliwosci drukarek (<i>Printer Capability Database</i>).
/usr/sbin/lpd	Demon (program uslugowy) drukarki wierszowej.
/var/spool/lpd/*	Katalogi sluzace do buforowania drukowania.
/var/spool/lpd/*/cf*	Pliki sterujace demonem <i>lpd</i> .
/var/spool/lpd/*/df*	Pliki danych wyszczególnione w plikach <i>cf*</i> .
/var/spool/lpd/*/tf*	Kopie tymczasowe plików <i>cf*</i> .

Drukowanie w systemach Microsoftu

Systemy operacyjne Microsoftu sa powszechnie stosowane, wobec czego wzarna jest znajomosc sposobów zarzadzania zadaniami drukowania w systemach Windows. Proces drukowania w komputerze Windows moze byc opisany w postaci nastepujacej procedury:

1. Uzytkownik decyduje sie wydrukowac dokument.
2. Uzytkownik wysyla dokument do drukowania z systemu Windows albo z komputera nie uzywajacego tego systemu operacyjnego. Jesli zadanie drukowania uruchomione jest w systemie Windows, uzytkownik wykorzystuje aplikacje o nazwie GDI (*Graphics Device Interface* — interfejs urzadzenia graficznego). GDI komunikuje sie ze sterownikiem drukarki, skojarzonym z drukarka, do ktorej zadanie drukowania ma zostac wyslane. Nastepnie GDI i sterownik wymieniaja dane i przygotowuja zadanie drukowania w jezyku drukarki. Drukarka interpretuje zadanie, ktore zostaje przeslane do programu buforujacego po stronie klienta. Jesli zadanie drukowania jest uruchomione w komputerze innym niz Windows, to GDI zostaje zastapiony przez inny skladnik, wlasciwy dla uzywanego systemu

operacyjnego, który wykonuje niezbędne czynności. GDI jest używany tylko z systemami operacyjnymi Windows 2000.

3. Klient przesyła zadanie do serwera drukowania. W przypadku klientów Windows 2000 i Windows NT, program buforujący po stronie klienta dokonuje połączenia RPC (*Remote Procedure Call*) z serwerem. RPC po stronie klienta za pomocą routera łączy się ze zdalnym dostawcą usługi drukowania. Następnie zdalny dostawca usługi drukowania wysyła kolejne wywołanie RPC do programu buforującego serwera, który następnie odbiera przez sieć zadanie drukowania.
4. Serwer drukowania identyfikuje zadania wysłane przez komputery Windows jako typ danych EMF (*Enhanced Metafile*). Zadania wysłane przez aplikacje z systemów innych niż Windows 2000 są w większości identyfikowane jako gotowe do drukowania — *typ danych surowych* (RAW). Ten typ danych nie pozwala zmieniać lub modyfikować zadań przed wydrukowaniem.
5. Ruter serwera drukowania jest odpowiedzialny za przesłanie zadania drukowania do lokalnego dostawcy drukowania w serwerze. Następnie lokalny dostawca drukowania buforuje zadanie drukowania — inaczej mówiąc, zapisuje je na dysku.
6. Procesor wydruku odbiera zadanie drukowania po zidentyfikowaniu typu danych w tym zadaniu i, w zależności od typu danych, dokonuje odpowiedniej konwersji.
7. Jeśli podanie nazwy docelowej drukarki konfiguruje komputer kliencki, to usługa serwera drukowania decyduje, czy program buforujący serwera powinien zmienić zadanie drukowania, czy przydzielić do zadania inne dane. Zadanie drukowania jest następnie przekazywane do lokalnego dostawcy usług drukowania, a później zapisane na dysku.
8. Sterowanie zadaniem drukowania zostaje przekazane do procesora strony rozdzielającej. Ten procesor dodaje na początek zadania stronę rozdzielającą, jeśli została zazadana.
9. Zadanie zostaje przekazane z bufora do monitorów drukowania. W przypadku drukarek dwukierunkowych monitor języka zarządza dwustronna komunikacja pomiędzy nadawcą i drukarką. Zadanie zostaje przekazane do monitora portu. Jeśli drukarka nie jest dwukierunkowa, zadanie drukowania przechodzi bezpośrednio do monitora portu, który wysyła zadanie do docelowej drukarki.
10. Po odebraniu zadania drukarka przetwarza każdą stronę do formatu grafiki rastrowej i drukuje ją.

Drukowanie z klienta

Komputery w większości organizacji należą do wewnętrznych sieci, które pozwala użytkownikom udostępniać wzajemnie pliki i zasoby, a także korzystać z zasobów serwerów — na przykład z drukarek. W sieci klient-serwer użytkownicy łączący się z serwerami noszą nazwę klientów. Aby możliwe było drukowanie dokumentów, pomiędzy klientem i serwerem musi istnieć połączenie. Możliwych jest kilka konfiguracji połączenia: klient uniksowy lub linuksowy z serwerem uniksowym lub linuksowym, klient uniksowy lub linuksowy z serwerem Windows, klient Windows z serwerem uniksowym lub linuksowym oraz klient Windows z serwerem Windows.

Aby polaczyc klienta Windows z serwerem Windows, potrzebny jest protokol SMB bedacy własoscia Microsoftu. Mówiac o drukowaniu przez TCP/IP, będziemy zawsze mieli na mysli lpr i lpd — uslugi drukowania w srodowisku TCP/IP.

Konfiguracja serwera lpd

W serwerach linuksowych mozemy skonfigurowac piec typów drukarek; wybrany typ zależy od naszych wymagan. Wymagania te zaleza od stosowanego systemu operacyjnego, od uzytkowników sieci oraz od systemu operacyjnego serwera, w którym zamierzamy zainstalowac drukarke sieciowa.

- ◆ *Lokalna drukarka podlaczona do komputera linuksowego* — drukarka podlaczona jest do portu równoleglego w komputerze linuksowym. Jedynie uzytkownik komputera, w którym drukarka jest zainstalowana moze z niej korzystac.
- ◆ *Zdalna drukarka uniksowa lub linuksowa (lpd)* — podlaczona do komputera uniksowego lub linuksowego. Uzytkownicy tych systemów moga zgłaszac do danego komputera zadania drukowania.
- ◆ *Drukarka pod Windows 9x lub Windows NT (SMB)* — drukarka fizycznie podlaczona do komputera Windows 9x lub NT. Uzytkownicy Linuksa, chcacy korzystac z tej drukarki, musza skonfigurowac usluge SMB.
- ◆ *Drukarka w systemie Novell Netware (NCP)* — przylaczona do serwera Novell Netware.
- ◆ *Drukarka sieciowa (direct to port printer)* — nie przylaczona do komputera (serwera), lecz bezposrednio do sieci. Wszyscy uzytkownicy w sieci moga z niej korzystac bezposrednio.

Zdalne drukarki w systemach Unix i Linux

Zanim zaczniemy korzystac z drukarki przylaczonej do zdalnego komputera uniksowego lub linuksowego, musimy podac nastepujace informacje:

- ◆ nazwe drukarki przylaczzonej do zdalnego komputera,
- ◆ polozenie katalogu uzywanego do buforowania drukowania,
- ◆ maksymalny rozmiar tego katalogu,
- ◆ nazwe hosta zdalnego komputera,
- ◆ filtr wejsciowy — czyli sterownik drukarki.



Komputer lokalny posiada katalog buforowania wydruku na potrzeby lpd. Jest on uzywany tylko wtedy, gdy drukarka jest zajeta lub niedostepna; wówczas zadania drukowania z lokalnego komputera czekaja w obszarze buforowania, dopóki nie beda mogly zostac wyslane.

Zmiany dokonane podczas konfiguracji zdalnej drukarki linuksowej zostaja odzwierciedlone w pliku */etc/printcap*. Dla drukarki pojawiaja sie wpisy :rm i :rp. Parametr

`rm` zawiera nazwe zdalnego hosta, zas parametr `rp` definiuje nazwe zdalnej drukarki (`lp0`, `lp1` i tak dalej).

Aby wydrukowac plik na zdalnej drukarce, musimy otrzymac od niej wymagane zwolnienie. Inaczej mówiac, zdalny komputer, do którego drukarka jest przyłączona, musi zaakceptowac zadanie drukowania z komputera klienckiego.

Narzedzie printtool

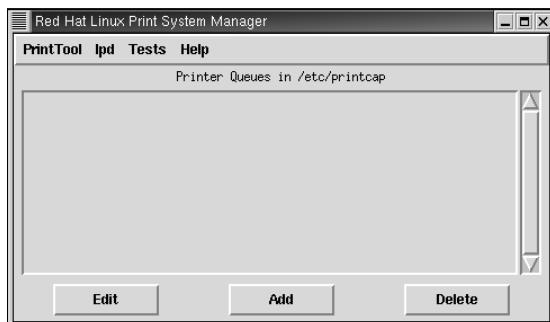
printtool jest bardzo wygodnym narzedziem, sluzacym do konfiguracji drukarek w systemie Linux. Aby je uruchomic, nalezy w wierszu polecen wpisac `printtool`. Narzedzie to moze sluzyc do konfiguracji lokalnej drukarki, zdalnej drukarki uniksowej (`lpd`), drukarek SMB, Windows 95 lub NT, drukarek NetWare i drukarek sieciowych.

Laczenie z lokalna drukarka

Narzedzie printtool moze posluzyc do skonfigurowania drukarki lokalnej w naszym komputerze. Zanim zaczniemy konfigurowac drukarke, musimy upewnic sie, czy jest ona zgodna z systemem Linux. Nastepnie, polecenie `printtool` wydane z wiersza polecen spowoduje otwarcie okna dialogowego Red Hat Linux Print System Manager (menedzeru systemu drukowania Red Hat Linux), przedstawionego na rysunku 14.1.

Rysunek 14.1.

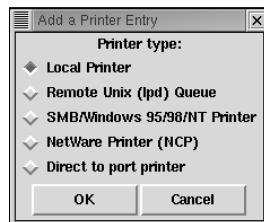
Okno powitalne
narzedzia printtool



Teraz kliknij przycisk *Add* (Dodaj), aby zainstalowac nowa drukarke. Pojawi sie okno dialogowe *Add a Printer Entry* (Dodaj wpis drukarki), przedstawione na rysunku 14.2.

Rysunek 14.2.

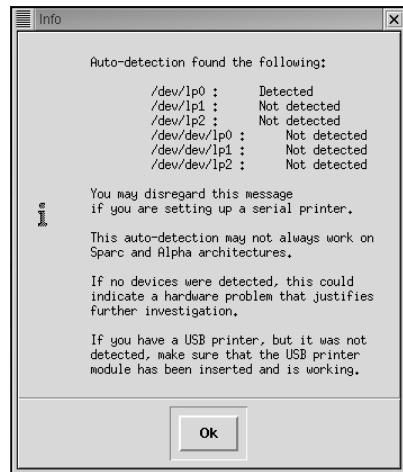
Okno dialogowe
Add a Printer Entry



W tym oknie wymienionych jest piec różnych opcji konfiguracji drukarek. Do wyboru mamy instalacje drukarki lokalnej, zdalnej kolejki uniksowej (`lpd`), drukarki SMB, Windows 95 lub NT, drukarki NetWare (NCP) oraz drukarki sieciowej (*Direct to port*).

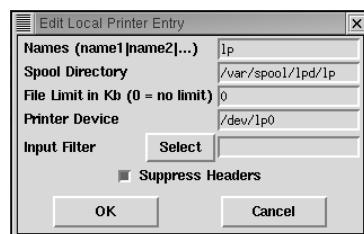
Wybierz *Local Printer* (drukarka lokalna) i kliknij *OK*. Pojawi sie okno dialogowe *Info*, przedstawione na rysunku 14.3.

Rysunek 14.3.
Okno dialogowe Info



To okno dialogowe wyświetla informacje o wykrytych urządzeniach drukarek na portach równoległych. Dodatkowy komunikat w tym oknie wskazuje, iż jeśli nie zostały wykryte żadne urządzenia, może istnieć problem ze sprzętem. Kliknij przycisk *OK*, aby przejść dalej. Teraz pojawi się okno dialogowe *Edit Local Printer Entry* (Edytuj wpis dla lokalnej drukarki), pokazane na rysunku 14.4.

Rysunek 14.4.
Okno dialogowe Edit Local Printer Entry



Powyzsze okno dialogowe pozwala zmieniac nazwe drukarki (domyslnie *lp*), sciezke dostepu do katalogu buforowania, maksymalne rozmiary plików, urządzenie drukarki i filtr wejsciowy. Filtr wejsciowy (*Input Filter*) moze posluzyc do skonfigurowania sterownika drukarki. Aby wejsc do okna dialogowego konfiguracji filtra (patrz rysunek 14.5), kliknij przycisk *Select* (Wybierz) obok opcji *Input Filter*.

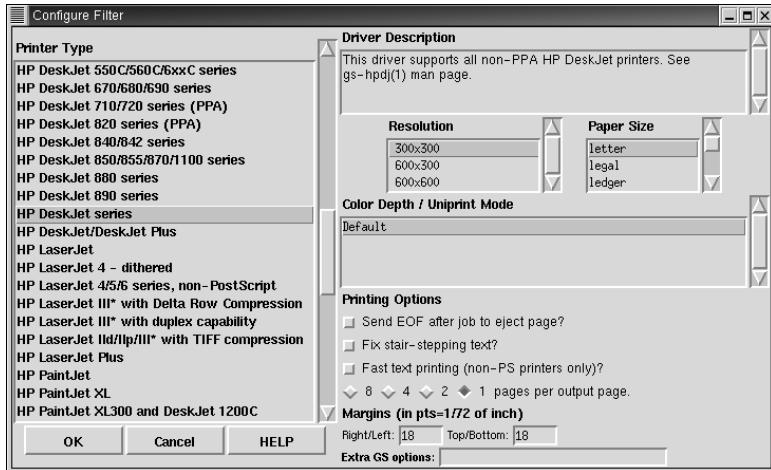
To okno dialogowe pozwala ustawić rozdzielczosc, marginesy, rozmiar papieru i liczbę stron na drukowanej stronie dla określonego dokumentu. Rysunek 14.5 pokazuje, jak dostosować ustawienia drukarki do własnych potrzeb. Po wybraniu modelu drukarki i ustawien kliknij *OK*.

Po dokonaniu wymaganych zmian, w polu *Input Filter* w oknie dialogowym *Edit Local Printer* pojawiła się nazwa drukarki, której instalację wybraliśmy. Kliknij *OK*, by zakończyć instalację drukarki. Ustawienia wybrane za pomocą narzędzia printtool sa au-

tomatycznie wprowadzane do pliku */etc/printcap*. Okno z rysunku 14.6 pojawia się po naciśnięciu przycisku *OK*. Proszę zwrócić uwagę na wpis dla drukarki, która dodaliśmy.

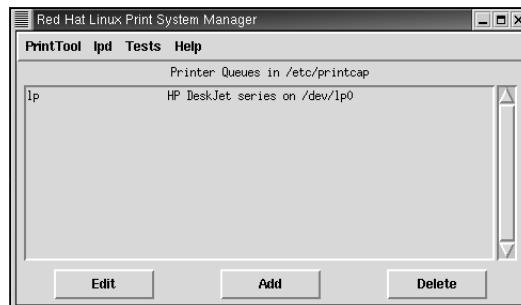
Rysunek 14.5.

Okno dialogowe
Configure Filter



Rysunek 14.6.

Okno dialogowe
Red Hat Linux Print System Manager

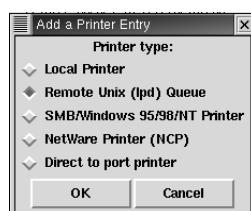


Laczenie ze zdalna drukarka

Pracując w sieci linuksowej, możemy chcieć skonfigurować drukarkę przy linuksowym serwerze. Aby można było z niej drukować pliki ze zdalnych klientów, niezbędne jest ustawienie kilku parametrów. Ten typ drukarki możemy skonfigurować za pomocą narzędzia printtool, przedstawionego wcześniej na rysunku 14.1. Zaczniemy od wyboru *Remote Unix (lpd) Queue* (zdalna kolejka uniksowa lpd) w oknie dialogowym *Add a Printer Entry*, jak na rysunku 14.7, a następnie kliknij *OK*.

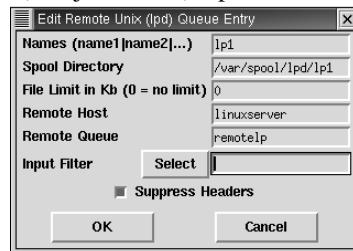
Rysunek 14.7.

Okno dialogowe
Add a Printer Entry

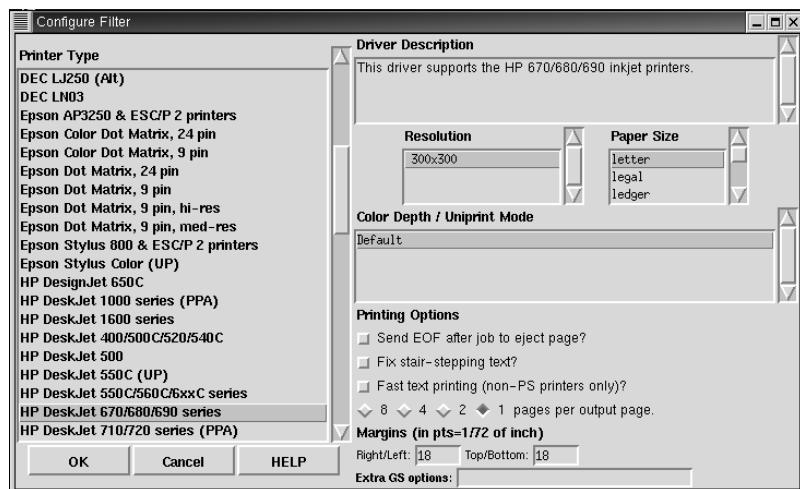


W oknie dialogowym *Edit Remote Unix (lpd) Queue Entry* (Edytuj wpis dla zdalnej kolejki uniksowej), przedstawionym na rysunku 14.8, musimy podać informacje dotyczące

ce zdalnego hosta, zdalnej kolejki i filtra wejsciowego. W kolumnie *Remote Host* (host zdalny) wpisz nazwe zdalnego serwera linuksowego, do którego jest przylaczona drukarka. W kolumnie *Remote Queue* (kolejka zdalna) wpisz nazwe zdalnej kolejki.

Rysunek 14.8.*Okno dialogowe**Edit Remote Unix (lpd)**Queue Entry*

Kliknij przycisk *Select*, aby otworzyc okno dialogowe *Configure Filter* (konfiguruj filtr), jak na rysunku 14.9.

Rysunek 14.9.*Okno dialogowe**Configure Filter*

W tym oknie dialogowym mozemy wybrac model drukarki przylaczonej do zdalnego serwera linuksowego lub uniksowego. Mozna tu również dostosowac ustawienia drukowania. Po dokonaniu niezbednych zmian kliknij *OK*. Narzedzie printtool automatycznie doda niezbedne informacje do pliku */etc/printcap*.

Aby pozwolic uzytkownikom drukowac dokumenty na zdalnej drukarce w systemie Unix lub Linux, musimy utworzyc w katalogu */etc* zdalnego komputera macierzystego plik o nazwie *hosts.lpd*. W tym pliku nalezy wpisac liste nazw hostów lub adresów IP uzytkowników, którym bedzie wolno zadac zadan drukowania od serwera. Zawartosc pliku *hosts.lpd* powinna byc nastepujaca:

```
# vi hosts.lpd
john.home.org
172.17.55.135
172.17.55.10
renne.home.org
steve.home.org
172.17.55.255
```

Jak widzimy, plik zawiera zarówno nazwy hostów, jak i adresy IP komputerów klienckich. Jedynie użytkownicy, których adresy IP lub nazwy hostów zostały wyszczególnione w tym pliku, sa w stanie drukować pliki na zdalnej drukarce. Dodatkowe informacje o wpisach dla zdalnych drukarek można znaleźć na stronach podręcznika dla polecenia `printcap` (`man printcap`). Po skonfigurowaniu drukarki, w głównym oknie dialogowym narzędzia `printtool` pojawi się wpis dla zdalnej drukarki.



Nazwy hostów możemy podawać w pliku `hosts.pld` tylko pod warunkiem skonfigurowania DNS-u. W przeciwnym razie musimy wpisywać adresy IP komputerów.

Polecenia związane z drukowaniem

Do drukowania wybranych plików można używać polecen, dostępnych w interfejsie wiersza poleceń. Najczęściej używane polecenia przedstawione są w tabeli 14.3.

Tabela 14.3. Polecenia związane z drukowaniem

Polecenie	Opis
<code>lpr</code>	Zgłasza zadanie drukowania.
<code>lpq</code>	Sprawdza kolejkę do drukowania.
<code>lprm</code>	Usuwa zadanie z kolejki.
<code>lpc</code>	Steruje funkcjonowaniem systemu drukarek wierszowych.
<code>lpstat</code>	Wyswietla obecny status usługi drukowania wierszowego.

Przyjrzymy się teraz szczegółowo wszystkim poleceniom wymienionym w tabeli 14.3. Omówimy również opcje najczęściej używane z tymi poleceniami.

Polecenie lpr

Polecenie `lpr` służy do wysyłania zadań drukowania do drukarki po jej pomyslnym skonfigurowaniu. Składnia tego polecenia jest następująca:

`lpr [opcje] nazwa`

Z poleceniem `lpr` można użyć kilku opcji. Najczęściej używane przedstawia tabela 14.4.

Tabela 14.4. Opcje polecenia lpr

Opcja	Funkcja
<code>-t lub -T</code>	Służy do przydzielenia tytułu do zadania drukowania.
<code>-d</code>	Określa miejsce przeznaczenia zadania drukowania.
<code>-P</code>	Służy do wyszczególnienia nazwy drukarki, do której chcemy wysłać zadanie.
<code>-o</code>	Służy do wyszczególnienia dodatkowych opcji.
<code>-n</code>	Ustala liczbę kopii.

Polecenie lpq

Polecenie `lpq` sluzy do sprawdzania stanu kolejki drukowania i przydaje sie np. do kontroli statusu wlasnie wyslanego zadania. Skladnia polecenia jest nastepujaca:

```
lpq [opcje]
```

W poleceniu `lpq` mozemy zastosowac kilka opcji. Tabela 14.5 wymienia kilka najczesciej uzywanych.

Tabela 14.5. Opcje polecenia `lpq`

Opcja	Funkcja
<code>-a</code>	Sluzy do listowania zadan drukowania wyslanych do wszystkich drukarek skonfigurowanych w danym komputerze.
<code>-P [nazwa_drukarki]</code>	Wyswietla zadania drukowania wyslane do podanej drukarki.
<code>-V</code>	Podaje informacje o wersji programu drukujacego.
<code>-s</code>	Sluzy do wyswietlenia pojedynczego wiersza informacji o kazdej kolejce.

Polecenie lprm

To polecenie sluzy do anulowania zadania i usuniecia go z kolejki. Skladnia polecenia `lprm` jest nastepujaca:

```
lprm [opcje]
```



Zwykly uzytkownik, uzywajacy drukarki sieciowej, moze za pomoca polecenia `lprm` odwolac jedynie wlasne zadania drukowania. Aby mowc usuwac zadania innych uzytkownikow, trzeba miec uprawnienia administratora systemu.

Z poleceniem `lprm` mozemy uzyc kilku opcji, ktore spowoduja usuniecie określonych zadan z kolejki. Polecenie `lprm` bez zadnych opcji usuwa ostatnie wyslane przez uzytkownika zadanie drukowania. Tabela 14.6 wymienia kilka opcji polecenia `lprm`.

Tabela 14.6. Opcje polecenia `lprm`

Opcja	Funkcja
<code>-a</code>	Usuwa wszystkie zadania z wszystkich kolejek drukowania.
<code>-P [nazwa_drukarki]</code>	Usuwa zadanie z kolejki podanej drukarki.
<code>-U [nazwa_uzytkownika]</code>	Usuwa zadanie określonego uzytkownika. Aby mowc skorzystac z tego polecenia, musimy posiadac odpowiednie uprawnienia lub uzywac konta administracyjnego root.

Polecenie lpc

Polecenie `lpc` sluzy do sterowania systemem drukarek. Skladnia tego polecenia jest nastepujaca:

```
lpc [opcje]
```

Tylko użytkownik root może używać tego polecenia. Ma ono duże możliwości i pozwala administratorowi systemu:

- ♦ wyłączać określona drukarkę lub wszystkie zainstalowane drukarki,
- ♦ załączając określona drukarkę lub wszystkie zainstalowane drukarki,
- ♦ przenosić zadania drukowania na początek kolejki,
- ♦ przenosić zadania drukowania z jednej drukarki do innej,
- ♦ zatrzymać chwilowo i wznowić dowolne zadanie drukowania,
- ♦ ponowić zadanie.

Kilka opcji polecenia `lp` zawiera tabela 14.7.

Tabela 14.7. Opcje polecenia `lp`

Opcja	Funkcja
<code>-P [nazwa_drukarki]</code>	Podaje kolejkę (bufor) drukarki, na której będą wykonywane operacje.
<code>-V</code>	Podaje informacje o wersji programu drukującego.
<code>-U [nazwa_użytkownika]</code>	Podaje nazwę użytkownika, którego zadanie dotyczy.

Polecenie `lpstat`

To polecenie służy do wyświetlenia stanu usługi drukowania. Polecenie `lpstat` bez opcji wyświetla bieżący stan wszystkich zadań drukowania, wysłanych do domyślnej drukarki. Składnia tego polecenia jest następująca:

```
lpstat [opcje]
```

Tabela 14.8 wymienia opcje, których można uzyskać z poleceniem `lpstat`.

Tabela 14.8. Opcje polecenia `lpstat`

Opcja	Funkcja
<code>-a</code>	Opcja podana z nazwami drukarek służy do kontroli, czy urządzenia przyjmuje zadania drukowania, czy nie.
<code>-d</code>	Służy do wyświetlenia domyślnego miejsca przeznaczenia zadań drukowania.
<code>-P</code>	Z następującymi po niej nazwami drukarek służy do wyświetlenia ich statusu.
<code>-t</code>	Wyswietla wszystkie bieżące informacje o stanie usługi drukowania.



Dodatkowe informacje o zdalnym drukowaniu za pomocą `lpr` i `lpd` zawiera RFC 1179.

Internet Printing Protocol Microsoftu

Protokół IPP (*Internet Printing Protocol* — internetowy protokół drukowania) został zaprojektowany przez Microsoft dla własnych systemów operacyjnych. Pozwala on drukować przez Internet lub intranet bezpośrednio do adresu URL (*Uniform Resource Locator*). Dodatkowa zaleta protokołu IPP jest możliwością instalowania drukarek z Internetu lub intranetu za pomocą Internet Explorera.



Podczas instalowania drukarki za pomocą IPP, jeśli nie są dostępne wymagane sterowniki dla drukarki, może pojawić się komunikat o błędzie.

Poniższe punkty opisują procedury konfiguracji IPP. Jak widać, procedury stosowane przez administratorów i innych użytkowników różnią się od siebie, ponieważ administrator konfiguruje serwer IPP, zaś użytkownicy jedynie własne komputery do korzystania z serwera IPP. Administrator instaluje sterowniki dla drukarki.

Administratorzy

Przed skorzystaniem z IPP należy zainstallować sterowniki dla drukarek, które będą instalowane. Procedura wygląda następująco:

1. Wybierz *Start* → *Ustawienia* → *Drukarki*, aby otworzyć okno *Drukarki*.
2. Kliknij dwukrotnie *Dodaj drukarke*, aby uruchomić *Kreatora dodawania drukarki*, a następnie kliknij *Dalej*.
3. Wybierz *Drukarka lokalna* (jeśli ustawienie to nie zostało domyślnie zaznaczone).
4. Wybierz *Utwórz nowy port* i upewnij się, czy w polu *Typ* został wybrany *Local port*. Kliknij *Dalej*. Pojawi się teraz okno dialogowe *Nazwa portu*.
5. W polu tekstowym *Wprowadź nazwę portu* wpisz nazwę udziału (np. `\serwerdrukowania\nazwaudzialu`).
6. Przejdz pozostałą część *Kreatora*, instalując niezbędne sterowniki urządzenia.

Pozostali użytkownicy

1. Wybierz *Start* → *Ustawienia* → *Drukarki*, aby otworzyć okno *Drukarki*.
2. Kliknij dwukrotnie *Dodaj drukarke*, aby uruchomić *Kreatora dodawania drukarki*, a następnie kliknij *Dalej*.
3. Wybierz *Drukarka lokalna* (jeśli ustawienie to nie zostało domyślnie zaznaczone).
4. Wybierz *Utwórz nowy port* i upewnij się, czy w polu *Typ* został wybrany *Standard port monitor*. Kliknij *Dalej*. Pojawi się teraz okno dialogowe *Nazwa portu*.
5. Wpisz adres IP serwera drukowania IPP.
6. Przejdz pozostałą część *Kreatora*, instalując niezbędne sterowniki urządzenia.

Rozdział 15.

Aplikacje

i protokoly WWW

W tym rozdziale:

- ◆ Wprowadzenie do WWW
- ◆ Aplikacje oparte na WWW
- ◆ Wideo i inne współczesne typy danych

Pojecie *infostrady* (*information superhighway*) — które odnosi się do globalnej sieci telekomunikacyjnej i technologii używanych w handlu, edukacji, rozrywce i tak dalej — zrewolucjonizowało sposób, w jaki ludzie komunikują się ze sobą. Szkieletem tej sieci globalnej jest Internet i WWW.

Niniejszy rozdział omawia strukturę i funkcje WWW, role World Wide Web Consortium (W3C) w rozwoju standardów dla Sieci, język HTML, służący do tworzenia dokumentów w Sieci, oraz protokół HTTP. Przedstawione zostały w nim również różne aplikacje związane z WWW.

Podstawy WWW

Czym jest Internet? Czym jest WWW? Czy tych dwóch pojęć możemy używać zamiennie? Jak powstały te technologie? Bieżący podrozdział odpowiada na powyższe pytania.

Internet — wprowadzenie

Internet jest zbiorem komputerów, połączonych ze sobą w celu wspólnego korzystania z informacji. Internet nie jest pojedyncza sieć, lecz raczej sieć złożona z innych sieci, które używają do komunikacji wspólnego protokołu TCP/IP.

Internet powstał z sieci ARPANET, utworzonej przez agencję rządową Stanów Zjednoczonych DARPA (Defense Advanced Research Project Agency) w 1969 roku. Sieć ARPANET stanowiła odporny na uszkodzenia komputerowy system łączności, który był w stanie przetrwać utratę jednego lub kilku centrów komputerowych, na przykład

baz wojskowych lub miast. Na poczatku ta siec skladala sie z czterech komputerow glownych, uzywajacych do komunikacji protokolu NCP (*Network Control Protocol*). Technologia stosowana w lacznosci pomiedzy nimi nosila nazwe *komutacji pakietow* (*packet switching*). Protokol NCP nie byl jednak w stanie obsluzyc rosnacego stale ruchu sieciowego, wobec czego w roku 1974 zostaly zaproponowane i zaimplementowane TCP (*Transmission Control Protocol*) oraz IP (*Internet Protocol*) — bardziej solidne protokoly komunikacyjne.



Proces komutacji pakietow polega na podziale komunikatow z danymi na male pakiety. Pakiet przypomina list — zawiera czeesc komunikatu i adres odbiorcy. Kazdy pakiet jest przesyłany przez siec indywidualnie. Po osiągnieciu przez wszystkie pakiety miejsca przeznaczenia, sa one reorganizowane z powrotem w kompletny komunikat. Na technologii komutacji pakietow opiera sie protokol TCP/IP.

W latach 80. naukowcy i organizacje zdaly sobie sprawe z korzysci, jakie przynosi siec ARPANET. Na skutek tego siec rozrosla sie, obejmujac sieci uniwersytetow, korporacji i spolecnosci uzytkownikow. Od roku 1982 siec ARPANET jest znana powszechnie pod nazwa *Internet*. Mozliwosci i zasieg Internetu spowodowaly, ze w ciagu zaledwie kilku lat nastapil niewiarygodny wrecz rozwój Sieci. Różnorodne uslugi — Gopher, WAIS (*Wide Area Information Server*) i WWW — zostaly opracowane w jednym celu: aby pomóc uzytkownikom korzystac z danych w Internecie.



Gopher (dosl. susel) jest programem opracowanym w 1991 r., który dzieli informacje na logiczne kategorie i organizuje je w hierarchiczna strukture drzewa. WAIS jest programem wyszukujacym dokumenty w Internecie.

Korzystanie z uslug Gopher i WAIS skonczylo sie wraz z rozwojem WWW. Wielkosć baz danych Gophera jest przekształcana na strony WWW, łatwo dostepne za pomocą wyszukiwarek internetowych.

Ewolucja WWW

Najpopularniejsza metoda udostepniania informacji w Internecie jest format znany pod nazwa WWW (*World Wide Web* — ogólnoswiatowa pajeczyna lub po prostu *Web*). WWW sklada sie z plików zwanych stronami WWW, które zawieraja informacje i lacza do innych stron WWW. Przed pojawiением sie tej uslugi, dane byly przesyłane jako tekst lub w postaci kodu binarnego. WWW dodaje zdolosc do zawarcia tekstu, grafiki, dzwieku i animacji w pojedynczym pliku.

Standard WWW został opracowany w 1991 roku w instytucie CERN (European Center for Nuclear Research). Fizycy w CERN potrzebowali szybkiego mechanizmu udostepniania swoich informacji badawczych innym naukowcom na calym swiecie. Jeden z tych fizykow, Tim Berners-Lee, zaproponował tekstowy system hipertekstu majacy sluzyc do wymiany danych pomiedzy fizykami zaangazowanymi w badania w dziedzini fizyki wysokich energii. W uzytej przez niego technice hipertekstowej wskazanie na wyróżnione slowo lub fraze kierowalo uzytkownika do nowej strony w tym samym komputerze lub do zdalnego komputera w sieci.

W roku 1993 w National Center for Supercomputer Applications (NCSA) opracowano przyjazny dla uzytkownika program, który udostepniał graficzny interfejs WWW. Program ten, noszacy nazwe *Mosaic*, był pierwsza internetowa przeglądarka graficzna, która

pozwalała użytkownikom pobierać informacje przez proste wskazanie na lączce i kliknięcie. Spowodowało to szybki rozwój WWW. W chwili obecnej dostępnych jest wiele przeglądarek WWW opartych na Mosaic; zaliczają się do nich popularne Netscape Navigator i Internet Explorer.



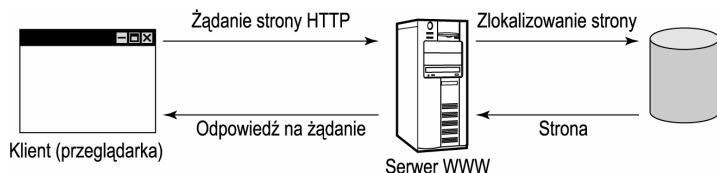
Nazwa *World Wide Web* odnosi się do zbioru informacji w Internecie, które charakteryzuje kolorowa grafika i lączce hipertekstowe. *Przeglądarka (browser)* to narzędzie, które pozwala użytkownikowi przeglądać informacje zawierające grafiki i lączce.

Jak funkcjonuje WWW

WWW opiera się na modelu klient-serwer. W tym modelu program kliencki wysyła zadanie do programu serwera, który zwykle funkcjonuje w zdalnym komputerze. Klient i serwer komunikują się ze sobą przez Sieć. Serwer po otrzymaniu zadania od klienta nawiązuje z nim połaczenie, przetwarza zadanie, wysyła wyniki do klienta i zamyka połaczenie. W przypadku WWW zadania do serwera zgłasza przeglądarki WWW, na przykład Internet Explorer lub Netscape Navigator. Kazdy komputer, który przechowuje strony WWW zawierające informacje zadane przez klienta, może grać role serwera. Strony WWW są pisane w języku HTML (*HyperText Markup Language* — język hipertekstowego znakowania informacji). HTML daje przeglądarkę instrukcje, jak należy wyświetlić stronę. Klient i serwer komunikują się ze sobą za pomocą protokołu warstwy aplikacji, noszącego nazwę HTTP (*HyperText Transfer Protocol* — protokół przesyłania hipertekstu).

Użytkownik, który chce skorzystać z informacji w Sieci, podaje w przeglądarce adres URL — *Uniform Resource Locator* (jednolity lokalizator zasobów). Adres URL jest unikalnym identyfikatorem, który definiuje trasę do pliku w komputerze przyłączonym do Internetu. URL może być również osadzony w dokumencie i przedstawiony użytkownikowi jako lączce hipertekstowe. Funkcjonowanie WWW przedstawia rysunek 15.1.

Rysunek 15.1.
Działanie
World Wide Web



Format URL wygląda następująco:

<Identyfikator protokołu>://<nazwa serwera>[:<port>]
Σ[/<ściezka do dokumentu HTML>] <nazwa pliku HTML>]

Na przykład:

<http://www.helion.pl/katalog.htm>

http oznacza protokół używany do komunikacji pomiędzy klientem i serwerem. Możemy podać jeden z następujących protokołów:

- ◆ http — protokół dostępu do WWW,
- ◆ ftp — do przesyłania plików,

- ♦ *wais* — dostep do *Wide Area Information Server*,
- ♦ *mailto* — dostep do poczty elektronicznej,
- ♦ *gopher* — dostep do serwera Gopher,
- ♦ *file* — dostep do pliku w systemie lokalnym.

Po protokole następuje adres serwera WWW (w naszym przykładzie *www.webknowledgebase.com*). Sufiks *.com* wskazuje na organizację komercyjną. Pozostałe przyrostki to:

- ♦ *edu* — instytucja edukacyjna,
- ♦ *gov* — agencja rządowa,
- ♦ *org* — organizacja niekomercyjna,
- ♦ *mil* — organizacja wojskowa,
- ♦ *net* — organizacja sieciowa.

W przykładowym URL numer portu nie został podany, wobec tego zostanie użyty domyślny dla HTTP port 80. *definition* jest folderem w serwerze WWW, w którym zadana strona jest przechowywana, zas *ARPANET.html* jest zadana strona.

Gdy użytkownik wpisze URL w polu przeglądarki lub kliknie lączne hiperlinki, przeglądarka wysyła do wyszczególnionego serwera WWW zadanie strony. Serwer WWW pobiera zadana stronę i wysyła ją do przeglądarki, która stronę odczytuje, interpretując zawarte w niej instrukcje i wyświetla wynik.

HTML

Jezyk HTML (*HyperText Markup Language* — język hiperlinkowego znakowania informacji), jak nazwa wskazuje, jest *językiem znakowania (adnotacyjnym)*, używanym do tworzenia stron WWW. Język znakowania używa zbioru etykiet, zwanych *znacznikami (tag)*, osadzonych w tekscie. Znaczniki są niewidoczne dla czytelnika i nie stanowią elementu składowego zawartości dokumentu, lecz wzbogacają dokument, definiując jego strukturę i sposób wyświetlania.

HTML pochodzi od języka SGML (*Standard Generalized Markup Language* — standardowy uniwersalny język znakowania), lecz jest o wiele łatwiejszy w użytkowaniu. HTML stanowi standard de facto sposobu, w jaki informacje są organizowane i wyświetlane. Pozwala więc różnym producentom tworzyć różne przeglądarki dla różnych platform programowych i sprzętowych, wyświetlające dane w sposób zblizony.



Jezyk SGML, wprowadzony w 1986 r., był pierwszym opracowanym językiem znakowania, który dostarczał i wyświetlał dokumenty niezależnie od używanej platformy. Ponieważ SGML jest językiem rozbudowanym i trudnym do naużenia, Tim Berners-Lee opracował i zdefiniował (w roku 1990) język HTML, jako język znakowania służący do tworzenia stron WWW.

Wersje języka HTML

Od momentu powstania, HTML przeszedł szereg zmian. Istnieją następujące wersje tego języka:

- ◆ *HTML 2.0* — pierwsza ustalona wersja HTML, która zawierała większość używanych obecnie znaczników, lecz nie obsługiwała tabel i możliwości justowania tekstu.
- ◆ *HTML 3.2* — wersja, do której zostały wprowadzone tabele, grafika i atrybuty justowania.
- ◆ *HTML 4.0* — zawiera pewne rozszerzenia firm Microsoft i Netscape, na przykład ramki (FRAME) — wprowadzone przez Microsoft rozszerzenie, pozwalające podzielić stronę WWW na dwie lub więcej części.
- ◆ *HTML 4.01* — obecnie obowiązujący standard, który zawiera szereg rozszerzeń własnych i inne funkcje, jak np. formularze i arkusze stylów. HTML 4.01 obsługuje również lokalizację.



Internakcjalizacja oznacza proces tworzenia aplikacji w sposób, który pozwala na jej łatwą adaptację do różnych języków i regionów bez wprowadzania zmian. Internakcjalizacja ma kilka zalet, do których należą konsekwentny wygląd oraz obsługa wyświetlanego i wprowadzania znaków w różnych językach.

Struktura dokumentu HTML

W języku HTML dokument podzielony jest na logiczne bloki zwane *elementami*. Elementy te określają wygląd osadzonego w nich tekstu. Na przykład, HTML używa elementu ANCHOR (zakotwiczenie), który pozwala wyświetlać tekst jako hipertekst. Elementy reprezentowane w dokumencie w postaci znaczników stanowią bloki konstrukcyjne dokumentu HTML. Na przykład, element ANCHOR jest reprezentowany w dokumencie przez znacznik <A>. Ważny jest fakt, iż znaczniki zwykle występują w parach, oznaczając początek i koniec elementu. Weźmy pod uwagę taki przykład:

<A>Kliknij tutaj

W tym przykładzie początek i koniec elementu ANCHOR są reprezentowane przez pary znaczników <A> i . Tekst pomiędzy tymi znacznikami będzie wyświetlony jako lączce hipertekstowe.

Dokument HTML jest generalnie podzielony na dwie części: nagłówek (HEAD) i część główna (BODY). Sekcja HEAD zawiera informacje o samym dokumencie, na przykład tytuł przeznaczony do wyświetlenia w pasku tytułowym okna przeglądarki lub słowa kluczowe zawarte w dokumencie. Sekcja BODY zawiera sam tekst, wyświetlany w oknie przeglądarki. Spójrzmy na następujący dokument HTML:

```
<HTML>
  <HEAD>
    <TITLE>MOJA PIERWSZA STRONA WWW</TITLE>
  </HEAD>
  <BODY>
    <FONT COLOR="RED">
      <H1 ALIGN="CENTER">HTML - wprowadzenie</H1>
    </FONT>
    <FONT SIZE=4>
      <P>
        Jezyk HTML, jak nazwa wskazuje, jest jezykiem znakowania
        (adiustacyjnym), uzywanym
        do tworzenia stron WWW. Jezyk znakowania uzywa zbioru etykiet, zwanych
        znacznikami, osadzonymi w tekscie. Znaczniki sa niewidoczne dla
```

```

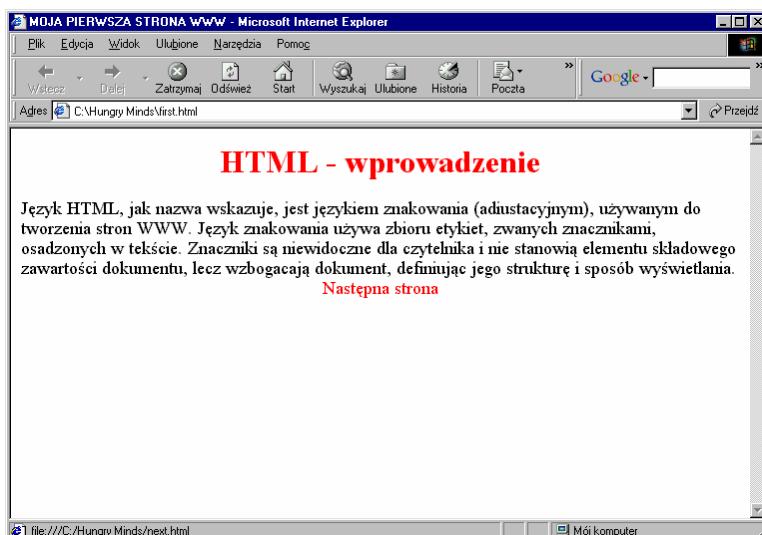
czytelnika i nie stanowią elementu składowego zawartości dokumentu, lecz
wzbogacają dokument, definiując jego strukturę i sposób
wyświetlania.<BR>
<CENTER><A HREF="next.html">Następna strona</A></CENTER>
</P>
</FONT>
</HTML>

```

Po otwarciu powyższego pliku w programie Internet Explorer przeglądarka zinterpretuje znaczniki i wyświetli tekst, jak na rysunku 15.2.

Rysunek 15.2.

Przykładowa strona HTML



Jak widać, znaczniki są odczytywane i interpretowane przez przeglądarkę, lecz nie pojawiają się na ekranie. Przeglądarka, napotkawszy znacznik `<P>`, interpretuje go jako początek akapitu. Podobnie, napotkawszy znacznik ``, zmienia rozmiary i kolor tekstu, zgodnie z parametrami podanymi w znaczniku. Kształt kurSORA zmienia się po naprowadzeniu na tekst *Następna strona*, który jest lączem hipertekstowym. Jeśli klikniemy lączę, przeglądarka wysła zadanie innej strony (w tym przykładzie *next.html*). Plik zostanie zinterpretowany i wyświetlony w podobny sposób przez różne przeglądarki, niezależnie od używanego sprzętu i platformy oprogramowania.

HTTP

HTTP (*HyperText Transfer Protocol* — protokół przesyłania hipertekstu) jest protokołem warstwy aplikacji, służącym do komunikacji pomiędzy serwerem WWW i przeglądarkami. Określa, jak komunikaty mają być formatowane i przesyłane oraz jakie czynności powinien podjąć serwer WWW (lub przeglądarki) w reakcji na określone polecenia. HTTP jest protokołem bezstanowym, co oznacza, że nie zachowuje informacji o poleceniach, które poprzedzały aktualne polecenie.

HTTP umożliwia klientowi wysłanie do serwera WWW listy wszystkich symboli, które potrafi zinterpretować. Na podstawie tych informacji serwer odpowiada w sposób

optymalny dla klienta. Pozwala to klientom i serwerom uporac sie z licznymi formatami graficznymi, jak np. *GIF* lub *JPEG*.

Protokół HTTP uzywa adresu URL wprowadzonego w przeglądarce przez użytkownika, aby znalezc zasób w Internecie. Komunikacja pomiedzy przeglądarka i serwerem WWW odbywa sie za pomoca różnych polecen HTTP, do których naleza:

- ◆ GET — kaze serwerowi pobrac dane polozone w miejscu określonym przez URL.
- ◆ HEAD — polecenie podobne do GET, lecz w tym przypadku serwer zwraca do przeglądarki jedynie nagłówek dokumentu, bez czesci głównej.
- ◆ POST — ta instrukcja zleca serwerowi WWW utworzenie nowego obiektu. Pole identyfikacji komunikatu w nowym obiekcie moze zostac wypełnione zarówno przez serwer WWW, jak i przez przeglądarkę. Serwer przydziela do tego obiektu nowy URL, który nastepnie wysyla do klienta. Nowy obiekt staje sie czescia zazadanego dokumentu.

Wszystkie transakcje HTTP odbywaja sie za pomoca protokolu TCP/IP. Transakcja HTTP sklada sie z nastepujacych faz:

- ◆ *Polaczenie* — w tej fazie przeglądarka usiluje polaczyc sie z serwerem WWW. Ten z kolei odbiera zadanie i nawiiazuje polaczenie.
- ◆ *Zadanie* — po utworzeniu polaczenia klient wysyla do serwera zadanie, określając protokół komunikacyjny i plik, który ma zostać pobrany i wysłany. Klient wysyla ponadto informacje o typach danych (*GIF*, *TIFF*, *JPEG* itp.), które jest w stanie obsluzyc.
- ◆ *Odpowiedz* — jeśli serwer znajdzie i przetworzy informacje zadane przez klienta, wysyla do niego odpowiedz. Jeśli nie może znalezc informacji, to zwraca komunikat o błędzie. Serwer wysyla odpowiedzi zależne od typów danych, obsługiwanych przez klienta. Na przykład, jeśli chcemy otworzyć stronę zawierającą pliki w formacie Flash, a nasza przeglądarka nie obsługuje tego formatu, zostanie wyświetlone okno dialogowe z zapytaniem, czy chcemy zainstalować dodatkowe składniki, niezbędne do wyświetlania tych plików.
- ◆ *Zamkniecie* — polaczenie pomiedzy klientem i serwerem może zostać zamknięte przez każdą ze stron.

World Wide Web Consortium

Organizacja World Wide Web Consortium (w skrócie W3C) została założona przez Tima Bernersa-Lee w roku 1994, aby dopomóc naukowcom i uczelniam z całego świata w pełni wykorzystać potencjal WWW. W3C tworzy standardy i protokły, wspierające rozwój WWW, oraz zapewnia współpracę między różnymi produktami związany z WWW. Organizacja W3C rozwinęła się z garski osób w dużej konsorcjum, zrzeszające dzisiaj około 500 organizacji członkowskich.

Do długofalowych zadań W3C, wymienionych w deklaracji misji organizacji, naleza:

- ◆ *Powszechny dostęp* — udostępnienie Sieci wszystkim ludziom, niezależnie od kraju zamieszkania, kultury, wykształcenia, zdolności, ograniczeń i zasobów fizycznych.

- ♦ *Siec zaufania* — stworzenie kultury zaufania w obszarach handlu, technologii i zagadnień społecznych.
- ♦ *Semantyka Sieci* — opracowanie środowiska programowego, które pozwoli wszystkim użytkownikom w pełni wykorzystać zasoby dostępne w Sieci.

W ciągu ubiegłych pieciu lat organizacja W3C opracowała ponad 20 specyfikacji technicznych dla infrastruktury Sieci. Specyfikacje te, podobnie jak cała filozofia W3C, opierają się na trzech regulach projektowych:

- ♦ *Współoperatywność* — specyfikacje języków, protokołów i produktów używanych w Sieci muszą być ze sobą wzajemnie zgodne, w wyniku czego produkty różnych firm powinny być proste w użyciu i implementacji.
- ♦ *Ewolucja* — przyszłe technologie muszą być łatwo dostosowywane do istniejącej infrastruktury Sieci. Prostota, modułowość i rozszerzalność istniejących technologii mają wpływ na łatwość, z jaką technologie przyszłości będą przyjmowane.
- ♦ *Decentralizacja* — rozłożenie Sieci na całym świecie, przy równoczesnym ograniczaniu zależności od centralnych urzędów.

Konsorcjum W3C, pozostając wierne swojej misji i celom, stworzyło zalecenia, które stały się elementami konstrukcyjnymi rosnącej wydajności Sieci, a co za tym idzie, jej popularności. Jak dotąd, do najbardziej udanych zaleceń W3C zaliczają się:

- ♦ Język HTML, służący do tworzenia stron WWW i udostępniania informacji w Sieci. W chwili obecnej tworzony jest rygorystycznie język XHTML (*eXtensible HyperText Markup Language* — rozszerzalny HTML), pozwalający na tworzenie bardziej interaktywnych i dynamicznych stron.
- ♦ CSS (*Cascading Style Sheets* — kaskadowe arkusze stylów), nadające informacjom w Sieci styl i kolory.
- ♦ DOM (*Document Object Model* — model obiektów dokumentów), dający dostęp do struktury dokumentu, stylów, zdarzeń i tak dalej. DOM zawiera obiekty reprezentujące różne składniki strony WWW: akapity, pozycje listy, obrazy itp. Za pomocą DOM można uzyskać dostęp do różnych części strony WWW i zmienić ich wygląd. Dzięki temu DOM daje twórcom serwisów WWW większą kontrolę nad dokumentami i ich wyglądem.
- ♦ Dynamiczny HTML (DHTML), który stanowi połączenie innych technologii, w tym HTML, CSS i DOM. DHTML pozwala zmienić wygląd składnika strony WWW po stronie klienta — na przykład, możemy utworzyć dynamiczną stronę WWW, która wyświetla listę zabawek dostępnych na stronie WWW. Gdy użytkownik wskazuje na zabawkę, możemy za pomocą DHTML wyświetlić jej obraz na stronie WWW.
- ♦ XML (*eXtensible Markup Language*), pozwalający społeczności internetowej projektować języki znakowania spełniające określone wymagania.

W prawdziwe konsorcjum W3C do pewnego stopnia osiągnęło swoje cele — Sieć jest dostępna, zdecentralizowana i współoperatywna, lecz wiele jeszcze pozostaje do zrobienia. W porównaniu z innymi dziedzinami przemysłu, Internet i WWW rozwijają się w zjawiskowym tempie. Nowe technologie i produkty są tworzone tak szybko, że trudno

za nimi nadazyc, co wywoluje jeszcze wieksza presje na W3C i jego misje neutralnosci względem producentów, koordynacji i zgodnosci.

Aplikacje WWW

Różne spolecnosci uzytkowników korzystaja z WWW na różne sposoby. Na przyklad, regularni uzytkownicy wykorzystuja WWW, aby odbierac poczte elektroniczna i gawędzic z innymi uzytkownikami z calego swiata. Spolecznosc biznesowa uzywa WWW jako poteznego nosnika pozwalajacego reklamowac i sprzedawac produkty. W tym podrozdziale przedstawimy kilka najczesciej uzywanych serwerów WWW, pozwalajacych udostepniac aplikacje WWW. Przyjrzymy sie również różnym typom aplikacji internetowych.

Serwery WWW

Serwerem WWW moze byc dowolny komputer, przechowujacy strony WWW i zawierajacy oprogramowanie serwera WWW. Oprogramowanie to przyjmuje zadania klien-tów WWW, takich jak Internet Explorer czy Netscape, oraz zwraca do nich rezultaty. Do najczesciej uzywanych programów serwerów WWW naleza:

- ♦ *Internet Information Server (IIS)* — oprogramowanie serwera uslug Sieci firmy Microsoft, dzialajace pod systemami Windows NT Server i Windows 2000 Server. IIS jest bardzo efektywny i latwy do skonfigurowania. Oprócz obslugi protokolu HTTP 1.1, IIS oferuje dodatkowe narzedzia, na przyklad Microsoft Transaction Server (MTS), sluzacy do tworzenia rozproszonych aplikacji, Index Server, sluzacy do indeksowania i przeszukiwania stron WWW i dokumentów Microsoft Word, oraz Site Analyst, sluzacy do zarzadzania witryna i analizowania jej wykorzystania. IIS posiada szereg funkcji obejmujacych: ochronie w przypadku awarii, obsluge technologii ASP i Java, uruchamianie skryptów i zarzadzanie zawartoscia.



ASP (Active Server Pages) jest technologia WWW dzialajaca po stronie serwera. Dodatkowe informacje o ASP zawiera punkt „Języki” w dalszej czesci rozdziału.

- ♦ *Personal Web Server (PWS)* — PWS to również oprogramowanie serwera WWW firmy Microsoft, bedace okrojona wersja IIS. Nie zawiera niektórych zaawansowanych funkcji IIS, takich jak Index Server, lecz obsluguje ASP i uruchamianie skryptów. PWS dziala pod systemami operacyjnymi Windows 9x i Windows NT Workstation. Serwer ten umozliwia publikowanie prywatnych stron i malych witryn WWW, nadaje sie do tworzenia uslug intranetowych i zawiera kreatory dla takich zadan, jak tworzenie stron prywatnych i udostepnianie plików. Do uruchamiania i zatrzymywania PWS oraz udostepniania plików mozna wykorzystac interfejs Explorera lub narzedzia Personal Web Manager. PWS moze również sluzyc do testowania strony WWW w systemie Windows 9x lub Windows NT Workstation przed opublikowaniem w Internecie. Po sprawdzeniu poprawnosci laczy i skryptów mozemy nadal korzystac z PWS lub za pomoca Microsoft FrontPage skoprowac witryne WWW z serwera PWS do IIS. Zarówno

PWS, jak i IIS znajdują się w pakiecie Windows NT Option Pack, który można pobrać za darmo z Internetu pod adresem:

<http://www.microsoft.com/NTServer/nts/downloads/recommended/NT4OptPk/default.asp>



Pakiet Microsoft FrontPage pozwala z łatwością budować kompletnie witryny WWW. W tym pakiecie wszystkie strony mają wspólny spójny i łatwy do modyfikacji wygląd. Pakiet zawiera również narzędzia do nawigacji, które tworzone są automatycznie podczas dodawania i usuwania stron. Edytor FrontPage jest typu WYSIWYG (wynik identyczny z tym, co widać) i udostępnia okna dialogowe służące do rozmieszczania obiektów (grafiki, plików tekstowych, reklam) na stronie WWW.

- ♦ **Apache** — niezawodne oprogramowanie serwera WWW, opracowane przez grupę dwudziestu programistów-ochotników pod nazwą Apache Group. Apache pochodzi od serwera WWW HTTPd, a ponieważ zawiera oryginalny kod HTTPd z dodatkowymi łatami, otrzymał nazwę *A Patchy Server* („Polatany serwer”), która zmieniona później na *Apache Server*. Apache został opracowany dla systemu operacyjnego Unix, lecz nowsze wersje serwera mogą również korzystać z platform OS/2 i Windows. Apache ma sporo zalet: po pierwsze, jest dostępny w Sieci za darmo. Ponadto, programiści mogą łatwo rozszerzać jego funkcjonalność, ponieważ mogą pobrać kod źródłowy Apache. Dzięki temu Apache jest dobrym rozwiązaniem dla przedsiębiorstw i indywidualnych użytkowników, którzy używają Uniksa lub kombinacji platform Unix i Windows NT.



HTTP daemon (HTTPd) jest oprogramowaniem serwera WWW, opracowanym przez National Center for Supercomputer Applications (NCSA), dostępnym dla różnych wersji systemu Unix. Dodatkowe informacje o HTTPd można znaleźć pod adresem www.ncsa.uiuc.edu.

- ♦ **Java Web Server** — oprogramowanie stworzone w technologii Java. Ten serwer jest najlepszą platformą do uruchamiania aplikacji i serwletów Java. Java Web Serwer jest rozwiązaniem rozszerzalnym, ponieważ każdy może napisać własny kod w języku Java i dodać go do serwera.

Oprócz wymienionych powyżej programów dostępne są inne serwery WWW, na przykład iPlanet Web Server (oprogramowanie firmy Netscape serwera klasy high-end dla dużych przedsiębiorstw, dla systemów Unix i Windows NT) oraz serwer WWW Lotus Domino, który umożliwia integrację z Lotus Notes i hosting stron WWW. Inaczej mówiąc, użytkownicy nie muszą instalować klienta Lotus Notes w swoim komputerze; mogą zamiast tego korzystać z aplikacji Lotus Notes za pomocą dowolnej przeglądarki WWW.

Aplikacje w Internecie

Internet jest skladnicą informacji, rosnącą z dnia na dzień. I wprawdzie pozwala w łatwy sposób korzystać z informacji, lecz czasem trudno jest znaleźć odpowiednie informacje w tak dużym zbiorze. Na szczęście różnorodne narzędzia zwane *aparatami wyszukiwania* (*search engine*), lub popularnie wyszukiwarek, umożliwiają szybkie i skuteczne znajdowanie informacji. Narzędzia te używają do szukania podanych słów bazy danych — zbioru informacji o dokumentach dostępnych w Sieci. Do najpopularniejszych wyszukiwarek należą AltaVista, Excite, WebCrawler, Yahoo i Google. Poza

wyszukiwaniem informacji w Internecie pewne aplikacje — na przykład Netscape Messenger i Microsoft Internet Mail — pozwalają wykonywać następujące czynności:

- ♦ wysyłać i odbierać poczty elektronicznej,
- ♦ uczestniczyć w grupach dyskusyjnych poswięconych różnorodnym tematom,
- ♦ tworzyć własne dokumenty i publikować je w Internecie,
- ♦ przesyłać i pobierać pliki,
- ♦ łączyć się z bibliotekami na całym świecie,
- ♦ przeprowadzać transakcje handlowe przez Internet.

Poczta elektroniczna i grupy dyskusyjne

Poczta elektroniczna (e-mail), która umożliwia użytkownikom wysyłanie i odbieranie wiadomości, jest najpowszechniej używana usługa w Internecie. Usługa ta jest popularna nawet w małych sieciach, nie połączonych z Internetem.

Forum poczty elektronicznej pozwala grupie osób uczestniczyć w dyskusji grupowej. Lista osób, do których wiadomości innych osób z listy są wysyłane, nosi nazwę *listy adresowej* lub *listy dyskusyjnej (mailing list)*. Dostępnych jest wiele programów poczty elektronicznej, pozwalających utworzyć alias dla listy adresowej. Jeśli chcemy uczestniczyć w formalnej dyskusji, możemy zapisać się do dostępnych publicznie list adresowych. Przykładami forum poczty elektronicznej mogą być Netscape Messenger i Microsoft Internet Mail.

Grupy dyskusyjne (newsgroups) pozwalają tysiącom użytkowników uczestniczyć w dyskusjach grupowych i dzielić się w Sieci swoimi poglądami. Grupa dyskusyjna przypomina komputerowa tablice ogłoszeń — każdy subskrybent może wysyłać i czytać wiadomości. Do programów obsługujących grupy dyskusyjne należą Collabra Discussions oraz Internet News Microsoftu.

Aplikacje służące do tworzenia stron WWW

Do tworzenia stron WWW służą język HTML. Możemy do tego celu używać edytora tekstu (na przykład vi w systemach uniwersalnych i Notatnik w systemie Microsoft Windows) lub dowolnych edytorów graficznych. Dostępne są również wyspecjalizowane edytory HTML-a. Narzędzia służące do tworzenia stron WWW możemy podzielić na następujące kategorie:

- ♦ Rozszerzenia edytorów i procesorów tekstu, pozwalające używać standardowego edytora do tworzenia i modyfikacji stron HTML.
- ♦ Autonomiczne edytory HTML-a typu WYSIWIG (*What You See Is What You Get* — wynik będzie identyczny z tym, co widzimy na ekranie), na przykład edytor FrontPage Microsoftu.
- ♦ Narzędzia do konwersji, pozwalające przetworzyć dokument w określonym formacie na HTML.

Aplikacje pomocnicze

Typowa przeglądarka WWW może wyświetlać obrazy graficzne i odtwarzac standar-dowe pliki dźwiękowe. Serwer WWW i przeglądarka w celu dopasowania formatu dan-nych przesyłanych pomiędzy sobą używają mechanizmu MIME (*Multipurpose Internet Mail Extensions* — uniwersalne rozszerzenia poczty internetowej). Jednakże istnieją typy danych, które wymagają skorzystania z aplikacji pomocniczych, zwanych zewnętrzny-mi przeglądarkami (*external viewer*). Jeśli typ danych w zadanym pliku nie należy do obsługiwanych przez MIME, przeglądarka przekazuje sterowanie do aplikacji pomocni-czej, która obsługuje dany plik.

Nowa wersja protokołu MIME jest Secure/MIME (S/MIME), opracowany w odpowie-dzi na rozpowszechnione przechwytywanie i falszowanie poczty. Protokół ten umozli-wia szyfrowanie wiadomości. Wprawdzie S/MIME nie jest jak na razie stosowany po-wszechnie, lecz przedżej czy później użytkownicy będą mogli go używać do wysyłania bezpiecznych wiadomości e-mail.

Aplikacje służace do przesyłania plików

Serwery FTP pozwalają przesyłać pliki przez Internet. Najczęściej stosowanym rozwia-zaniem jest korzystanie z anonimowego FTP do pobierania plików z publicznych serwe-rów FTP. Aby jednakże przesłać plik do serwera FTP, musimy posiadać odpowiednie uprawnienia. Dostępnych jest wiele aplikacji, które pozwalają pobierać pliki z serwe-rów FTP i zapisywać je w nich. Aplikacje FTP dla systemów Unix i MS-DOS posiadają interfejs wiersza poleceń, natomiast aplikacje dla Windows (na przykład WS_FTP) po-siadają interfejs graficzny.

Aplikacja Telnet

Telnet jest bardzo stara aplikacja internetowa, która pozwala logować się do zdalnych komputerów. Za pomocą tej aplikacji możemy połączyć się ze zdalnym komputerem, zalogować i pracować tak, jak na konsoli lokalnej. Telnet może też służyć do łączenia się z tysiącami katalogów bibliotecznych na całym świecie.

Aplikacje e-commerce

Aplikacje e-commerce pozwalają użytkownikom przeprowadzać transakcje handlowe za pomocą Internetu, zarówno ze strony konsumenta, jak i dostawcy. Handel w Sieci ma następujące zalety:

- ◆ Pomiedzy klientem i sprzedawcą mogą być wymieniane informacje: szczegółowe dane o dostępnych produktach i usługach, pomoc techniczna dla klienta dostępna online, odpowiedzi na pytania klienta itd.
- ◆ Firmy mogą łączyć własne zakresy kompetencji z innymi, aby udostępniać produkty i usługi.
- ◆ Dystrybucja produktów może odbywać się fizycznie i elektronicznie.

Jezyki

Jezyki uzywane do tworzenia uslug w Sieci mozemy ogólnie podzielic na jezyki znakowania, jezyki programowania, jezyki do pisania skryptów wykonywanych po stronie klienta i technologie serwerowe.

Jezyki znakowania

Przydatnosc jezyków znakowania (adiustacyjnych), takich jak SGML i HTML, zostala omówiona wczesniej. Do innych jezyków znakowania uzywanych w Sieci zaliczaja sie XML (*eXtensible Markup Language*) i XHTML (*Extensible HyperText Markup Language*).

XML

Jezyk XML (*eXtensible Markup Language* — rozszerzalny jezyk znakowania) jest uzywany obecnie w aplikacjach handlu elektronicznego (e-commerce) jako wspólny format wymiany danych. XML jest standardem, sluzacym do opisywania danych w Sieci, który do tego celu uzywa znaczników. Struktura dokumentu XML jest podobna do struktury dokumentu napisanego w jezyku HTML, jednakze w przeciwienstwie do tego jezyka, XML nie posiada zadnych wstepnie zdefiniowanych znaczników. Jest to meta-jezyk, który pozwala tworzyc własne jezyki znakowania (inaczej slowniki) na potrzeby opisu danych zawartych w dokumencie. Mozemy na przyklad utworzyc znacznik <NAZWISKO-KLIENTA> i w jego obrebie definiowac nazwiska klientów. W ten sposob mozemy identyfikowac praktycznie dowolne typy danych (np. produkt, przedstawiciela handlowego, nalezna kwote), co pozwala uzywac stron WWW jak rekordów bazy danych. XML opisuje w dokumencie jego zawartosc, natomiast HTML koncentruje sie na prezentacji zawartosci na stronie WWW. XML jest formatem danych opartym na zwykłym tekscie, dzieki czemu nie zależy od platformy. Inaczej mówiac, dokumenty XML moga byc wyswietlane w dowolnym urzadzeniu, na przyklad komputerze osobistym, laptopie lub palmtopie. W istocie dzieki XML-owi powstalo kilka nowych jezyków znakowania, na przyklad WML — zastosowanie XML-a, pozwalajace tworzyc aplikacje dostepne z telefonów komórkowych.

Spójrzmy na nastepujacy dokument XML:

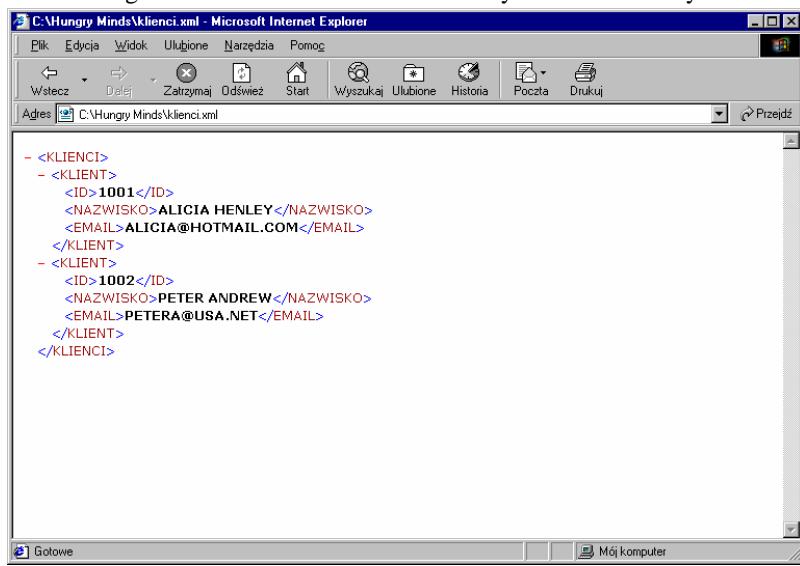
```
<KLIENCI>
  <KLIENT>
    <ID>1001</ID>
    <NAZWISKO>ALICIA HENLEY</NAZWISKO>
    <EMAIL> ALICIA@HOTMAIL.COM</EMAIL>
  </KLIENT>
  <KLIENT>
    <ID>1002</ID>
    <NAZWISKO>PETER ANDREW</NAZWISKO>
    <EMAIL>PETERA@USA.NET</EMAIL>
  </KLIENT>
</KLIENCI>
```

Ten dokument XML zawiera informacje o klientach. Jak widac, dokument w jezyku XML *nie zawiera* informacji o wygladzie danych, lecz same dane. Jesli otworzymy go w przegladarkce Internet Explorer 5.0, otrzymamy domyslny uklad dokumentu XML, który zostanie wyswietlony w formie drzewa, jak na rysunku 15.3.

Aby zmienic domyslny wyglad dokumentu XML, musimy utworzyc arkusze stylów. Dokument XML mozemy wyswietlic w zadanym formacie uzywajac CSS (*Cascading Style Sheets*) lub jazyka XSL (*eXtensible Style Sheet Language*). Na przyklad, jesli dane zawarte w dokumencie XML maja byc wyswietlone w formie tabeli, mozemy wykorzystac CSS albo XSL. Arkusze stylów zawieraja polecenia dla przeglądarki, jak ma ona tlumaczyc strukture zródlowego dokumentu XML na strukture wyswietlana dla uzytkownika.

Rysunek 15.3.

Przykładowy fragment kodu XML



XHTML

XHTML (*Extensible HyperText Markup Language* — rozszerzalny HTML) jest jazykiem HTML 4.0 zredefiniowanym jako aplikacja XML. XHTML jest podobny do jazyka HTML 4.0 z kilkoma niewielkimi różnicami. Na przyklad, XHTML wymaga poprawnego zamkniecia wszystkich znaczników. Po drugie, zagniezdzanie znaczników również musi odbywac sie prawidlowo. Spójrzmy na ponizsza próbke:

```

<HTML>
  <BODY>
    <FONT COLOR="RED">
      <P>Pierwszy paragraf
    </FONT>
    </P>
  </BODY>
</HTML>

```

Znacznik `<P>` został otwarty wewnatrz znacznika `` i zamkniet po zamknieciu ``. W jazyku HTML nie spowodowałoby to bledu, natomiast w XHTML-u — tak. Jazyk XHTML wymaga zamkniecia wewnętrznych znaczników w pierwszej kolejnosci.

Kolejna różnica pomiedzy HTML-em i XHTML-em jest niezmienny zestaw znaczników w HTML-u, do którego nie mozna wprowadzic zadnych zmian. XHTML oprócz obslugi wszystkich znaczników HTML-a posiada mozliwosc rozszerzania. Mozemy definiowac i dodawac nowe znaczniki i atrybuty do istniejacych, co umozliwia nowe m-

tody osadzania zawartosci i programowania stron WWW. W standardzie XHTML 1.0 autorzy moga mieszac znane skladniki jazyka HTML 4 z elementami innych aplikacji XML, miedzy innymi opracowanymi przez W3C dla zastosowan multimedialnych. XHTML jest jazykiem zalecanym przez W3C do tworzenia stron WWW.

Jezyki programowania

Najpopularniejszym jazykiem programowania stron WWW jest Java, m.in. z tego powodu, ze pozwala budowac aplikacje, ktore mozna wykonywac na dowolnej platformie sprzutowej lub programowej bez koniecznosci zmian. Jazyk Java został opracowany przez Sun Microsystems i opiera sie na jazyku C++. Kod Javy moze byc zarówno interpretowany, jak i kompilowany. Gdy program w tym jazyku jest kompilowany, kompilator przekształca kod w tzw. *kod posredni (bytecode)*, niezalezny od platformy sprzutowej. Interpreter jazyka Java, Java Virtual Machine (JVM), tlumaczy kod posredni na kod, który dany komputer potrafi „zrozumieć”. Dzieki temu programy napisane w jazyku Java mozna uruchamiac w dowolnym komputerze zawierajacym oprogramowanie JVM.

W Javie mozemy pisac aplikacje niezalezne lub apety. Gdy uzytkownik zada aplikacji w Javie, jest ona ladowana i wykonywana w serwerze WWW, natomiast apety sa osadzane w stronach HTML. Gdy uzytkownik pobiera strone HTML zawierajaca aplet, wówczas przegladarka wywoluje JVM, konwertuje instrukcje z apletu na jazyk maszynowy i wykonuje je.

Skrypty wykonywane po stronie klienta

Skrypt oznacza program zawierajacy zestaw instrukcji dla aplikacji. Mozna go osadzic w stronie WWW za pomoca znacznika <SCRIPT>. Jazyki skryptowe, w przeciwienstwie do pelnych jazykow programowania, wymagaja srodowiska macierzystego i sluzą do rozszerzania funkcjonalnosci lub manipulacji programistycznych aplikacji macierzysta.

Skrypty typu *client-side* sa wykonywane po stronie klienta — ich interpretacja i wykonaniem zajmuje sie przegladarka. Gdy uzytkownik wysyla zadanie dokumentu HTML, który zawiera skrypt wykonywany po stronie klienta, serwer wysyla ten dokument HTML razem ze skryptem do przegladarki. Ta z kolei wykonuje polecenia skryptu, gdy wystapi określony warunek — na przykład klikniecie mysza. Dzieki temu skrypty wykonywane po stronie klienta zwiększają interaktywnosc dokumentu — następuje reakcja na zdarzenia ze strony uzytkownika. Do najpopularniejszych jazyków skryptowych uzywanych po stronie klienta naleza:

- ♦ *ECMAScript, JavaScript i JScript* — ECMAScript jest międzyplatformowym jazykiem skryptowym o standardzie przemysłowym, zaprojektowanym do współpracy z różnorodnymi przegladarkami. Jazyk ten łączy funkcjonalność JavaScript i JScript.

JavaScript został opracowany przez NetScape Communications. Na tym jazyku opiera się standard przemysłowy ECMA (European Computer Manufacturers Association). JavaScript jest podziobrem jazyka programowania Java o ograniczonych możliwościach — inaczej mówiąc, nie pozwala tworzyć autonomicznych aplikacji.

JScript, opracowany przez Microsoft, jest bardzo podobny do JavaScript i stanowi rozszerzoną implementację ECMAScript.

- ◆ *VBScript* — „odchudzona” wersja języka Microsoft Visual Basic. Nie jest tak powszechnie obsługiwany jak JavaScript, lecz część użytkowników preferuje ten język z uwagi na łatwość użycia.

Spójrzmy na poniższy kod:

```
<HTML>
<HEAD>
<TITLE>PRZYKŁAD SKRYPTU</TITLE>
<SCRIPT LANGUAGE="VBSCRIPT">
FUNCTION CHANGEFONT()
    PARA1.style.color="RED"
    PARA1.style.fontSize=28
END FUNCTION
</SCRIPT>
</HEAD>
<BODY>
    <P ID="PARA1" ONMOUSEOVER="CHANGEFONT">WITAMY W HTML-U
    </P>
</BODY>
</HTML>
```

Ten przykład zawiera skrypt wykonywany po stronie klienta, osadzony w stronie HTML. Gdy zazadamy tej strony, w oknie przeglądarki zostanie wyświetlony w kolorze czarnym tekst WITAMY W HTML-U. Gdy naprowadzimy kurSOR na tekst, zajdzie zdarzenie ONMOUSEOVER i przeglądarka wykona funkcję skojarzoną ze zdarzeniem. Funkcja CHANGEFONT zmienia kolor i rozmiar tekstu zawartego w akapicie, posiadającym identyfikator PARA1. W podobny sposób za pomocą skryptów wykonywanych po stronie klienta możemy zmieniać obrazy, tworzyć interaktywne menu itp.

Technologie serwerowe

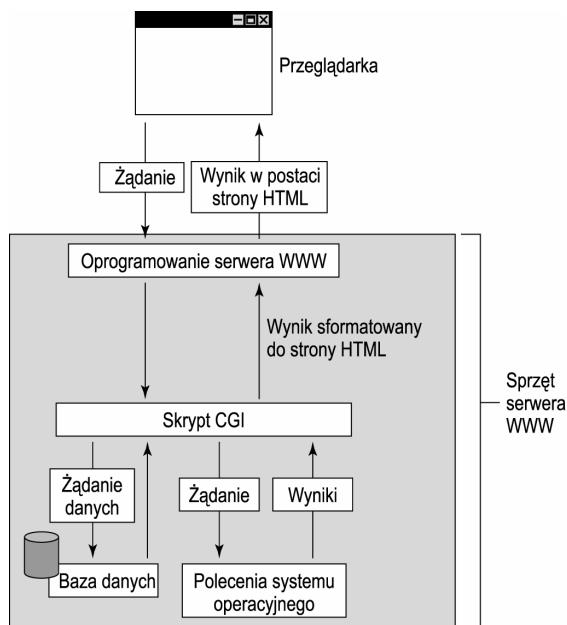
W przeciwieństwie do powyższych skryptów, wykonywanych przez przeglądarkę, skrypty serwerowe (*server-side*) są wykonywane przez serwer WWW. Do popularnych technologii WWW należą:

- ◆ *Skrypty CGI (Common Gateway Interface — wspólny interfejs bramy)* — programem CGI może być mały program napisany w dowolnym języku, między innymi C, C++, Java lub Perl. Oprócz tego program CGI jest niezależny od platformy; może działać pod dowolnym systemem operacyjnym, w tym Windows NT, Unix, Mac i OS2. CGI jest brama pomiędzy dokumentem HTML i innymi programami, uruchamianymi w serwerze WWW. Na przykład, strona HTML może wywołać skrypt CGI po kliknięciu przycisku przez użytkownika. Skrypt może pobierać dane z bazy danych, sformatować wyniki do postaci strony HTML i wysłać z powrotem do użytkownika. Początkowo skrypty CGI służyły do tworzenia interakcji stron WWW z bazami danych, jednakże wraz z rozwojem WWW zostały opracowane bardziej wydajne technologie skryptów wykonywanych po stronie serwera, między innymi ASP (*Active Server Pages*), JSP (*Java Server Pages*) i serwlety. Funkcjonowanie skryptu CGI przedstawia rysunek 15.4.



Perl (od *Practical Extraction Recursive Language*) jest jednym z najpopularniejszych z dostępnych dzisiaj serwerowych języków skryptowych, używanym w różnorodnych zastosowaniach.

Rysunek 15.4.
Sposób działania skryptów CGI



- ◆ **ASP** — technologia serwerowa Microsoftu. W przeciwienstwie do skryptów CGI, które są odrebnymi programami, kod ASP jest osadzony w stronie WWW pomiędzy symbolami <% i %>. Pliki zawierające kod ASP mają rozszerzenie .asp. Kod ten może być napisany w języku VBScript lub JScript. ASP używa do komunikacji z przeglądarką zbioru wstępnie zdefiniowanych obiektów — na przykład, do wysłania wyniku do użytkownika służy obiekt Response (odpowiedź). Analogicznie, do uzyskania informacji od użytkownika służy obiekt Request (zadanie). Kod ASP może być uruchomiony w serwerach IIS (*Internet Information Server*) lub PWS (*Personal Web Server*).

Gdy użytkownik zada pliku zawierającego kod ASP, IIS przekazuje zadanie do *silnika ASP*. Jest to składnik działający w serwerze WWW, który odczytuje plik zawierający kod ASP wiersz po wierszu i wykonuje ten kod. Na koniec strona, zawierająca znaczniki HTML, zostaje zwrocona do przeglądarki.

ASP pozwala dynamicznie modyfikować dane, odpowiadając na zapytania użytkowników i korzystając z baz danych. Co więcej, jest techniką szybszą od CGI i Perla oraz łatwiejszą do naużenia się.

- ◆ **Serwlety** — serwlet jest aplikacja Java, działająca w serwerze WWW. Ta technologia została opracowana przez firmę Sun, z zamarem zastąpienia CGI. Serwlety pozwalają budować strony WWW „w locie” na podstawie informacji dostarczonych przez użytkownika. Serwlety są bardziej wydajne niż CGI, a ponieważ są pisane w języku Java, zyskują ważną cechę — przenośność.

- ♦ *JSP* — rozwiazanie alternatywne wzgledem ASP, opracowane przez firmy Sun i Netscape. JSP jest strona HTML z osadzonym kodem zrodlowym w jzyku Java. HTML zapewnia rozklad strony, zas Java dodaje zdolnosc do przetwarzania. JSP sa konwertowane na serwlety, zas napotkany po raz pierwszy serwlet jest kompilowany.

Bezpieczenstwo w Sieci

W ciagu ostatnich dziesieciu lat rosnaca popularnosc WWW spowodowala, iz coraz wiecej uzytkownikow uzyskuje polaczenie z Internetem. W tej sytuacji musimy rozwazyc kwestie bezpieczenstwa w Sieci. Podstawowym wymogiem bezpieczenstwa w Sieci jest zdolnosc do skladowania, wysylania i odbioru waznych danych w sposob nie pozwalajacy osobom nieupowaznionym na manipulacje nimi.

Jesli system operacyjny nie jest bezpieczny, moze pasc ofiara wlamania, wobec czego powinnismy stosowac wszelkie mozliwe srodkie, aby system zabezpieczyc. Radzimy kierowac sie nastepujacymi zasadami:

- ♦ Uruchom tylko wymagane uslugi i usun niepotrzebne.
- ♦ Zamknij nieuzywane otwarte porty. Porty takie moga posluzyc hakerowi do wlamania sie do systemu.
- ♦ Instaluj najnowsze pakietы uslugowe (Service Pack).
- ♦ Instaluj najnowsze aktualizacje i laty zwiazane z bezpieczenstwem.
- ♦ Korzystaj regularnie z oprogramowania antywirusowego szukajacego wirusow i koni trojanskich.
- ♦ Nie odwiedzaj witryn, do ktorych nie masz zaufania.
- ♦ Unikaj korzystania z Internetu z konta Administrator (w systemach Windows) oraz root (w systemach Unix i Linux). Moze to narazic system na bardzo niebezpieczne ataki.

Poziom bezpieczenstwa, który chcemy uzyskac zalezy od kilku czynnikow: zasobow, infrastruktury technologicznej i poziomu wymagan dla skladowania, wysylania i odbierania waznych informacji przez Internet. Zarowno dla systemow operacyjnych Windows, jak i Unix/Linux dostepne sa narzeczaowych producentow, pozwalajace zapewnic bezpieczenstwo podczas pracy w Sieci.

W ciagu ostatnich dziesieciu lat bezpieczenstwo systemow operacyjnych uleglo ogromnej poprawie. Certyfikowane i zatwierdzone rozwiazania zostaly wprowadzone dla takich systemow operacyjnych, jak Unix, Linux i Windows NT/2000.



Zaimplementowanie zabezpieczeń systemu z wykorzystaniem odpowiednich narzędzi wymaga sporej mocy obliczeniowej i przestrzeni dyskowej. Gdy by zaczac implementowac zabezpieczenia w serwerach o nizszych osiagach, liczba uzytkownikow, ktora serwer moze obsluzyc, moze sie zmniejszyc.

Aby ochronic system przed zagrozeniami bezpieczenstwa, powinnismy stosowac określone konwencjonalne elementy zabezpieczeń internetowych. Naleza do nich zapory firewall, szyfrowanie i uwierzytelnianie.

Zapory firewall

Zapory firewall sa wyjątkowo popularnym zabezpieczeniem, powszechnie stosowanym przez organizacje. Zapory zezwalaja jedynie na autoryzowane polaczenia pomiędzy komputerami w sieci wewnętrznej i komputerami w sieciach zewnętrznych. Zapora firewall moze byc tak skonfigurowana, by tylko autoryzowani uzytkownicy mieli dostep do Internetu. Zapory moga tez blokowac poczta elektroniczna, FTP i zdalne logowanie. Producenci systemów zapór firewall koncentruja sie ponadto na dodawaniu nowych funkcji do zapór, na przykład szyfrowania i wirtualnych sieci prywatnych (VPN).

Szyfrowanie

Szyfrowanie jest bardzo przydatnym sposobem na zabezpieczenie danych w Internecie. Szyfrowanie wykorzystuje algorytmy zmieniajace sposób zakodowania danych. Danych po zaszyfrowaniu nie da sie zinterpretowac lub odszyfrowac bez spełnienia określonych warunków wstępnych, dzięki czemu szyfrowanie jest dobrym sposobem na zapewnienie prywatności. Zaleta szyfrowania jest możliwość zabezpieczenia danych składowanych w sieci komputerowej, jak również przesyłanych przez sieć. Najłatwiej dostępny standardem szyfrowania jest DES (*Digital Encryption Standard*), wspierany przez National Institute of Standards and Technology od roku 1975. Programy korzystające z DES moga byc obecnie używane poza granicami USA.

RSA

W roku 1978 Ron Rivest, Adi Shamir i Len Adleman opracowali algorytm, który później otrzymał nazwę RSA (*Rivest Shamir Adleman*). RSA jest jednym z pierwszych i najpowszechniej stosowanych algorytmów klucza publicznego. Moze służyć do szyfrowania i deszyfracji oraz podpisywania i weryfikacji danych, aby zapewnić ich integralność.

PGP

PGP (*Pretty Good Privacy* — całkiem niezła prywatność) jest również bardzo przydatnym narzędziem szyfrowania. PGP pomaga użytkownikom szyfrować dane zapisane w ich komputerach, jak również wysyłać zaszyfrowane wiadomości poczta elektroniczna. PGP generuje dwa typy kluczy — publiczny i prywatny. Klucz publiczny użytkownika jest wysyłany do serwera WWW i dostępny dla wszystkich. Klucz prywatny jest poufny i znany tylko użytkownikowi. Za każdym razem, gdy użytkownik wysyła zaszyfrowaną wiadomość, dokonuje szyfrowania za pomocą klucza publicznego odbiorcy. Po otrzymaniu tej wiadomości odbiorca może ją odszyfrować za pomocą swojego klucza prywatnego. PGP udostępnia również narzędzia do tworzenia kluczy i zarządzania nimi.

SSL

SSL (*Secure Sockets Layer*) jest protokołem, który służy do zabezpieczania różnorodnych aplikacji używanych do poruszania się w Sieci. Do aplikacji tych zalicza się poczta elektroniczna, e-commerce i oparte na WWW usługi subskrypcyjne. SSL używa połączenia technologii klucza tajnego i klucza publicznego, aby zapewnić bezpieczeństwo.

stwo danych przesyłanych przez siec komputerowa. Protokół ten zapewnia prywatnosc i uwierzytelnianie uzytkowników, oraz integralnosc wiadomosci przesyłanych przez siec.

Uwierzytelnianie i integralnosc

Te dwa zagadnienia sa dzis wzajemnym problemem dla administratorow systemow. Uwierzytelnianie mozna przeprowadzac za pomoca hasel, lecz hasla przesyłane przez Internet w postaci nie zaszyfrowanej sa latwe do przechwycenia. Dlatego tez powinno sie uzywac narzedzi szyfrujacych: SecureSSH, SSL, Kerberosa lub LDAP.

Integralnosc danych oznacza, iz dane nie zostaly zmodyfikowane podczas transmisji. Trudno jest dzis zapewnic integralnosc i poufnosc przesyłanych przez Internet danych, gdy dostepnych jest tak wiele narzedzi hakerskich. Lecz sa na to sposoby — na przyklad podpisy cyfrowe. Podpisy cyfrowe sa etykietami dolaczonymi do konca wiadomosci, gwarantujacymi wyslanie wiadomosci z autentycznego zródła. Oprócz podpisów cyfrowych mozemy stosowac tzw. *odciski palca (fingerprint)*, bedace 14-znakowym podpisem uwierzytelniającym. Podpisy cyfrowe i odciski palców szybko zyskują na popularnosci.

Handel elektroniczny w Internecie

W dzisiejszym srodowisku handlowym najswiezszym sloganem jest *e-commerce*. W doslownym znaczeniu e-commerce oznacza handel przeprowadzany przez dowolny nosnik elektroniczny, lecz obecnie slowo to oznacza handel w Internecie. W przeciagu bardzo krótkiego czasu handel elektroniczny otworzył nowe rynki i rozszerzył wydajnosc tradycyjnych sklepów. W chwili obecnej ponad 60 procent uzytkowników Internetu w USA kupuje towary w Sieci. Popularnosc handlu elektronicznego rośnie, poniewaz ma on kilka zalet w stosunku do tradycyjnych metod handlu:

- ◆ *Dostepnosc* — witryny sklepowe sa dostepne caly czas, z dowolnego miejsca na swiecie.
- ◆ *Zmniejszone koszty wlasne* — dzieki mniejszej ilosci pracy papierkowej, koszty wlasne sa minimalne w porównaniu z handlem tradycyjnym.
- ◆ *Skrócony czas transakcji* — transakcje dokonywane sa szybciej w porównaniu z tradycyjnym handlem.

Jedna z podstawowych różnic pomiędzy e-commerce i handlem tradycyjnym jest sposób, w jaki odbywa sie komunikacja. W przypadku e-commerce komunikacja odbywa sie przez Internet, natomiast w tradycyjnym srodowisku handlowym sa to rozmowy twarza w twarz, przez telefon lub korespondencja pomiędzy sprzedajacym a klientem. W handlu elektronicznym inne sa również formy platosci. Poniewaz wiekszosc transakcji zachodzi elektronicznie, prawie bez kontaktu miedzy sprzedawcą i klientem, decydujace znaczenie ma bezpieczenstwo.

Handel przez Internet mozemy ogólnie podzielic na dwa różne modele: przedsiebiorstwo-przedsiebiorstwo i przedsiebiorstwo-konsument.

Model przedsiębiorstwo-przedsiębiorstwo

W tym modelu, określonym skrótem B2B (*business-to-business*) transakcje odbywają się przez Internet pomiędzy dwoma różnymi przedsiębiorstwami. Interakcja pomiędzy nimi może przybierać formę składania zamówień, odbierania faktur i dokonywania płatności. B2B charakteryzuje się transakcjami na dużą skalę i na niskiej marży. Oznacza to, że ilość kupowanych produktów lub usług jest wysoka, lecz narzuty cenowe na produkt są niskie. Ten model handlu funkcjonuje w Internecie od dłuższego czasu.

Model przedsiębiorstwo-konsument

W tym modelu, określonym terminem B2C (*business-to-consumer*), transakcje odbywają się pomiędzy firmą i konsumentem. W B2C umowy kupna-sprzedazy charakteryzuje mała skala i wysokie marże, co oznacza niewielkie ilości kupowanych produktów lub usług (w porównaniu z modelem B2B), lecz narzut cenowy jest wysoki. W chwili obecnej konsumenti mogą za pomocą Internetu kupować towary bardzo różne — książki, artykuły spożywcze lub samochody. Usługa ta nosi nazwę zakupów elektronicznych i jest idealnym przykładem modelu przedsiębiorstwo-konsument.

Wideo i inne zaawansowane typy danych

W ciągu ostatnich dziesięciu lat Internet stał się stopniowo podstawowym środkiem przesyłania danych. Początkowo informacje były przesyłane przede wszystkim w postaci tekstu i grafiki, lecz wraz z pojawieniem się rozwiniętych technologii multimedialnych zaczęto wykorzystywać obrazy, dźwięk i wideo zapisane w postaci cyfrowej. W chwili obecnej około 40 procent danych pobieranych z Internetu ma formę potoków audio i wideo. Pobieranie i odtwarzanie plików audiowizualnych o wysokiej jakości z Internetu stało się rzeczywistością. Popularne przeglądarki WWW, na przykład Netscape Navigator i Internet Explorer, w pełni obsługują multimedialne typy danych, obejmujące tekst, dźwięk, nieruchome obrazy, obiekty graficzne i cyfrowe wideo.

Potokowa transmisja audio i wideo

Do niedawna najczęściej pobieranymi przez Internet plikami były pliki multimedialne. Jednak musi on zostać sciagnięty w całości do klienta przed odtworzeniem. Zależnie od szybkości połączenia, ładowanie dużych plików może zająć dużo czasu i zasobów sieciowych. Jeśli pliki takie są wyjątkowo duże, mogą zająć sporo cennej przestrzeni na dysku. Co więcej, ładowane pliki multimedialne najlepiej nadają się do małych grafik i krótkich „wizytówek” dźwiękowych. Nie są zbyt przydatne do odtwarzania dużych plików audio i wideo.

Potokowa transmisja obrazu i dźwięku (*streaming*) pozwala klientom odbierać treści audio i wideo z serwerów w dowolnym miejscu kuli ziemskiej i zacząć odtwarzanie pliku już po dotarciu do klienta pierwszych kilku bajtów potoku. Inaczej mówiąc, odbierający dane klient odtwarza nadsyłany strumień multimedialny w czasie rzeczywistym, w miarę napływu danych. W prawdzie pliki wysyłane strumieniowo nie są zapisywane na komputerze klienta, lecz można je zapisać na dysku twardym użytkownika do późniejszego odtworzenia. Transmisja potokowa najczęściej jest używana do odtwarzania

plików zarchiwizowanych. Technika ta nosi nazwę *transmisji potokowej na zadanie* (*on-demand streaming*). Technologia transmisji potokowej pozwala również na rozglądzanie i odbiór przez Internet zdarzeń rejestrowanych na żywo, na przykład koncertów muzycznych.



Wprawdzie potokowa transmisja dźwięku wciąż zyskuje na popularności, lecz przesyłanie w ten sposób wideo nie jest równie powszechnie akceptowane. Wciąż prowadzone są badania nad ta technologią, z nadzieją, że transmisja potokowa wideo przedżej czy później zdobędzie zainteresowanie naukowców i opinii publicznej.

Transmisja potokowa obejmuje kompresję danych audio i wideo, metody formatowania potoku, protokoły sieciowe i podział transmisji na pakiety. Technologia ta obejmuje również po stronie klienta projektowanie odtwarzania i synchronizacji różnych strumieni danych, a po stronie serwera projektowanie metod składowania i dostarczania danych. Do zalet transmisji potokowej należą:

- ◆ Łatwość pobierania plików przesyłanych potokowo.
- ◆ Możliwość odtwarzania plików już po kilku sekundach od kliknięcia przez użytkownika lacza.
- ◆ Transmisja potokowa łatwiej trafia do różnych użytkowników, ponieważ ludzi bardziej przyciąga dźwięk i wideo niż prosty opis tekstowy.



Do popularnych przykładów plików potokowych należą RealAudio, RealVideo, VDOLive i StreamWorks.

Coraz więcej witryn zaczyna używać technologii transmisji potokowej, aby przyciągnąć klientów. Istnieją dwa sposoby implementacji tej technologii w witrynach WWW, aby je ozycić i uatrakcyjnić: odsyłacze i osadzanie:

- ◆ *Odsyłacze* — w tej metodzie plik potokowy audio lub wideo nie jest bezpośrednio umieszczany na stronie WWW. Zamiast niego tworzy się odsyłacz (lacze) do *metapliku*, który jest plikiem odniesienia, zawierającym ścieżkę do pliku potokowego. Odniesienie to mówi przeglądarkę, gdzie znaleźć plik. Gdy gosc kliknie lacze reprezentowane przez metaplik, ten kieruje odtwarzacz danych potokowych do pliku potokowego. Zaleta tej metody jest szybkie ładowanie strony WWW, ponieważ dane potokowe nie stanowią części strony WWW. Jednakże dostęp do pliku potokowego przez kliknięcie lacza może zająć trochę czasu, zależnie od szybkości połączenia.
- ◆ *Osadzanie* — plik potokowy jest zawarty w samej stronie WWW, przez co załadowanie tej strony może potrwać. Osadzanie jest znacznie bardziej skomplikowane w porównaniu z odsyłaczami. Technika ta daje jednakże administratorowi WWW pełną kontroli nad plikiem medialnym.

Aby odtworzyć pliki potokowe na stronie WWW, konieczne jest zainstalowanie aplikacji pomocniczej lub moduł dodatkowy przeglądarki (*plug-in*), które pozwala na odtwarzanie w komputerze biurkowym plików audio lub wideo. Aby zastosować technologie transmisji potokowej w witrynie WWW, należy:

- 1. Zapisac i zredagowac pliki potokowe.** Po zainstalowaniu odpowiedniego oprogramowania musimy utworzyc i zredagowac pliki przeznaczone do transmisji potokowej. Do popularnych programów sluzacych do edycji potokowych plików dziekowych zaliczaja sie SoundForge (dla Windows), SoundEdit 16 plus Deck II (dla komputerów Macintosh), oraz CoolEdit. Jednym z najpopularniejszych programów sluzacych do edycji potokowych plików wideo jest Adobe Premiere.
- 2. Zakodowac pliki potokowe.** Po zapisaniu i edycji nagran, pliki potokowe musza zostac zakodowane tak, by mogly byc odtwarzane przez aplikacje pomocnicze lub moduly dodatkowe przegladarek przeznaczone do tego celu. Najpopularniejszymi dzis programami sluzacymi do kodowania sa Real Audio Encoder (dla plików audio) oraz Real Video Encoder (dla plików wideo). Warto zakodowac ten sam plik audio (wideo) w kilku różnych formatach, co pozwoli na odtwarzanie przez rózne programy pomocnicze.
- 3. Skopiowac pliki do serwera WWW.** Aby udostepnic pliki innym uzytkownikom, musimy je zapisac w serwerze WWW. Do udostepniania tych plików musi byc zainstalowane i skonfigurowane po stronie serwera specjalne oprogramowanie, na przyklad Real Audio. Wszystkie potokowe pliki wideo i czesc potokowych plików audio wymaga obecnosci tego oprogramowania. Inne potokowe pliki audio, na przyklad w formacie Internet Wave, moga byc odtwarzane ze zwyklego serwera WWW, lecz wymagaja modyfikacji pewnych ustawien konfiguracji takiego serwera. Po skonfigurowaniu oprogramowania po stronie serwera nalezy skopiowac do niego zakodowane pliki za pomoca odpowiedniej aplikacji, na przyklad FTP.
- 4. Przetestowac pliki.** Wprawdzie ten krok jest czasami pomijany, lecz wzazne jest, by przetestowac udostepniane pliki w celu weryfikacji, czy sa właściwie odtwarzane. Do tego celu moze posluzyc odpowiednie oprogramowanie pomocnicze lub modul rozszerzajacy.



Jesli uzywamy do odtwarzania plików potokowych modulu rozszerzajacego (plug-in), konieczne jest skonfigurowanie przegladarki WWW; samo zainstalowanie modulu plug-in nie wystarczy. Jesli zas do odtwarzania plików potokowych sluzy program pomocniczy, po jego zainstalowaniu w komputerze musimy do korzystania z tej aplikacji skonfigurowac przegladarkę WWW.

Co trzeba brac pod uwage przy transmisji potokowej

Wprawdzie dazymy obecnie do szybszych technologii sieciowych typu ISDN (*Integrated Services Digital Network*) oraz ATM (*Asynchronous Transfer Mode*), lecz istniejace infrastruktury sieciowe — w tym Internet — nie byly projektowane do obslugi transmisji potokowych. Transmisje plików wymagaja przesyłania danych z serwerów do klientów z duza predkoscia, która trzeba utrzymac na stalem pozio mie. Powoduje to, ze musimy brac pod uwage wiele czynnikow zwiazanych z uzytkowaniem technologii transmisji potokowej w dostepnej obecnie infrastrukturze sieci komputerowych:

- ◆ Standardy transmisji potokowej nie sa jeszcze w pelni dojrzale, co prowadzi do sytuacji, w której kazdy producent tworzy własne standardy. Na skutek tego wsparcie ze strony niezale znych producentów jest wyjatkowo ograniczone.
- ◆ Przegladarki obsługujace te technologie obsluguja media bez problemu, lecz inne przegladarki, nie obsługujace transmisji potokowej, moga zawiesic sie na dobre.

- ♦ Transmisja potokowa jest kosztowna technologia. Osrodki WWW korzystajace z niej donosza o wysokich kosztach utrzymania.
- ♦ Witryny WWW stosujace technologie transmisji potokowej laduja sie dluzej.

Wprawdzie ta technologia wyszla juz z wieku dziecięcego i — przy rosnącym zapotrzebowaniu na dane audiowizualne w Internecie — szybko zmierza w stronę dojrzalosci i ustabilizowania, lecz nadal na tym polu pozostaje wiele do zrobienia. Standardy i protokoly transmisji potokowej sa nadal opracowywane.

Rozdział 16.

Dostęp

do poczty elektronicznej i

grup dyskusyjnych

W tym rozdziale:

- ◆ Działanie poczty elektronicznej
- ◆ Działanie grup dyskusyjnych

Obecnie w biznesie szybki dostęp do informacji jest wyjątkowo ważny. Pojawienie się Internetu i elektronicznych usług przesyłania wiadomości pozwala użytkownikom wysyłać i odbierać informacje w ciągu sekund. Wiele osób kupuje swój pierwszy komputer po to, aby uzyskać dostęp do usług poczty elektronicznej.

W niniejszym rozdziale omówimy jedno z najczęściej spotykanych zastosowań TCP/IP: do wysyłania i odbierania poczty elektronicznej i korzystania z sieciowych grup dyskusyjnych. Opiszemy cały proces przesyłania poczty elektronicznej oraz protokły z pakietu TCP/IP (SMTP i NNTP), które pozwalają przesyłać poczty i wiadomości z grup dyskusyjnych. Na koniec omówimy również ważne zagadnienie — zasady etykiety internetowej.

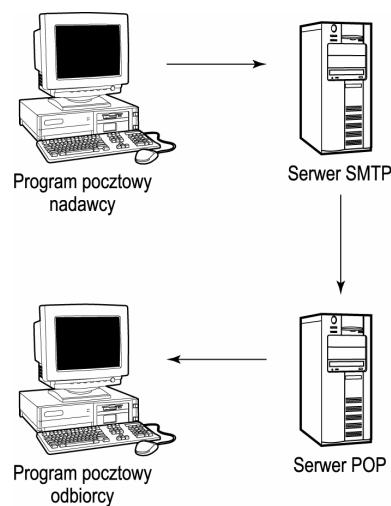
Wprowadzenie do poczty elektronicznej

Dla użytkowników, którym wystarczy kliknąć przycisk *Wyślij* po napisaniu listu, działanie poczty elektronicznej może wydawać się stosunkowo proste; jednakże proces wysyłania wiadomości obejmuje dość złożone czynności. Zanim omówimy techniczne aspekty wysyłania i odbierania wiadomości e-mail, przedstawimy ogólne streszczenie procesu przesyłania poczty.

Gdy użytkownik wysyła wiadomość za pomocą programu pocztowego, na przykład Microsoft Outlook Express, wiadomość ta najpierw dociera do serwera SNMP. Na podstawie adresu adresata serwer SNMP decyduje, jak przekierować wiadomość. Jeśli serwer docelowy wiadomości e-mail i nadawcy to ten sam serwer, wiadomość pozostaje w serwerze i zostaje przesłana do skrzynki odbiorczej adresata. W przeciwnym razie

wiadomosc zostaje wyslana do odpowiedniego serwera. Wielosc uzytkownikow nie posiada stalego polaczenia z Internetem, wobec tego wiadomosci musza pozostawac w serwerze, dopoki nie zostana odczytane. Wiadomosci te zbiera i przechowuje serwer POP. Klienci lacza sie z serwerem POP, podajac nazwe uzytkownika i haslo. Po zwierdzeniu nazwy uzytkownika, serwer wysyla wiadomosci do jego lub jej skrzynki komunikatow wchodzacych (*Inbox*). Proces przesyłania poczty elektronicznej jest przedstawiony na rysunku 16.1.

Rysunek 16.1.
Dzialanie poczty elektronicznej



W procesie wysylania wiadomosci e-mail wymagane sa: agent uzytkownika poczty (*Mail User Agent*), agent przesyłajacy poczte (*Mail Transfer Agent*) i agent dostarczajacy poczte (*Mail Delivery Agent*). Zadania tych składników sa nastepujace:

- ◆ *Mail User Agent (MUA)* — sluzi do tworzenia i wysyłania poczty. MUA pelni funkcje interfejsu pomiedzy uzytkownikiem i agentem przesyłajacym poczte.
- ◆ *Mail Transfer Agent (MTA)* — gra role urzedu pocztowego, przesyłajac wiadomosci otrzymane od MUA do miejsca przeznaczenia.
- ◆ *Mail Delivery Agent (MDA)* — oprogramowanie dostarczajace wiadomosci do skrzynki pocztowej uzytkownika.

Aby wysylac i odbierac wiadomosci, program pocztowy potrzebuje:

- ◆ Polaczenia z Internetem.
- ◆ Serwera POP, który udostepnia protokół POP (*Post Office Protocol* — protokół urzedu pocztowego). Serwer ten zajmuje sie przychodzacymi wiadomosciami. Serwery POP zwykle identyfikowane sa przez numer wersji, na przyklad POP3.
- ◆ Serwer SMTP, który udostepnia protokół SMTP (*Simple Mail Transfer Protocol* — prosty protokół przesyłania poczty). SMTP wykorzystuja do przesyłania wiadomosci serwery poczty elektronicznej w Internecie i sieciach intranetowych.



Jeden serwer możemy skonfigurować zarazem jako serwer SMTP i POP.

Sam proces wysyłania wiadomości e-mail jest mniej więcej taki sam dla wszystkich programów pocztowych. Na przykład, Microsoft Outlook Express i Netscape Messenger wysyłają poczty następująco:

1. Użytkownik tworzy wiadomość i wpisuje identyfikator lub adres e-mail odbiorcy w polu *To:* (*Do:*).
2. Jeżeli użytkownik chce wysłać kopie wiadomości do innych użytkowników, ich adresy poczty elektronicznej powinny zostać wpisane do pola *Cc:* (*carbon copy*). Wypełnienie pola *Subject:* (*Temat*) nie jest konieczne, lecz stanowi dobry zwyczaj.
3. Użytkownik kliknie przycisk *Send* (*Wyslij*), aby wysłać wiadomość.

Do wiadomości e-mail możemy dodawać pliki. W programie Microsoft Outlook Express dodanie załącznika odbywa się następująco:

1. Napisz list. Z paska narzędzi na górze okna wiadomości wybierz *Wstaw/Plik załącznika*.
2. Otworzy się okno dialogowe *Wstaw załącznik*. Wybierz plik przeznaczony do wysyłania i kliknij *Dolacz*.
3. Kliknij *Wyslij*.

Aby wysłać załączniki za pomocą programu pocztowego Netscape Messenger:

1. Napisz list. Kliknij przycisk *Attach* (*Dolacz*) z paska narzędzi Messenger. Z rozwijanego menu, które się pojawi, wybierz *File* (*Plik*).
2. Otworzy się okno dialogowe *Attach* (*Dolacz*). Wybierz plik przeznaczony do wysyłania i kliknij *Open* (*Otwórz*).
3. Kliknij *Send* (*Wyslij*).

SMTP

Simple Mail Transfer Protocol jest protokołem TCP/IP obsługującym przesywanie wiadomości poczty elektronicznej z jednego systemu pocztowego do drugiego. Do wykonywania tego zadania SMTP używa portu o numerze 25. SMTP przesyła poczty w sposób wiarygodny i wydajny.



Protokół SMTP jest opisany w RFC 821.

Gdy wiadomość e-mail dociera do serwera SMTP, zostaje umieszczona w buforze. Serwer SMTP co pewien czas sprawdza, czy są wiadomości do wysyłania. Proces przesyłania wiadomości przebiega następująco:

1. W pierwszej kolejnosci serwer SMTP nawiiazuje polaczenie TCP z serwerem docelowym. Jesli ten jest gotów do współpracy, przesyła do serwera SMTP komunikat o gotowosci. Jesli jednak docelowy serwer nie jest dostepny, to otrzymamy informujacy o tym komunikat. Gdy serwer docelowy nie jest przymocowany do Internetu lub intranetu, wtedy serwer SMTP ponawia próby polaczenia z serwerem docelowym az do uplywu wyznaczonego dozwolonego czasu.
2. Jesli serwer docelowy jest dostepny, serwer SMTP wysyla do niego polecenie HELO. W odpowiedzi na nie serwer docelowy zwraca swoja nazwe domeny. Serwer SMTP uzywa tej nazwy do zweryfikowania, czy nawiagal polaczenie z wlasciwym serwerem docelowym.
3. Serwer SMTP rozpoczyna transakcje wiadomosci przez wyslanie polecenia MAIL do serwera docelowego. Ewentualne bledy sa zgłoszane z wykorzystaniem wstecznej trasy, zawartej w poleceniu MAIL.
4. Serwer SMTP wysyla teraz polecenie DATA, które powiadamia serwer docelowy o nastepujacej po nim wiadomosci.
5. Serwer SMTP za pomoca funkcji send wysyla wiadomosc do jednego lub kilku adresatow. Funkcja send przyjmuje adres poczty elektronicznej i wiadomosc jako swoje argumenty. Serwer docelowy potwierdza otrzymanie wiadomosci komunikatem OK (albo w razie problemow wysyla komunikat o bledzie) do serwera SMTP.

Wiadomosc jest skladowana w serwerze SMTP az do przeslania do serwera docelowego. SMTP opiera sie na dostawach dwupunktowych (*end-to-end delivery*), w których laczy sie z serwerem docelowym w celu przeslania wiadomosci. Jesli uzytkownik-adresat jest niedostepny, to poczta zostaje odeslana do nadawcy.

POP

Post Office Protocol jest protokolem internetowym, sluzacym do przesyłania wiadomosci poczty elektronicznej z serwera POP do skrzynki pocztowej uzytkownika w lokalnym komputerze. Ten protokol funkcjonuje w architekturze klient-serwer. Serwer POP uzywa tego protokolu, natomiast lokalny komputer z programem pocztowym musi zostac skonfigurowany jako klient POP. Najnowsza wersja tego protokolu jest POP3; wczesniejsza — POP2 — jest juz wycofana. Protokol POP2 wymagal do wysylania wiadomosci serwera SMTP, natomiast POP3 moze funkcjonowac zarówno z serwerem, jak i bez niego. Ponadto protokol POP3 nie jest zgodny z POP2. POP2 uzywal portu TCP 109, zas POP3 uzywa portu 110.



Protokol POP3 jest opisany w RFC 2449.

Aby przeslac wiadomosci z serwera POP do lokalnej skrzynki pocztowej, musi zostac nawiiazana sesja POP, w nastepujacy sposob:

1. Klient POP nawiiazuje polaczenie TCP z serwerem POP.

2. Po nawiązaniu połączenia serwer POP wysyła komunikat do klienta POP. W tej chwili sesja wchodzi w *stan uwierzytelniania*. W tym stanie klient musi podać nazwę użytkownika i hasło, aby uwierzytelnic się w serwerze POP.
3. Serwer POP uwierzytelnia klienta, jeśli nazwa użytkownika i hasło są poprawne. Sesja wchodzi w *stan transakcji*. Następnie klient wysyła polecenia do serwera POP, aby odebrać wiadomość e-mail.
4. Po przeniesieniu wiadomości do skrzynki pocztowej w lokalnym komputerze, klient wysyła polecenie QUIT kończące sesję. W tym momencie sesja wchodzi w *stan aktualizacji*.
5. Wiadomość e-mail dociera do skrzynki pocztowej w lokalnym komputerze, skąd może zostać odczytana przez użytkownika.

IMAP

Internet Mail Access Protocol (IMAP) jest protokołem internetowym, który umożliwia dostęp do wiadomości e-mail składowanych w serwerze pocztowym w taki sposób, jakby były zapisane w lokalnym komputerze. Jest to cenna możliwość, ponieważ pozwala użytkownikom mobilnym sprawdzac poczta z domu, z pracy lub dowolnego innego miejsca o dowolnej porze. IMAP pozwala również użytkownikom modyfikować wiadomości zapisane w folderze w serwerze pocztowym. Protokół IMAP jest zgodny ze standardami internetowymi, takimi jak MIME. Protokół IMAP używa portu 143 TCP, a jego najnowsza wersja jest IMAP4.



Protokół IMAP został opisany w RFC 2060.

IMAP funkcjonuje w architekturze klient-serwer. Sesja IMAP może się znajdować w jednym z następujących stanów:

- ♦ *Stan nie uwierzytelniony* — sesja wchodzi w ten stan w chwili rozpoczęcia połączenia pomiędzy klientem i serwerem. Klient, zanim będzie mógł wydawać polecenia, musi dostarczyć informacje uwierzytelniające. Jeśli jednak połączenie jest z góry uwierzytelnione, sesja wchodzi bezpośrednio w stan uwierzytelniony.
- ♦ *Stan uwierzytelniony* — sesja wchodzi w ten stan po uwierzytelnieniu klienta; po nim klient musi wybrać skrzynkę pocztową, do której chce uzyskać dostęp.
- ♦ *Stan wybrany* — sesja wchodzi w ten stan po wybraniu przez użytkownika odpowiedniej skrzynki.
- ♦ *Stan wylogowania* — sesja wchodzi w ten stan po przerwaniu połączenia przez serwer. Serwer kończy połączenie na życzenie klienta lub po upływie czasu oczekiwania dla połączenia.

Protokół IMAP4 jest podobny do POP3, lecz istnieją pomiędzy nimi pewne znaczace różnice. Podczas gdy POP3 nadaje się idealnie do korzystania z wiadomości e-mail z pojedynczego komputera w trybie offline, IMAP4 daje użytkownikom dostęp do wiadomości nowych i zapisanych z różnych komputerów. Ponadto IMAP4 posiada dodat-

kowe funkcje, nieobecne w POP3, na przyklad wyszukiwanie slów kluczowych. Moze my wyszukiwac slowa kluczowe we wiadomosciach wciaz przebywajacych w serwerze pocztowym. Na podstawie wynikow wyszukiwania mozemy zdecydowac, ktore wiadomosci chcemy sciagnac do wlasnego komputera. Co wiecej, IMAP4 obsluguje trzy tryby dostepu do wiadomosci: offline, online i odlaczony:

- ◆ *Tryb offline* — wiadomosci pobrane z serwera pocztowego do komputera uzytkownika sa z serwera pocztowego usuwane.
- ◆ *Tryb online* — wiadomosci pozostaja w serwerze pocztowym. Uzytkownik moze czytac wiadomosci i manipulowac nimi za pomocą programu pocztowego.
- ◆ *Tryb odlaczony* — po polaczeniu z serwerem pocztowym program pocztowy tworzy kopie podreczna wybranych wiadomosci przed rozlaczeniem. Uzytkownik moze dokonywac manipulacji na tych buforowanych wiadomosciach, natomiast póżniej, po polaczeniu programu pocztowego z serwerem, odbywa sie resynchronizacja wiadomosci. W tym trybie wiadomosci pozostaja w serwerze.

IMAP4 pozwala na zdalne zarzadzanie folderami. Uzytkownicy moga tworzyk wielopoziomowe foldery w serwerze pocztowym i zarzadzac nimi, jesli zostana do tego upowaznieni przez administratora. IMAP4 pozwala również wielu uzytkownikom korzystac ze wspólnej skrzynki pocztowej równoczesnie z różnych miejsc. Duza zaleta protokolu IMAP4 jest zdolosc do oddzielenia dolaczonych plikow od tekstu lub nagłówka wiadomosci. Daje to uzytkownikom mozliwosc pobrania jedynie określonej czesci wiadomosci. Poza wymienionymi funkcjami IMAP4 pozwala również na dostep do informacji innych niz poczta elektroniczna, na przyklad NetNews i innych dokumentow.

Czytanie poczty

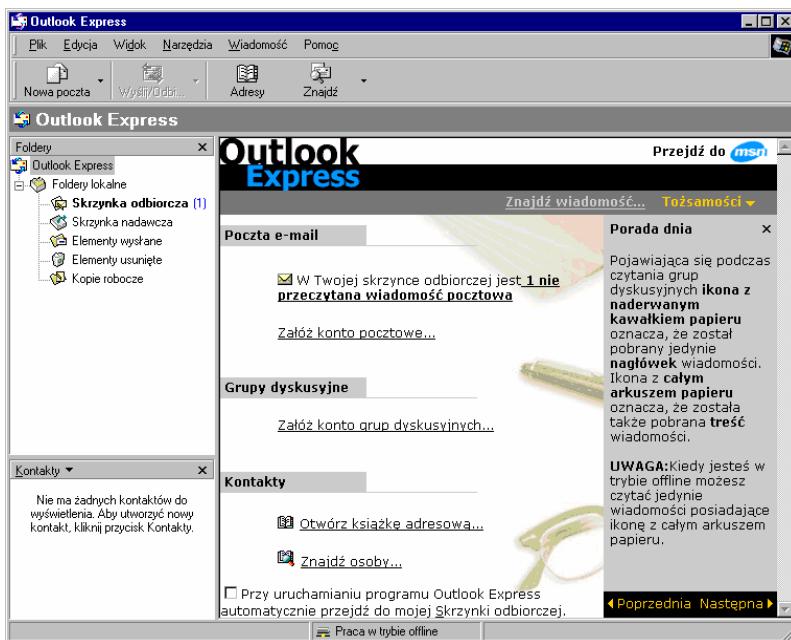
Aby móc czytac wiadomosci e-mail, potrzebne jest nam polaczenie z Internetem i program kliencki poczty elektronicznej. Dostepne sa różnorodne programy pocztowe, z których najbardziej popularnymi sa Microsoft Outlook Express i Netscape Messenger. Wprawdzie otwieranie i czytanie poczty moze różnic sie nieco w zaleznosci od programu, lecz podstawowy proces pozostaje niezmieniony. W wiekszosci klienckich programow pocztowych wiadomosci zapisywane sa w folderze o nazwie *Inbox (Skrzynka odbiorcza)*. Z tej skrzynki wybieramy i otwieramy wiadomosci, aby je przeczytac.

Microsoft Outlook Express

Microsoft Outlook Express jest jednym z najpowszechniej uzywanych klientów poczty elektronicznej dla systemu Windows. Sluzy do tworzenia, wysylania i odbierania wiadomosci. Okno Microsoft Outlook Express jest domyslnie podzielone na dwa panele. Lewy panel zawiera drzewo folderow, natomiast prawy panel przedstawia zawartosc zaznaczonego folderu, jak na rysunku 16.2. Domyslnie wszystkie odbierane wiadomosci sa umieszczone w skrzynce odbiorczej. Aby odebrac nowe wiadomosci, nalezy kliknac przycisk *Wyslij i odbierz* na pasku narzedzi okna Microsoft Outlook Express. Otwory sie okno dialogowe Outlook Express, przedstawiajace stan wiadomosci wyslanych i odebranych. Po odebraniu wszystkich wiadomosci okno sie zamknie. Aby odczytac wiadomosci:

Rysunek 16.2.

Okno Microsoft
Outlook Express



1. Kliknij ikone *Skrzynka odbiorcza* z listy folderów w lewym panelu. W prawym panelu pojawi się lista wiadomości. Wiadomości nie przeczytane są wyświetlane domyślnie czcionka wytluszczone.
2. Aby otworzyć nową wiadomość, kliknij ją dwukrotnie. Otworzy się nowe okno, zawierające wiadomość.

Netscape Messenger

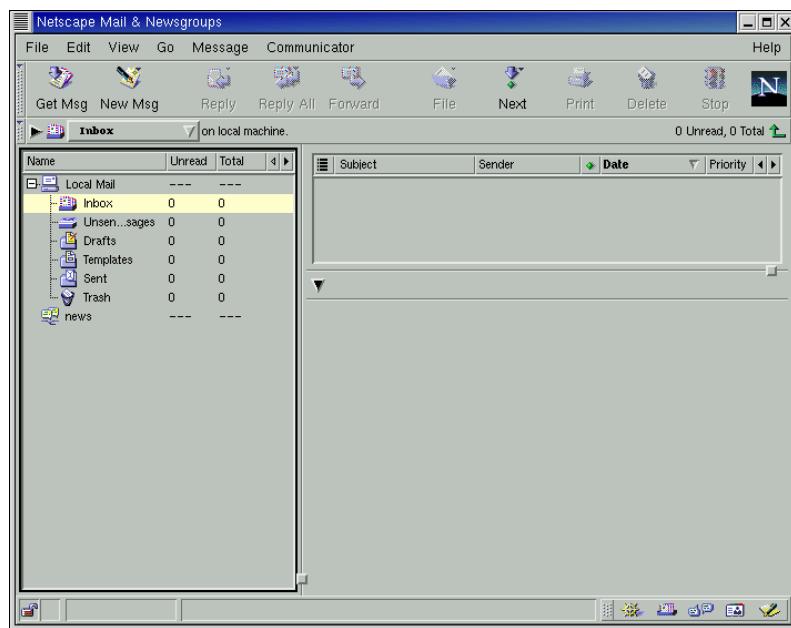
Netscape Messenger jest kolejnym powszechnie używanym programem pocztowym, który służy do tworzenia, wysyłania i czytania wiadomości e-mail. Okno Netscape Messenger domyślnie zawiera trzy panele, jak na rysunku 16.3. Lewy panel zawiera listę wszystkich dostępnych folderów (*Folder list*). Lista wiadomości (*Message list*) zajmuje prawy górny panel i przedstawia listę wiadomości zawartych w folderze, który jest aktualnie zaznaczony w lewym panelu. Prawy dolny panel (*Message body*) zawiera treść wiadomości aktualnie zaznaczonej na liście wiadomości w prawym górnym panelu.

Aby odebrać i przeczytać wiadomości:

1. Kliknij przycisk *Get Msg* (pobierz wiadomości) na pasku narzędzi Netscape Messenger.
2. Program pocztowy zapyta teraz o hasło. Jeśli hasło zostanie wprowadzone poprawnie, nowe wiadomości zostaną pobrane z serwera POP do skrzynki odbiorczej (*Inbox*).
3. Kliknij ikonę *Inbox* w lewym panelu. Zostanie wyświetlona lista wiadomości w panelu *Message list*. Nie przeczytane wiadomości wyświetlane są czcionką wytluszczone.

Rysunek 16.3.

Okno programu
Netscape Messenger



4. Aby otworzyac i przeczytac wiadomosc, kliknij ja w panelu *Message list*. Tresc wiadomosci pojawi sie w prawym dolnym panelu.

MIME i S/MIME

Poczta internetowa odniosla sukces dzieki standardowemu mechanizmowi, który Siec udostepnia do wymiany wiadomosci. Standardy internetowe (RFC 822) — podstawy formatu e-mail — zostaly po raz pierwszy zdefiniowane w 1982 r. Standardy te pozwalaly jednakze tylko na znaki ASCII w tresci listu i ograniczaly jego dlugosc do ok. 1000 znakow. Z uplywem czasu pojawiла sie potrzeba standardu bardziej elastycznego i otwartego na rozszerzenia.

MIME (Multipurpose Internet Mail Extensions — rozszerzenia poczty internetowej do wielu zastosowan) jest standardem opracowanym przez IETF w roku 1992. MIME definiuje, jak wiadomosci poczty elektronicznej powinny byc formatowane, aby mozna bylo je przesyłac pomiedzy różnymi systemami poczty elektronicznej. MIME pozwala na umieszczenie w listach e-mail tekstu, obrazów, dzwieku, wideo, plików skojarzonych z aplikacjami i wiadomosci multimedialnych. MIME posiada nastepujace właściwości:

- ◆ Jasno definiuje zbiór dozwolonych typów zawartosci (*content-type*). Oznaczaja one rodzaj danych, jakie mozna wstawic do tresci listu. Do typów danych, jakie mozemy umiescic w liscie naleza: tekst (*text*), obrazy (*image*), wiadomosci (*message*), dzwiek (*audio*), wideo (*video*), dane wieloczesiowe (*multipart*) i aplikacje (*application*).
- ◆ Umożliwia kodowanie danych w innych standardach niz ASCII.
- ◆ Obsluguje typ zawartosci multipart (wieloczesiowy), który pozwala na umieszczenie wiecej niz jednej tresci (*body*) w pojedynczej wiadomosci.

Standard MIME obejmuje następujące pola nagłówka, definiujące określona wiadomość:

- ♦ *MIME-version* — to pole nagłówka zawiera numer wersji deklarującej zgodność wiadomości ze standardem MIME. Za pomocą tego numeru agenty przetwarzania poczty mogą zidentyfikować wiadomości zgodne ze standardem MIME.
- ♦ *Content-type* — określa typ danych w treści listu oraz podtyp definiujący format określonego typu danych.
- ♦ *Content-transfer-encoding* — określa, jak zawartość wiadomości jest zakodowana. Pozwala to przesyłać wiadomości za pomocą mechanizmów transportu poczty, ograniczających zbiór znaków lub typ danych.
- ♦ *Content-ID* i *Content-Description* (identyfikator i opis zawartości) — te pola dokładniej identyfikują i opisują dane zawarte w wiadomości.

Gdy poczta jest przesyłana przez Internet, najważniejszym zagadnieniem jest jej bezpieczeństwo. S/MIME (*Secure/Multipurpose Internet Mail Extensions*) jest standardem internetowym, który zapewnia bezpieczeństwo wiadomości w formacie MIME. S/MIME jest rozszerzeniem MIME o nowe typy zawartości (zaszyfrowana i wydana kluczy), pozwalającym na szyfrowanie. S/MIME służy do zabezpieczania elementów MIME — na przykład nagłówka lub treści MIME. Dwa dowolne pakiety oprogramowania obsługujące MIME mogą komunikować się bezpiecznie w Internecie. S/MIME udostępnia kryptograficzne funkcje zabezpieczeń dla poczty elektronicznej, na przykład poufnosc i autentyczność. Ponizsze podpunkty omawiają sposoby, w jakie S/MIME zapewnia bezpieczeństwo.

Poufnosc

Szyfrowanie wiadomości e-mail zapewnia poufnosc przesyłanych informacji. W S/MIME stosowane jest szyfrowanie asymetryczne, do którego generowana jest para kluczy: *klucz publiczny* i *klucz prywatny*. Dla każdego użytkownika para kluczy jest unikatowa. Aby zaszyfrować wiadomość, tworzony jest losowy klucz symetryczny. Gdy zaszyfrowana wiadomość dociera do adresata, musi zostać odszyfrowana. W tym celu klucz symetryczny musi zostać wysłany również do odbiorcy w postaci zaszyfrowanej. Do jego zaszyfrowania użyty zostaje klucz publiczny odbiorcy. Zaszyfrowany klucz symetryczny jest przesyłany do adresata przed zaszyfrowaniem wiadomości. Wiadomość zostaje odszyfrowana po stronie odbiorcy za pomocą klucza symetrycznego, który z kolei zostaje przez odbiorcę odszyfrowany za pomocą jego klucza prywatnego.

Informacje o kluczu publicznym zostają umieszczone podczas generowania pary w *certyfikacie cyfrowym*, który może być dodany do dokumentu lub wiadomości e-mail, co gwarantuje jej autentyczność i pozwala na bezpieczne szyfrowanie. Certyfikat cyfrowy zawiera dodatkowo informacje o właścicielu klucza publicznego (na przykład nazwę użytkownika właściciela). Gdy protokół S/MIME jest używany dla klienta poczty elektronicznej, dla użytkownika generowane są para kluczy i certyfikat cyfrowy. Aby w przyszłości uzyskać dostęp do pary kluczy, użytkownik musi podać hasło.

Certyfikaty cyfrowe można otrzymać od różnych urzędów certyfikacji, na przykład VeriSign lub Thawte Corporation. Aby otrzymać podpis cyfrowy w programie Microsoft Outlook Express:

1. Wybierz *Narzedzia/Opcje*, aby otworzyc okno dialogowe *Opcje*.
2. Wybierz zakładke *Zabezpieczenia*.
3. Kliknij przycisk *Pobierz identyfikator cyfrowy*, co uruchomi przeglądarkę i otworzy strone z adresami różnych urzędów certyfikacji, udostepniajacych certyfikaty cyfrowe.
4. Z listy dostepnych urzędów wybierz odpowiedniego dostawce. Zostanie otwarta witryna WWW dostawcy. Wypelnij formularz zgloszeniowy.
5. Po wyslaniu formularza zostanie wygenerowany list weryfujacy i wyslany pod adres e-mail podany w formularzu zgloszeniowym. Wiadomosc ta zawiera instrukcje, jak zainstalowac certyfikat, numer PIN i adres WWW sluzacy do zatwierdzenia i zainstalowania certyfikatu.
6. Przejdz do strony WWW podanej w wiadomosci weryfujacej, podaj numer PIN i zainstaluj certyfikat cyfrowy zgodnie z instrukcjami dostawcy.

Po otrzymaniu certyfikatu cyfrowego nalezy skojarzyc go z kontem pocztowym. W tym celu:

1. Wybierz *Narzedzia/Konta*, aby otworzyc okno dialogowe *Konta internetowe*.
2. Wybierz konto, z którym chcesz skojarzyc certyfikat cyfrowy, i kliknij przycisk *Wlasciwosci*. Otworzy sie okno dialogowe *Wlasciwosci* dla wybranego konta.
3. Kliknij zakładke *Zabezpieczenia*.
4. Wybierz opcje *Uzyj identyfikatora cyfrowego przy wysylaniu bezpiecznych wiadomosci z:*.
5. Kliknij *Identityfikator cyfrowy* i wybierz domyslny identyfikator dla konta.

Po skojarzeniu certyfikatu cyfrowego z kontem mozna go uzywac do wysylania bezpiecznych wiadomosci.

Mozemy zaszyfrowac wysylane wiadomosci tak, by jedynie zamierzony odbiorca mogle odczytac. Aby zaszyfrowac wiadomosc wysylana z programu Outlook Express, musimy znac certyfikat cyfrowy odbiorcy. Aby dodac do ksiazki adresowej certyfikat cyfrowy z wiadomosci podpisanej cyfrowo:

1. Otwórz podpisana cyfrowo wiadomosc. Wybierz *Plik/Wlasciwosci*, aby otworzyc okno dialogowe *Wlasciwosci*.
2. Wybierz zakładke *Zabezpieczenia*.
3. Kliknij przycisk *Dodaj identityfikator cyfrowy do ksiazki adresowej*.

Trzeba jeszcze ustawić relacje zaufania. Oznaczaja one, ze nasze konto jest zaufane i mozemy uzywac certyfikatu cyfrowego odbiorcy. Po otrzymaniu certyfikatu cyfrowego odbiorcy i ustanowieniu relacji zaufania, mozemy szyfrowac wysylane wiadomosci w dwojaki sposob. Pierwszy polega na skonfigurowaniu opcji zabezpieczen w programie Outlook Express tak, by wysylane wiadomosci byly automatycznie szyfrowane. Druga metoda szyfrowania polega na kliknieciu przycisku *Szyfruj* przed wyslaniem wiadomosci.

Autentyczosc

Autentyczosc zapewniaja podpisy cyfrowe, których mozemy uzywac do podpisywania wiadomosci. Podpis cyfrowy jest ciagiem bitow, generowanym dla kazdej wiadomosci i sluzacym do potwierdzenia tozsamosci nadawcy i integralnosci wyslanej wiadomosci. Gdy adresat otrzymuje wiadomosc, razem z nia odbiera informacje o nadawcy i o tym, czy wiadomosc nie ulegla zmianom w procesie przesyłania. Dzieki temu S/MIME zapewnia bezpieczenstwo, autentyczosc wiadomosci przesyłanych przez Internet — to główne powody, dla których S/MIME został zaadaptowany jako standard przez wszystkie liczace sie programy pocztowe. Niektóre produkty z dziedziny poczty elektronicznej, na przykład Microsoft Outlook Express 4.0 i nowsze, Microsoft Outlook 2000 i Netscape Communicator od wersji 4.03 posiadaja wbudowana obsługę S/MIME.

Wprawdzie poczta elektroniczna jest szybsza i wygodniejsza od tradycyjnej, lecz jest też mniej bezpieczna, poniewaz hakerzy mogą uzyskać do niej dostęp. Jednak, jak już wcześniej powiedzieliśmy, można zapewnić bezpieczeństwo poprzez szyfrowanie wysłanej poczty. Jednym z dostępnych programowych systemów szyfrowania jest Pretty Good Privacy, omówiony w następnym punkcie.

PGP

Pretty Good Privacy (PGP) jest programem szyfrującym, którego możemy używać do zabezpieczania listów e-mail i załączników. Moduły rozszerzające PGP są dostępne dla różnych programów pocztowych, na przykład Microsoft Outlook Express, Eudora i Microsoft Outlook. Moduły te instalują się w programie pocztowym i pojawiają się w opcjach menu lub jako przyciski w oknie programu pocztowego.

Podczas instalacji PGP generowana jest para kluczy (publiczny i prywatny), a następnie PGP publikuje klucz publiczny użytkownika w serwerze kluczy PGP. Klucz ten jest powszechnie znany i służy do szyfrowania wiadomości wysyłanych do danego użytkownika. Procesy szyfrowania i deszyfrowania w PGP i S/MIME są takie same. Jedyna różnica polega na tym, że S/MIME do przesłania klucza publicznego używa certyfikatu, natomiast PGP używa własnego formatu do jego składowania. Nawet stosowane algorytmy są takie same. Prywatnego klucza nie wolno ujawniać, ponieważ zapewnia on, że tylko prawowity właściciel będzie w stanie odszyfrować i odczytać odebrane wiadomości. PGP pozwala również podpisywać cyfrowo wysłane listy elektroniczne, aby zapewnić ich autentyczność. Podpis cyfrowy dla każdego dokumentu i użytkownika jest unikatowy.



Podpis cyfrowy jest unikatowy dla każdego dokumentu wysłanego przez użytkownika. Dwa różne dokumenty wysłane przez jednego użytkownika nie będą miały takiego samego podpisu.

Grupy dyskusyjne — wprowadzenie

Internet jest jednym z największych źródeł informacji. Możemy rozpowszechniać informacje w Internecie w dwóch sposobach: przez *listy dyskusyjne (mailing list)* i *grupy dyskusyjne (newsgroup)*. Lista dyskusyjna to grupa osób wysyłających informacje za

pomocą poczty elektronicznej. Kopia informacji jest wysyłana do wszystkich członków grupy. Listy tego typu pozwalają szybko i wygodnie komunikować się z grupą osób za pomocą pojedynczego adresu (adresu grupy) zamiast podawania adresów wszystkich uczestników grupy. Ponieważ jednak wysłanie osobnego listu do każdego członka wymaga sporej przepustowości łączna, ta metoda udostępniania informacji może być kłopotliwa. Lepsza metoda jest korzystanie z *grup dyskusyjnych*. Grupy dyskusyjne oznaczają wiadomości dostępne dla dużej liczby użytkowników Internetu, lecz nie wymagające wysyłania osobnej kopii informacji do każdego użytkownika.

Idea grup dyskusyjnych po raz pierwszy pojawiła się w roku 1979, gdy publikacja została rozieszana pomiędzy Duke i University of North Carolina. Wiadomości internetowe są podzielone na grupy dyskusyjne (*newsgroups*), czyli obszary tematyczne. Dostępnych jest mnóstwo tematów grup dyskusyjnych, od astrologii do zoologii. W zasadzie dla każdego tematu, o którym możemy pomyśleć, jest dostępna grupa dyskusyjna. Artykuły do każdej grupy pisane są przez zainteresowanych tematem subskrybentów. Artykuły te są następnie publikowane w grupie dyskusyjnej, gdzie inne osoby mogą je czytać i odpowiadać na nie. Inaczej mówiąc, grupa dyskusyjna stanowi forum online, gdzie użytkownicy mogą czytać artykuły opublikowane przez innych użytkowników oraz publikować wiadomości na ten temat. Cała grupa dyskusyjna zawiera wiadomości o najświeższych zdarzeniach, lecz większość została utworzona dla określonej tematyki. Na przykład, Usenet jest siecią użytkowników, składającą się z wielu serwerów mieszczących tysiące grup dyskusyjnych.

Każda grupa dyskusyjna ma swoje, zależne od dostępności, zaklasyfikowane jako zamknieta, moderowana lub nie moderowana:

- ◆ Grupy zamknięte nie są dostępne dla ogólnego. Aby przyląć się do zamkniętej grupy, musimy wysłać proszę do administratora grupy dyskusyjnej.
- ◆ Grupy moderowane wymagają zaaprobowania przez moderatora każdego zgłoszonego artykułu przed jego opublikowaniem.
- ◆ Grupy nie moderowane są otwarte dla ogólnego użytkownika.

Kategorie grup dyskusyjnych są zorganizowane w hierarchię strukturalną od kategorii ogólnej do szczegółowej. Nazwa każdej grupy daje dość dokładne pojęcie o jej tematyce. Każda nazwa składa się z tematu i podtematów oddzielonych kropkami. Na przykład, w nazwie grupy dyskusyjnej sci.med.nutrition głównym tematem jest nauka (science), podtematem nauki jest medycyna (medicine), a jej podtematem jest żywienie (nutrition) — rysunek 16.4.

Rysunek 16.4.
Hierarchia grup dyskusyjnych

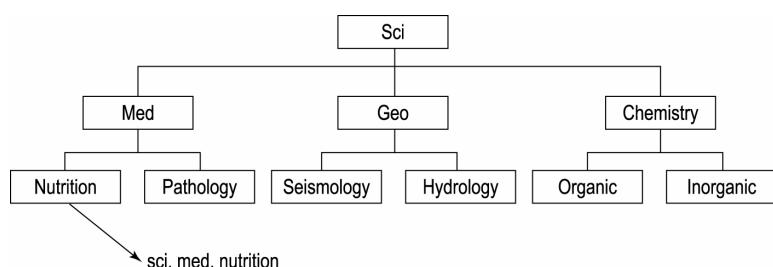


Tabela 16.1 przedstawia część prefiksów nazw grup dyskusyjnych i skojarzone z nimi tematy.

Tabela 16.1. Podstawowe grupy dyskusyjne

Prefiks	Tematyka
<i>comp</i>	Tematy związane z komputerami.
<i>humanities</i>	Sztuka i tematy humanistyczne.
<i>news</i>	Tematy związane z grupami dyskusyjnymi i ich oprogramowaniem.
<i>rec</i>	Rekreacja.
<i>sci</i>	Nauka i badania.
<i>soc</i>	Zagadnienia socjalne i podobne.
<i>talk</i>	Zagadnienia dyskusyjne.
<i>misc</i>	Różne tematy.

Dostęp do grupy dyskusyjnej można uzyskać przez jej subskrypcje. W programie pocztowym Microsoft Outlook Express można zapisać się do grupy po skonfigurowaniu serwera grup dyskusyjnych. Podczas przeglądania grup, do których nie jesteśmy zapisani, możemy subskrybować dowolną z nich przez kliknięcie przycisku *Subskrybuj* w oknie *Subskrypcje grup dyskusyjnych*. W programie Netscape Messenger możemy subskrybować dyskusyjną następująco:

1. W oknie Netscape Messenger kliknij ikone *Newsgroup* na pasku składników.
2. Z menu *File (Plik)* wybierz *Subscribe*. Otworzy się okno dialogowe. Wprowadź odpowiednie informacje i kliknij przycisk *OK*. Subskrybowana grupa pojawi się na liście subskrypcji.

Mozemy również zazadac utworzenia nowej grupy dyskusyjnej na określony temat. Aby założyć nową grupę dyskusyjną w Usenetie:

1. Zgłos propozycję nowej grupy dyskusyjnej do grupy *news.groups*. Propozycja taka nosi nazwę *Request for Discussion*.
2. Dla proponowanej grupy dyskusyjnej utworzona zostaje próbna grupa użytkowników, zas próbny moderator monitoruje czytelników grupy.
3. Po pięciu miesiącach proponowana grupa dyskusyjna zostaje zaakceptowana, jeśli zmieszcza się w czolowych 75% grup dyskusyjnych Usenet. W przeciwnym razie jest odrzucana.
4. Proponowana grupa dyskusyjna otrzymuje nową nazwę opartą na sugestiach próbnych czytelników i moderatora oraz zostaje przeniesiona do Usenetu.

Serwery i koncentratory

Sieciowe grupy dyskusyjne korzystają z architektury klient-serwer. Serwer pełni funkcje centralnego magazynu, w którym składowane są wszystkie artykuly grup dyskusyj-

nych. Serwer ten udostepnia informacje dla pozostałych komputerów w sieci. Klient musi subskrybowac grupy dyskusyjne obecne w serwerze. Aby uzyskac dostep do grupy dyskusyjnej, klient musi dysponowac programem klienckim, tzw. przegladarka grup dyskusyjnych, na przyklad Microsoft Outlook Express. Podczas konfigurowania przegladarki grup dyskusyjnych musimy wyznaczyc dla niej serwer. Program przegladarki grup dyskusyjnych jest również wymagany, gdy uzytkownik chce wyslac artykul lub odpowiedz. Przesylem artykulów z serwera do klienta zajmuje sie protokół NNTP (*Network News Transfer Protocol*).

NNTP

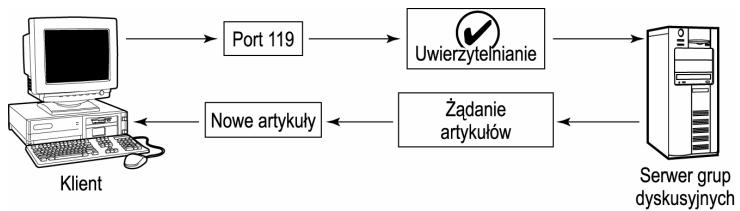
Protokół NNTP jest czescia pakietu TCP/IP i definiuje reguly dystrybucji, odbierania i publikowania artykułów w internetowych grupach dyskusyjnych. Protokół ten, uzywajacy portu 119, definiuje standardy dystrybucji i publikacji artykułów za pomoca modelu potokowego klient-serwer oraz umozliwia dostep uzytkowników do artykułów skladowanych w centralnym serwerze. NNTP udostepnia ponadto polecenia, które zajmuja sie wymiana artykułów pomiedzy serwerami grup dyskusyjnych, a takze interaktywny mechanizm przesyłania wiadomosci pomiedzy serwerami. Serwer, który chce wysylac lub odbierac nowe artykuły, laczy sie z innym serwerem za pomoca NNTP. Pliki z artykułami zostaja porównane i wszelkie zmiany, których nalezy dokonac, sa dodawane do tych plików. W ten sposob tylko unikatowe artykuły sa dodawane — nie zachodzi powielanie artykułów w serwerach grup dyskusyjnych.

Omówimy teraz szczegółowo proces przesyłania wiadomosci. Gdy klient chce skorzystac z dostępu do grupy dyskusyjnej:

1. Przegladarka grup dyskusyjnych w komputerze klienckim laczy sie z serwerem za pomoca NNTP.
2. Po nawiązaniu polaczenia przegladarka pobiera artykuły opublikowane w subskrybowanych grupach dyskusyjnych. Klient moze teraz czytac artykuły.
3. Jesli uzytkownik chce odpowiedziec na artykuł, pisze i wysyla odpowiedz.
4. Przegladarka grup dyskusyjnych wysyla te odpowiedz do serwera grup dyskusyjnych za pomoca NNTP.
5. Serwer zapisuje odpowiedz na koncu pliku grupy dyskusyjnej, który jest duzym plikiem tekstowym.
6. Serwer grup dyskusyjnych laczy sie z innym serwerem za pomoca NNTP i przesyła dolaczony plik grupy dyskusyjnej. Serwer porównuje ten plik z własnym i jesli znajdzie zmiany, dodaje je do własnego pliku grupy dyskusyjnej.
7. Proces trwa, dopóki wszystkie grupy dyskusyjne nie otrzymaja zaktualizowanych danych. Odpowiedz klienta jest teraz widoczna dla wszystkich uzytkowników grupy dyskusyjnej.

Proces przesyłania wiadomosci został przedstawiony na rysunku 16.5.

Rysunek 16.5.
Proces przesyłania
wiadomości
z grup dyskusyjnych



Netykieta

Wszyscy stosujemy się do niepisanych zasad etykiety, gdy komunikujemy się z innymi. Porozumiewanie się przez Internet nie stanowi wyjątku. Słowo *netykieta* (*netiquette*) jest połączeniem dwóch wyrazów: *net* i *etiquette* i oznacza zbiór zasad, których powinnismy się trzymać podczas korzystania z Internetu. Kilka z tych zasad przedstawiliśmy ponizej:

1. Nie pisz całych wiadomości DUZYMI LITERAMI. Jest to tekstowy odpowiednik krzyku.
2. Przed wysłaniem pytania do grupy dyskusyjnej sprawdź, czy odpowiedź nie jest już obecna na liście FAQ (*Frequently Asked Questions* — najczęściej zadawane pytania).
3. Używaj tzw. „grymasów” (*emoticon*) do wyrażania swoich emocji. Sa to symbole oznaczające wyraz twarzy lub emocje, których nie można zawrzeć w zwykłym tekscie. Grymasły składają się z różnych kombinacji znaków, przedstawiających schematyczny obraz twarzy obrócony o 90 stopni — na przykład, usmiechnięta twarz składa się z dwukropka i prawego nawiasu: :).
4. Sprawdź ton wiadomości przed jej wysłaniem.
5. Wysyłaj wiadomości zwiezle i na temat.
6. Wypełnij pole tematu przed wysłaniem wiadomości.
7. Nie wysyłaj listów lancuskowych.
8. Pamiętaj, by podpisywać wiadomości.

Rozdział 17.

Uslugi informacyjne

dla przedsiębiorstw

W tym rozdziale:

- ◆ Standard X.500
- ◆ Protokół LDAP
- ◆ Network Information Service (NIS)
- ◆ Network Information Service + (NIS+)
- ◆ Usluga katalogowa StreetTalk
- ◆ Novell Directory Service (NDS)
- ◆ Active Directory

Sieci komputerowe, a zwłaszcza największa z nich — Internet, zdominowały się na stałe we współczesnym życiu. Sieć składająca się z czterech komputerów, połączonych ze sobą dla szybszej komunikacji, rozwinała się w globalną bazę informacji. Wzrost liczby użytkowników podłączonych do sieci powoduje, że coraz bardziej trudno znaleźć informacje o nich. Co gorsza, informacje w sieci (na przykład numery telefonów i adresy poczty elektronicznej) mogą być trudne do wyszukiwania, jeśli są przechowywane w różnych elektronicznych książkach adresowych, ponieważ mogą one korzystać z różnych protokołów dostępu.

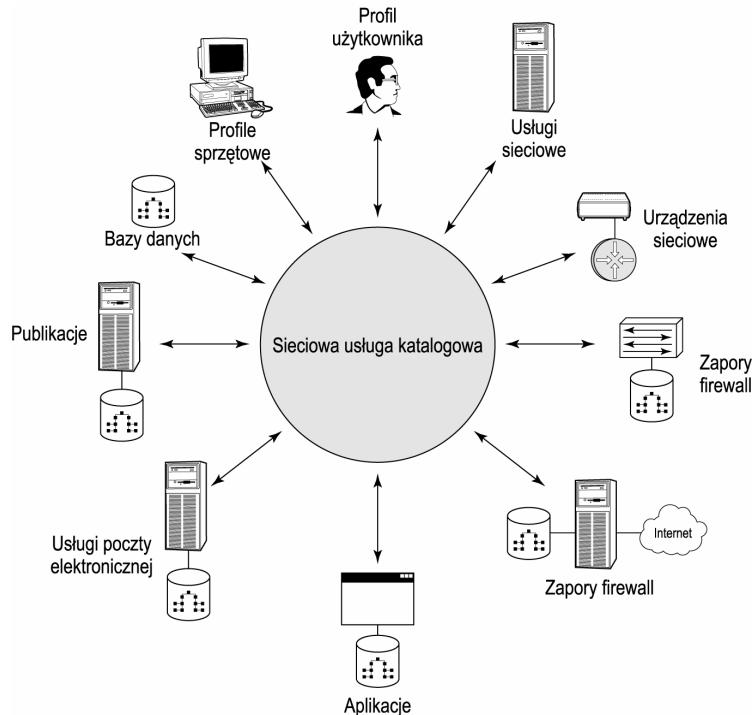
Wprowadzenie

do sieciowych usług katalogowych

Uslugi informacyjne dla przedsiębiorstw, czyli *sieciowe usługi katalogowe*, okazały się skutecznym rozwiązaniem tego problemu. Sieciowe usługi katalogowe (*Network Directory Service*) są jednymi z najważniejszych składników sieci przedsiębiorstwa lub sieci rozproszonych. Pozwalają one śledzić wszystkie nazwy, profile, adresy e-mail i adresy komputerów wszystkich użytkowników sieci. Oprócz udostępniania tych zasobów użytkownikom i aplikacjom, usługi katalogowe umożliwiają komunikację w sieci. Podstawowym zadaniem usług katalogowych jest nazywanie i znajdowanie zasobów sieciowych

i użytkowników; mogą one też przechowywać informacje o konfiguracji sieci. Ogólna idea sieciowej usługi katalogowej przedstawia rysunek 17.1.

Rysunek 17.1.
Sieciowe usługi katalogowe



Ogólnie mówiąc, usługa katalogowa kojarzy nazwy zasobów sieciowych z adresami IP. Odwzorowanie to powoduje, że użytkownik może znaleźć usługę podając jej nazwę. Ponieważ usługa katalogowa ukrywa fizyczną topografię sieci i protokoly przed użytkownikami, mogą oni korzystać z dowolnych zasobów nie wiedząc nic o ich położeniu i sposobie składowania. Aplikacje w sieci również mogą korzystać z usług katalogowych. Na przykład, aplikacja poczty elektronicznej może używać wyszukiwania w książce adresowej, aby znaleźć określonego użytkownika lub numer telefonu. Sieciowe usługi katalogowe do szybkiego wyszukiwania używają wysoce wyspecjalizowanych baz danych.

Do dodatkowych korzyści oferowanych przez sieciowe usługi katalogowe należą:

- ◆ Poprawa bezpieczeństwa sieci przez odmówienie dostępu intruzom i nieautoryzowanym użytkownikom.
- ◆ Rozłożenie informacji na wiele komputerów obecnych w sieci.
- ◆ Mechanizmy replikacji danych, które pozwalają wielu użytkownikom korzystać równocześnie z tych samych informacji i uodporniają sieć na awarie.
- ◆ Możliwość partycjonowania informacji, pozwalającej składować bardzo duże liczby obiektów w różnych serwerach.

X.500 był pierwszym krokiem w stronę sieciowych usług katalogowych, opracowany jako zestaw standardów i model informacji dla usługi katalogowej o globalnym dosta-

pie. Niezależność tego modelu od platformy przesadziła o jego sukcesie. Jednakże, podobnie jak wszelkie pionierskie rozwiązania, miał kilka wad, które doprowadziły do powstania protokołu LDAP (*Lightweight Directory Access Protocol* — uproszczony protokół dostępu do katalogu). Model X.500 był duży, złożony i nie nadawał się do środowiska komputerów osobistych, które z czasem stało się popularniejsze od środowisk dużych komputerów mainframe. Największymi atutami LDAP jest bezproblemowa obsługa TCP/IP i współpraca ze środowiskiem komputerów biurkowych.

W chwili obecnej w sieciach używane są różnorodne usługi katalogowe, do których zaliczają się *Network Information Service* (NIS), *Network Information Service+* (NIS+), *Banyan Street Talk Directory Service* (STDS), *Novell Directory Service* (NDS) i *Microsoft Active Directory* (AD). Najpowszechniej używanymi z nich są NDS i AD. Warto zwrócić uwagę, że większość z tych usług katalogowych opiera się na standardach LDAP i X.500 International Telecommunication Union (ITU).

Standard X.500

X.500 jest protokołem ITU (*International Telecommunication Union*), służącym do zarządzania sieciowymi katalogami użytkowników i zasobów. Ten sam standard został opublikowany przez Open Systems Interconnection (OSI) i IEC. Pierwsza specyfikacja X.500 została opublikowana w roku 1988. W 1993 r. została wydana bardziej rozwinięta wersja, zgodna z poprzednia.



Pierwsza działająca usługa katalogowa X.500 została uruchomiona przez SURFnet w 1992 r.

X.500 jest protokołem rozproszonym katalogu, który pozwala na katalogowanie hierarchii regionów, krajów, organizacji i poszczególnych osób. Inaczej mówiąc, X.500 pomaga organizjom na całym świecie utworzyć globalny elektroniczny katalog użytkowników, dostępny przez Internet. Ten katalog elektroniczny nazywany jest również *globalną książką teleadresową* i pozwala użytkownikowi w wygodny sposób szukać informacji o innych osobach. Wyszukiwanie może odbywać się według nazwiska, adresu poczty elektronicznej, numeru telefonu, organizacji itp. Jeśli informacje o organizacji zostaną umieszczone w tej książce adresowej, stana się dostępne globalnie.

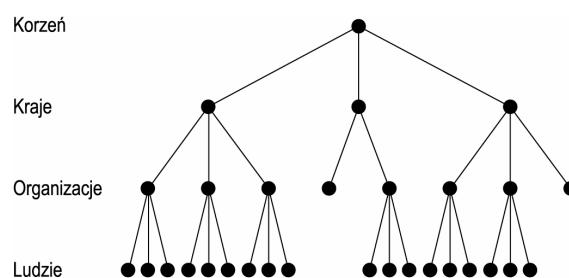
Protokół X.500 opiera się na modelu klient-serwer. Globalny katalog, stosowany przez X.500, nosi nazwę bazy danych informacji katalogowych (DIB — *Directory Information Database*) i jest potocznie nazywany *książką teleadresową* (*white pages*). Zgodnie z modelem informacji X.500, DIB jest wspólnie użytkowana przez agenty systemowe katalogu (DSA — *Directory System Agent*), czyli serwery X.500, które utrzymują lokalne informacje organizacji. DSA utrzymuje lokalna baza danych i może komunikować się z innymi DSA za pomocą protokołu DSP (*Directory System Protocol* — protokół systemu katalogowego). DSP jest protokołem należącym do zbioru zaleceń X.500. Do przeszukiwania serwerów X.500 służy agenci użytkowników katalogu (DUA — *Directory User Agent*).



DSA moze składowac informacje dla wiecej niz jednej organizacji, jesli ich rozmiary sa niewielkie. Gdy organizacja jest duza, jej dane katalogowe moga miec sie na wiecej niz jednym DSA (serwerze X.500). Rozklad informacji pomiedzy DSA jest calkowicie niewidoczny dla uzytkowników.

Wszystkie DSA w usludze katalogowej X.500 sa polaczone ze soba w wirtualne *drzewo informacji katalogowych* (DIT — *Directory Information Tree*), ktore jest hierarchiczna struktura danych. Cala struktura zaczyna sie od *korzenia* lub *wzla glownego* (*root*). Ponizej korzenia definiowane sa poszczegolne kraje, ktore z kolei dziela sie na organizacje. Organizacje moga dalej dzielic sie na jednostki organizacyjne lub indywidualne osoby. Uproszczona struktura DIT przedstawia rysunek 17.2.

Rysunek 17.2.
Uproszczona struktura drzewa informacji katalogowych X.500



Wszystkie informacje w katalogu X.500 sa przechowywane w postaci *wpisów* (*entry*), ktore naleza do *klas obiektów* (*class object*). Klasa obiektu moze byc kraj, organizacja, jednostka organizacyjna lub osoba. Wszystkie informacje zwiase z klasami obiektów sa przechowywane w postaci wpisów. Wpis stanowi zbiór atrybutów, definiujacych faktyczne informacje. Klasa obiektu, do której dany obiekt nalezy, definiuje atrybuty, ktore moze on posiadac. Na przyklad, klasa obiektu „kraj” pozwala na atrybuty „nazwa”, „kontynent” i tak dalej. Kazdy wpis jest identyfikowany przez *nazwe wyróżniajaca* (DN — *Distinguished Name*). Kazdy skladnik DN jest nazywany *wzgledna nazwa wyróżniajaca* (RDN — *Relative Distinguished Name*). Na przyklad, adres *skj@nst.co.be* reprezentuje nazwe DN, w ktorej atrybut ma nazwe *skj*, organizacja *nst*, a nazwa kraju *be*. Skladniki *skj*, *nst* i *be* sa nazwami RDN. Model X.500 pozwala również na *wpisy umowne* (*alias entry*), sluzace do tworzenia relacji innych niz hierarchiczne. Wpis umowny to wpis w krótkiej postaci. Model informacji X.500 jest przedstawiony na rysunku 17.3.

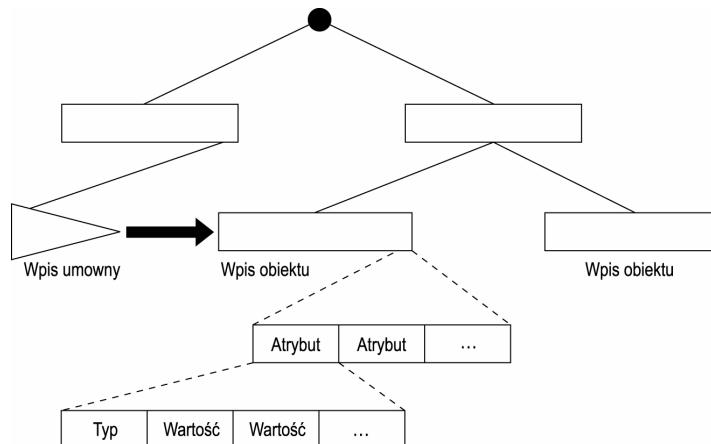


Atrybut *name* (nazwa), stosujacy sie do wszystkich klas atrybutów, musi posiadac unikatowa wartosc na poziomie, do którego nalezy. Na przyklad, jesli w organizacji pracuje dwóch Janów Kowalskich, ich wpisy w atrybucie *name* musza miec rózne wartosci.

Do wlosciowosci X.500 naleza replikacja, agent uzytkownika katalogu (DUA) i przeszukiwanie katalogu.

- ♦ *Replikacja* — podczas szukania informacji mozna znaczaco skrócić czas dostepu i poprawic jakosc uslugi (*Quality of Service*), jesli informacje zapisane w jednym DSA moga zostac skopowane (replikowane) do innych DSA. X.500 stosuje protokół DISP (*Directory Information Shadowing Protocol*), który pozwala na replikacje czesci DIT pomiedzy wezlami.

Rysunek 17.3.
Model informacji X.500



- ♦ *Directory User Agent (DUA)* — użytkownik za pomocą DUA uzyskuje dostęp do danych i pobiera zadane informacje. Inaczej mówiąc, DUA jest klientem X.500, który służy użytkownikom do pozyskiwania informacji z DIT. Gdy użytkownik szuka czegoś lub przegląda dane DIT, DUA kontaktuje się z najbliższym DSA w celu uzyskania zadanych informacji. Gdy te zostaną z powodzeniem zlokalizowane, DUA zwraca je do użytkownika. Interfejsy DUA dla usług książki teleadresowej są dostępne dla wszelkich typów platform: DOS, Windows, Macintosh, Unix i tak dalej. Do najpopularniejszych DUA zaliczają się whois i finger.
- ♦ *Kontrola dostępu* — użytkownik może odpytywać serwer X.500 (DSA) za pomocą DUA. Dostęp do katalogu (lub jego części) może być dozwolony lub zabroniony, zależnie od uprawnień użytkownika zadającego informacji.
- ♦ *Wyszukiwanie* — wyszukiwanie można przeprowadzać na dowolnym poziomie DIT, na podstawie typu atrybutu i jego wartości, podanych przez użytkownika. Katalog jest przeszukiwany w celu znalezienia wszelkich wpisów pasujących do wartości atrybutu. Jeśli na zapytanie użytkownika nie można odpowiedzieć lokalnie, zostaje ono przekazane do innych DSA. Po znalezieniu wyników zapytania, zostają one zwrócone do użytkownika. Cały proces jest niewidoczny dla użytkownika, który odnosi wrażenie, iż cały katalog jest dostępny z lokalnego DSA. Organizacje utrzymujące DSA mogą jednakże ograniczyć uprawnienia użytkownika do wyszukiwania.

Wersja X.500 z roku 1993 udostępnia wszystkie powyższe możliwości; wersja z roku 1998 nie posiadała mechanizmu kontroli dostępu i usług replikacji.



Wprawdzie wyszukiwanie katalogu X.500 jest szybkie i dostępne globalnie, lecz są z nim związane pewne wady, z których najważniejsze to:

- ♦ *Złożoność* — ponieważ X.500 jest złożonym standardem, ani jeden z twórców aplikacji nie zaimplementował go w pełni.
- ♦ *Brak obsługi katalogu firm (yellow pages)* — przeszukiwanie katalogu firm nie jest obsługiwane w X.500, przez co użytkownicy nie mają dostępu do pełnego zbioru informacji.

- ◆ *Brak obslugi TCP/IP* — X.500 nie obsługuje TCP/IP, przez co nie jest możliwe równoczesne korzystanie z informacji w Internecie i w X.500.

Protokół LDAP (*Lightweight Directory Access Protocol*) został opracowany w roku 1995, aby zaradzić niedociagnięciom poprzednika — X.500. Pomyśl polegał na zmniejszeniu trudności z dostępu do X.500 i udostępnieniu usługi katalogowej dla różnorodnych aplikacji i komputerów. Podobnie jak X.500, LDAP umożliwia dostęp do katalogów informacji, lecz jest od X.500 znacznie prostszy. I co szczególnie ważne, obsługuje protokół TCP/IP, niezbędny do korzystania z Internetu.



Dodatkowe informacje o X.500 można znaleźć w FRC 2116, 1279 i 1292.

LDAP

Lightweight Directory Access Protocol został opracowany w University of Michigan na potrzeby dostępu online do usług katalogowych X.500 w Internecie oraz do odpytywania o informacje i manipulowania nimi. W istocie University of Michigan i wiele innych uniwersytetów nadal za pomocą LDAP kieruje poczta elektroniczna i udostępnia wyszukiwanie nazwisk.

Informacje składowane w katalogach opartych na X.500 są uporzadkowane w hierarchiczna strukturze, przypominającej drzewo. Bedąc protokołem klient-serwer, LDAP umożliwia przeglądanie, odczyt i wyszukiwanie informacji zapisanych w usługach katalogowych X.500 w całym Internecie oraz pozwala na wykonywanie prostych zadań związanych z zarządzaniem.

LDAP jest nie tylko protokołem dostępu, lecz także prosta, szybka i skalowalna usługa katalogowa. Prosta, gdyż LDAP jest uproszczona wersja X.500, przez co dostęp do katalogów X.500 za pomocą LDAP nie jest skomplikowanym zadaniem. Z tego powodu LDAP czasami nazywany jest *X.500 Lite* (dosłownie „lekki X.500”). Choć tylko podzbiór funkcji X.500 wykorzystano ponownie w LDAP, to protokół ten posiada pełną funkcjonalność protokołu dostępu do katalogu X.500, utrzymując równoczesnie pełną zgodność z X.500. Ponieważ LDAP pozwala na łatwy dostęp do usług katalogowych i zapewnia bezproblemową obsługę TCP/IP, jest lepszym wyborem dla środowiska internetowego. Dzięki temu LDAP został obecnie w pełni zaakceptowany w roli internetowego standardu usług katalogowych, działających przez TCP/IP.



Skalowalność oznacza zdolność systemu — sprzętu, oprogramowania lub sieci — do przystosowania się w miarę potrzeb do przyszłego rozwoju i zmian.

Sukces LDAP był tak duży, iż wielu producentów usług katalogowych (w tym Microsoft z Active Directory, Banyan ze Street Talk i Netscape Directory Server) korzysta z LDAP w roli centralnej technologii usług katalogowych. Wiodący dostawca sieciowych usług katalogowych, Novell, również oferuje moduły rozszerzające LDAP dla usługi NDS i zintegrował całkowicie NetWare 4.11 (IntraNetware) i nowsze produkty z usługą LDAP.



Obecnie dostępne są trzy wersje LDAP: LDAP, LDAPv2 i LDAPv3. Usluga LDAP powstała w roku 1993. Wersja LDAPv2 wyszła w roku 1996 i została zaadaptowana komercyjnie. LDAPv3 jest aktualna wersja, zatwierdzona w 1997 r. Active Directory Service i NDS obsługują LDAPv3.

LDAP jest standardem otwartym, wobec czego pozwala dowolnej aplikacji, działającej na dowolnej platformie, na dostęp do autonomicznych usług katalogowych LDAP lub usług katalogowych opartych na serwerze X.500 i pobieranie z nich informacji. Do tego celu służy prefiks `LDAP://` w adresie URL serwera LDAP (metoda bardzo podobna do protokołów FTP i HTTP). Do usług obsługiwanych przez LDAP zalicza się wyszukiwanie adresów e-mail przez klienta poczty elektronicznej za pomocą LDAP, tekst, animacje, dźwięk, adresy URL i autoryzowany dostęp do ważnych informacji (na przykład kluczy publicznych).

Ponieważ LDAP jest oparty na X.500, ma wiele podobnych właściwości, na przykład strukturę katalogu, model danych i przestrzeni nazw oraz sposób dostępu do danych.

- ♦ LDAP, podobnie jak X.500, używa globalnej struktury katalogowej. Na przykład, informacje są przechowywane w serwerze LDAP w postaci wpisów. Typ jest definiowany przez klasę obiektu, do której wpis może należeć. Kazdy wpis jest zbiorem atrybutów, zawierających faktyczne informacje.
- ♦ Model danych i przestrzeni nazw LDAP przypomina model używany przez X.500. Jeden lub wiele serwerów LDAP zawiera dane składające się na drzewo katalogów LDAP. Ponadto LDAP używa tej samej hierarchii wpisów, która funkcjonuje w X.500. Hierarchia zaczyna się od korzenia drzewa informacji katalogowych (DIT — *Directory Information Tree*) i rozchodzi aż do poziomu osób.
- ♦ Sposób, w jaki informacje są organizowane oraz sposób dostępu do nich w LDAP i X.500 są podobne. Klient LDAP może wysłać zadanie informacji lub zgłosić dane do aktualizacji. Po otrzymaniu zadania serwer LDAP sprawdza prawa klienta do zadanego informacji. Jeśli klient posiada odpowiednie prawa dostępu, serwer odpowiada na zadanie lub kieruje klienta do innego serwera LDAP, w którym informacje są dostępne.
- ♦ Można zastosować kontrole dostępu do uprawnien odczytu, zapisu, wyszukiwania lub porównywania informacji dostępnych w serwerach LDAP. Zasady te mogą być implementowane dla pojedynczego użytkownika lub dla grupy. Kontrole dostępu można ograniczyć do poziomu części drzewa, wpisu lub nawet atrybutu.
- ♦ Replikacja informacji jest możliwa również dla serwerów LDAP.

W prawdziwej LDAP i X.500 są bardzo do siebie podobne, lecz jest między nimi kilka różnic:

- ♦ Protokół LDAP został zaprojektowany do współpracy ze stosem protokołów TCP/IP, co pozwala na dostęp do informacji znajdujących się w Internecie. X.500 nie posiada tej możliwości, ponieważ nie obsługuje TCP/IP.
- ♦ LDAP jest o wiele prostszy pojęciowo i łatwiejszy do zaimplementowania. Dzięki temu właśnie szybko zyskał aprobaty producentów.



Trwają prace nad umożliwieniem obsługi katalogów branżowych (*yellow pages*) przez LDAP.

Dostepnych jest wiele klientów LDAP, pozwalajacych przegladac katalogi zgodne z tym standardem. Najbardziej znanymi klientami LDAP sa programy pocztowe typu Outlook Express lub Netscape Communicator. Po zintegrowaniu z LDAP, klienci te moga korzystac z obszernych informacji w postaci ksiazek adresowych. Co wiecej, kazdy klient z wystarczajacymi uprawnieniami dostepu moze korzystac z dowolnego serwera LDAP.



Dodatkowe informacje o LDAP zawiera RFC 1777.

Najpopularniejsze uslugi katalogowe opieraja sie na standardach X.500 i LDAP. Do tych uslug naleza NIS, NIS+, NDS, STDS i Active Directory (AD). Tabela 17.1 przedstawia pobiczny przeglad wymienionych uslug katalogowych.

Tabela 17.1. Popularne uslugi katalogowe

Wlasciwosc	NIS	NIS+	STDS	NDS	AD
Hierarchia informacji	Plaska przestrzen nazw	Hierarchiczna przestrzen nazw	Hierarchia maks. do trzech poziomów	Hierarchiczna organizacja danych	Hierarchiczna organizacja danych
Aktualizacje danych	Tak	Tak	Tak	Tak	Tak
Replikacja danych	Ograniczona	Tak	Tak	Tak	Tak
Partyjonowanie danych	Tak	Tak	Tak	Tak	Tak
Dostep globalny	Nie	Nie	Ograniczony	Tak	Tak
Wyszukiwanie hostow	Tak	Tak	Tak	Tak	Tak
Interfejs	Wiersz polecen	Wiersz polecen	Graficzny	Graficzny	Graficzny
Poziom bezpieczenstwa	Wyjatkowo niski	Podatnosc na ataki	Niski	Wysoki	Wysoki
Obciaszenie sieci	Wysokie	Wysokie	Wysokie	Niskie	Niskie
Wspoloperatywnosc	Wyjatkowo ograniczona	Ograniczona	Ograniczona	Ograniczona	Wysoka

Aby komunikowac sie ze soba, uzytkownicy, komputery i aplikacje wymagaja określonych informacji: adresów innych komputerów, ustawien zabezpieczeń, adresów e-mail uzytkowników, danych interfejsu sieciowego, uslug dostepnych w sieci, istniejacych grup uzytkowników i tak dalej. Wraz z postepami w technologii sieciowej rośnie również ta lista. Co wiecej, siec komputerowa jest srodowiskiem dynamicznym. Do sieci dodawane sa wciaz nowe komputery i uslugi, co prowadzi do zmian w informacjach. Zmiany musza byc aktualizowane w kazdym komputerze. Przy braku centralnej uslugi

zarządzającej każdy komputer musiałby utrzymywać własna kopię informacji. W małych sieciach to zadanie jest nudzace, lecz do opanowania; w średnich i dużych sieciach opartych na systemach Unix lub Linux zadanie to staje się czasochłonne i wymaga się spod kontroli. Skutecznym rozwiązaniem tego problemu może być usługa NIS (*Network Information Service*).

NIS

Usluga NIS (*Network Information Service* — sieciowe usługi informacyjne) zarządza informacjami potrzebnymi użytkownikom, komputerom i aplikacjom do komunikowania się ze sobą. Usługa ta może być rozproszona w sieci lokalnej (LAN). Informacje, którymi zarządza NIS, noszą nazwę *przestrzeń nazw NIS* (*NIS namespace*). Informacje są składowane w sieci w różnych serwerach NIS i udostępniane każdemu komputerowi, który ich zaząda. Komputery te są *klientami NIS*.



Historyczna nazwa NIS to *Yellow Pages* (YP).

Istnieją dwa typy serwerów NIS: *główne (master)* i *podporządkowane (slave)*:

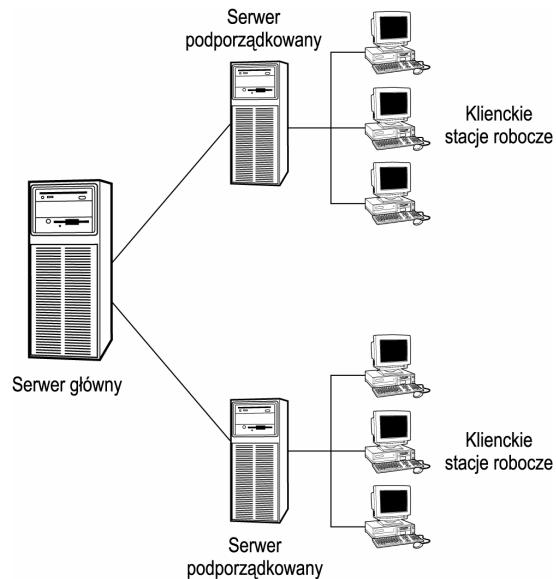
- ◆ *Serwery główne* — utrzymują główne bazy danych dla danej domeny, zawsze zawierające aktualizowane informacje. Serwery główne są zasadniczo dostępne tylko dla administratorów sieci.
- ◆ *Serwery podporządkowane (inaczej serwery-kopie)* — w dużej sieci kilka komputerów (hostów) zostaje wyznaczonych do roli serwerów podporządkowanych NIS, aby zapobiec przeciążeniu serwera głównego. Do serwerów podporządkowanych można uzyskać dostęp ze stacji roboczych, wobec czego muszą one utrzymywać dokładną kopię bazy danych, utrzymywanej przez serwer główny. Wszelkie zmiany w konfiguracji sieci są najpierw aktualizowane w serwerze głównym, a następnie propagowane do serwerów podporządkowanych, które dodatkowo pełnią funkcje kopii zapasowej serwera głównego.



Liczba serwerów podporządkowanych w domenie musi być tak dobrana, aby dostępność była wysoka, a czas odpowiedzi krótki, bez tworzenia kosztów niepotrzebnej replikacji danych do zbyt wielu hostów.

Klienci i serwery są zgrupowane w *domenie NIS* — zestawy serwerów i klientów, związane razem z powodu jakiegoś wspólnej cechy, na przykład położenia geograficznego. Każda domena ma skojarzony ze sobą zbiór właściwości. Właściwości te są przechowywane w bazach danych zwanych *mapami*, razem z innymi danymi systemowymi, na przykład nazwami użytkowników, hasłami i nazwami hostów. Informacje zapisane w mapach są uporządkowane w dwie kolumny: jedna przechowuje klucz, druga informacje o kluczu. Na przykład, nazwy stacji roboczych są zapisywane w mapie *hostsbyname*, a ich adresy w mapie *hosts.byaddress*. Informacje zadane przez klienta są znajdują się w tych kluczach. Domena NIS z serwerem głównym, dwoma serwerami podporządkowanymi i kilkoma klientami jest przedstawiona na rysunku 17.4.

Rysunek 17.4.
Domena NIS



Korzystanie z NIS przynosi wiele korzyści, miedzy innymi:

- ◆ *Bezpieczenstwo* — zapewnia wysoki poziom bezpieczeństwa, poniewaz kazda operacja moze byc uwierzytelniana.
- ◆ *Interfejs uzytkownika* — zapewnia uzytkownikom dostep z prawami odczytu i zapisu.
- ◆ *Przyrostowe aktualizacje danych* — jedynie czesc danych jest zmieniana.
- ◆ *Wyszukiwanie hostów* — pozwala w danej sieci wyszukiwac hosty na podstawie adresów IP, podobnie do DNS-u. DNS jest usluga pozwalajaca znajdowac hosty (lub urzadzenia sieciowe) w sieciach opartych na TCP/IP.
- ◆ *Obsluga danych binarnych i ASCII* — DNS obsluguje jedynie dane ASCII z ograniczeniami rozmiaru pakietu.



Dodatkowe informacje o usludze DNS zawiera rozdzial 10.

Usluga NIS ma również kilka wad:

- ◆ *Plaska przestrzen nazw* — poniewaz NIS nie obsługuje hierarchicznej przestrzeni nazw, w praktyce jest przydatna jedynie w sieciach lokalnych.
- ◆ *Scentralizowane bazy danych* — obsługiwane bazy danych sa scentralizowane i niehierarchiczne. Na skutek tego NIS nie obsługuje partycjonowania baz danych.
- ◆ *Ograniczone zdolnosci do replikacji* — jedna replika moze obsługiwać tylko jedna podsieć, przez co klienci z jednej domeny nie mają dostępu do serwerów w innych domenach.

- ♦ *Zależność od administratora* — dostęp do bazy danych mają tylko administratorzy, wobec tego tylko oni lub użytkownicy uprzywilejowani (root) mogą aktualizować i propagować informacje.



Dodatkowe informacje o NIS można znaleźć w RFC 2307.

Z powodu wad NIS, przez które było trudno było zarządzać systemem (zwłaszcza w zbyt duzych sieciach), opracowana została usługa *Network Information Service +* (NIS+).

NIS+

NIS+ jest rozszerzeniem usługi NIS, obsługującym przestrzeń nazw NIS+. W przeciwieństwie do płaskiej przestrzeni nazw oferowanej przez NIS, przestrzeń nazw NIS+ jest hierarchiczna. Pod tym względem przestrzeń nazw NIS+ jest bardzo podobna do struktury katalogów systemu Unix. Dzięki hierarchiczności przestrzeń ta można podzielić na wiele domen. Każda domena może mieć własnego administratora i nie musi polegać na zarządzaniu centralizowanym. Użytkownicy o odpowiednich uprawnieniach i prawach mogą uzyskać dostęp do serwerów położonych w innych domenach.



Sposób, w jaki ulozone są informacje w przestrzeni nazw, nie musi odpowiadać fizycznemu rozkładowi zasobów sieciowych.

Architektura NIS+ bardzo przypomina architekturę NIS: opiera się na modelu klient-serwer i podziale na domeny. Każda domena zawiera zbiór serwerów i klientów. W domenie jeden serwer główny kontroluje pozostałe serwery podporządkowane (repliki), oraz utrzymuje baze danych NIS+, składającą się z wielu tablic NIS+. Tablice te, w sposób ciągły aktualizowane przez serwer główny, są kopiowane do serwerów-replik. NIS+ stosuje złożone mechanizmy uwierzytelniania i autoryzacji, aby weryfikować, czy klient ma odpowiednie prawa i uprawnienia pozwalające na dostęp do bazy danych NIS+.

Zalety NIS+ w porównaniu z NIS są następujące:

- ♦ *Obsługa hierarchicznej przestrzeni nazw* — dzięki temu NIS+ może dokładniej odzworować hierarchię organizacyjną.
- ♦ *Obsługa partycjonowania NIS+* — baza danych NIS+ może zostać podzielona na katalogi, które z powodzeniem zapewniają obsługę autonomicznych domen.
- ♦ *Replikacja* — każda replika (serwer zapasowy) może obsługiwać wiele podsieci.
- ♦ *Aktualizacje danych* — każdy klient posiadający odpowiednie prawa, nawet zdalny, może w razie potrzeby aktualizować bazę danych NIS+. Aktualizacje mogą być zarówno automatyczne, jak i inicjowane przez serwer główny.

Do wad NIS+ zaliczają się:

- ♦ *Zwiększyły ruch sieciowy* — obsługiwane przez NIS+ okresowe automatyczne aktualizacje generują większy ruch w sieci.

- ◆ *Interfejs wiersza polecen* — NIS+ obsluguje tylko wiersz polecen, który dla części użytkowników może być niewygodny.
- ◆ *Bezpieczeństwo* — mimo że NIS+ stosuje złożony system zabezpieczeń do ochrony struktury przestrzeni nazw i zawartych w niej informacji, dostęp do danych jest łatwiejszy niż w NIS, co powoduje podatność na ataki.



Dodatkowe informacje o NIS+ zawierają RFC 2196 i 2065.

Jedna z najstarszych i najdłużej stosowanych sieciowych usług katalogowych jest *StreetTalk Directory Service* (STDS) firmy Banyan Systems, Inc. Była ona używana w sieciowym systemie operacyjnym Banyan Vines na dłużej przed tym, jak inni producenci zaczęli zajmować się ideą katalogów sieciowych. W istocie, usługa STDS jest tak stara, że czasami nazywana jest „prababcią” sieciowych usług katalogowych.

STDS

StreetTalk Directory Service udostępnia rozproszona bazę danych nazw i adresów, która kojarzy nazwy z użytkownikami i zasobami sieciowymi niezależnie od położenia zasobów w intranecie. Położenie to jest dla użytkownika całkowicie niewidoczne. StreetTalk najlepiej nadaje się dla przedsiębiorstw, które posiadają centra zarządzania rozrzucone po całym świecie. Podobnie jak NIS+, STDS rozprasza usługi katalogowe w sieci firmy na wiele serwerów, rozmieszczonych w wielu domenach. Każdy serwer w sieci obsługuje partie usługi katalogowej, które są automatycznie synchronizowane. Aby umożliwić użytkownikom dostęp do serwerów z dowolnego miejsca sieci, informacje o obiekcie są przechowywane w serwerze, który zawiera ten określony obiekt. StreetTalk poza istniejącymi atrybutami klasy obiektu pozwala dodawać nowe atrybuty, zgodnie z potrzebami przedsiębiorstwa.



Serwery StreetTalk potrafią automatycznie dodawać same siebie do katalogu. Część klientów również może być tak skonfigurowana, aby rejestrować się automatycznie w katalogu, przez co stają się widoczne dla administratorów.

W prawdziwej części zadań administracyjnych, na przykład konfiguracja sprzętu, musi odbywać się z konsoli serwera, lecz serwery StreetTalk udostępniają *Enterprise Network Services Management Tool* (narzędzie do zarządzania usługami sieciowymi przedsiębiorstwa), które jest narzędziem graficznym ułatwiającym pozostałe zadania administracyjne. Z uwagi na rozproszony charakter STDS, w klientach dostępne jest narzędzie do przeglądania katalogu, o nazwie *StreetTalk Directory Assistant* (STDA). STDA pomaga klientom szukać obiektów i innych zasobów sieciowych. STDA okresowo przegląda wszystkie serwery w sieci, gromadząc informacje o użytkownikach, urządzeniach periferyjnych i woluminach.

Drzewo StreetTalk może mieć najwyżej trzy poziomy, przez co usługa katalogowa StreetTalk obsługuje jedynie jednostki organizacyjne pierwszego poziomu, bez możliwości zagnieżdżania. Użytkownicy logują się do StreetTalk raz, po czym mają dostęp do wszystkich serwerów, do których dali im uprawnienia administratorzy.

Usluga STDS ma nastepujace zalety:

- ♦ Jest latwa do skonfigurowania i utrzymania.
- ♦ Udostepnia pojedynczy punkt zarzadzania, ułatwiajac wykonywanie zadan administracyjnych. Administratorzy sieciowi nie musza poswiecic zbyt duzo czasu na zarzadzanie uslugami katalogowymi, co pozwala im przeniesc uwage na zarzadzanie uzytkownikami.
- ♦ Tworzy partycje uslugi katalogowej w kazdym serwerze StreetTalk, rozkladajac w ten sposob obciaszenie na wszystkie serwery. Dzieki temu system jest bardziej odporny na bledy, gdyz nie zawiera pojedynczego punktu awarii.
- ♦ Zarzadza aktualizacjami i synchronizuje rozproszone partycje katalogu automatycznie, bez potrzeby interwencji administratora.
- ♦ Moze automatycznie laczyc dwa drzewa.

Korzystanie z STDS ma również kilka wad:

- ♦ Nie umozliwia niskopoziomowej kontroli nad katalogiem.
- ♦ Wsparcie innych producentów jest bardzo ograniczone.
- ♦ Obsluguje jedynie trzy poziomy drzewa katalogów. Z tego powodu mozna w niej zaimplementowac jedynie pierwszy poziom jednostek organizacyjnych, a duze organizacje moga miec klopoty z nazewnictwem. Zagniezdzanie również nie jest obsługiwane.
- ♦ Prawa dostepu opieraja sie na nazwach obiektów, co moze doprowadzic do problemów z bezpieczenstwem, związanych z dostepem.



Usluga StreetTalk jest obecnie dostepna również dla innych sieciowych systemów operacyjnych poza Banyan Vines — dla różnych wersji Uniksa, NetWare i Windows NT Microsoftu. Jednakze wypuszczenie na rynek wersji dla Windows NT zakonczylo sie porazka z powodu zbyt wielu problemów.

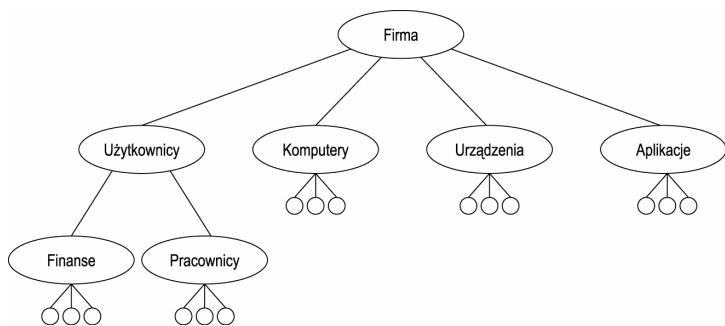
Wprawdzie firma Banyan Systems, Inc. usilowala osiagnac zgodnosc STDS z innymi waznymi sieciowymi systemami operacyjnymi, lecz spotkala sie z pewnymi niepowodzeniami i ostra rywalizacja na rynku uslug katalogowych. Jednym z najbardziej popularnych dostawcow uslug katalogowych jest Novell Corporation. Usluga Network Directory Service (NDS) Novella jest na rynku od dawna i udowodnila, ze jest stabilnym produktem, który wytrzymal nacisk konkurencji i wylonil sie jako lider rynku uslug katalogowych. NDS wprowadzono po raz pierwszy w NetWare 4.0 i usługa ta jest dostepna we wszystkich wersjach do 5.0. Oferowana jest również w najnowszej wersji NetWare 6.0 i nosi nazwe *eDirectory* oraz dla innych popularnych sieciowych systemów operacyjnych, na przyklad Windows NT i Solaris.

Network Directory Service Novella

NDS jest rozprosiona usługa katalogowa, która magazynuje informacje związane z Internetem, intranetem i innymi zasobami sieciowymi. Zapewnia również bezpieczny dostęp do tych zasobów poprzez zlozone usługi uwierzytelniania i kontroli dostępu. Usługa NDS opiera się na X.500 i jest zgodna z LDAP.

Wszystkie zasoby sieciowe, takie jak użytkownicy i różnorodne aplikacje, oraz wolimy w sieciowym systemie plików są reprezentowane w NDS jako obiekty. Urządzenia fizyczne (np. drukarki, faksy lub skanery) są również uznawane za obiekty. Na przykład, użytkownik jest reprezentowany przez *obiekt użytkownika* (*user object*). Działły, departamenty i grupy robocze są reprezentowane w organizacji jako *organizational-unit object* (obiekt jednostki organizacyjnej), inaczej *branch object* (obiekt gałęzi). Z każdym obiektem skojarzone są atrybuty, nazywane *właszczyściami* lub *polami*. Na przykład, z obiektem użytkownika mogą być skojarzone następujące pola: adres wezła, ID logowania (nazwa użytkownika), hasło, imię i nazwisko, adres, e-mail i numer telefonu. Drzewo katalogów NDS zostało przedstawione na rysunku 17.5.

Rysunek 17.5.
Struktura drzewa katalogów NDS



Novell sprzedaje NDS z przykładowymi drzewami katalogów dla dużych i małych przedsiębiorstw. Pomaga to administratorom sieciowym implementować NDS bez zbytnich problemów.

Dostęp do zasobów za pośrednictwem interfejsu NDS przypomina dostęp do plików i katalogów w Eksploratorze systemu Windows Microsoftu. Domyslnym widokiem NDS jest węzel główny (korzeń) struktury katalogów. Administrator sieci może budować drzewo katalogów reprezentujące strukturę organizacji. Każda gałąź drzewa odpowiada wtedy logicznej grupie użytkowników. Podział użytkowników na grupy może odbywać się według departamentów, działów lub lokalizacji geograficznej.

NDS jest najsielszta i najbardziej stabilna z hierarchicznych sieciowych usług katalogowych dostępnych dzisiaj na rynku. Niektóre z zalet usługi NDS to:

- ♦ *Scentralizowana kontrola* — niezależnie od stopnia rozproszenia sieci przedsiębiorstwa, administrator może kontrolować całą sieć z jednego miejsca. NDS udostępnia również graficzne narzędzia administracyjne, ułatwiające zarządzanie.
- ♦ *Globalny dostęp do zasobów sieciowych* — sieć przedsiębiorstwa może zawierać wiele serwerów, lecz fakt ten jest niewidoczny dla użytkowników, ponieważ NDS udostępnia prosty widok zlozonej infrastruktury. Gdy użytkownicy logują się do systemu, zamiast zbioru osobnych serwerów, widzą pojedynczy system informacyjny. Mogą korzystać z dowolnych zasobów, do których mają prawo dostępu. Ten system ułatwia dodatkowo administrację i redukuje koszty zarządzania.
- ♦ *Replikacja* — pozwala administratorom zapisywać całe kopie lub części bazy danych NDS w wielu serwerach, które mogą mieścić się w różnych domenach. Taka struktura udogodnia system na awarie i redukuje ruch sieciowy po laczach WAN.

- ♦ *Hierarchiczne drzewo katalogów* — ułatwia dostęp do zasobów i pozwala na administrację oparta na regulach. Różne widoki hierarchiczne w usłudze NDS pozwalają również administratorom zobaczyć logiczne i fizyczne rozmieszczenie obiektów katalogowych.



Administrowanie oparte na regulach pozwala administratorom przyznawać uprawnienia całej gałęzi użytkowników za jednym zamachem, przez co nadawanie praw dostępu wszystkim użytkownikom w firmie jest łatwe i szybkie. Potrzeba zarządzania poszczególnymi użytkownikami i zasobami jest dzięki temu również zminimalizowana.

- ♦ *Rozszerzalność* — niezależni producenci oprogramowania mogą integrować nowe usługi i dodawać nowe obiekty do sieci przez rozszerzenie schematu NDS.



Schemat NDS to zbiór reguł kontrolujących struktury drzewa katalogów. Schemat ustala definicje, atrybuty, właściwości i położenie wszystkich obiektów w drzewie.

- ♦ *Skalowalność* — NDS można dostosować do pracy w sieci o dowolnych rozmiarach. Usługa NDS jest również zdolna do obsługi fuzji przedsiębiorstw, które mogą prowadzić do fuzji dwóch sieci komputerowych.
- ♦ *Elastyczność* — projekt NDS jest bardzo elastyczny. Drzewo katalogów może być modyfikowane tak, by objąć wszelkie zmiany w strategii firmy lub w sieci. Obiekty, grupy, a nawet całe gałęzie drzewa mogą być przemieszczane prostą metodą „przeciagnij i upuść”.
- ♦ *Bezpieczeństwo* — NDS zapewnia maksymalne zabezpieczenie przed intruzami i nieupoważnionym dostępem. Gdy użytkownicy korzystają z zasobów lub usługi sieciowej, dostęp jest dozwolony po uwierzytelnieniu tylko wtedy, gdy posiadają wystarczające prawa do obiektu.



Caly proces uwierzytelniania jest dla użytkowników niewidoczny. Uwierzytelnianie opiera się na sesjach, zas podpis użytkownika jest ważny tylko dla danej sesji.

NDS i wszelkie inne usługi katalogowe (STDS, NIS+, NIS, LDAP i X.500) mają jedną wadę — stosują własny interfejs, który może, lecz nie musi być zgodny z innymi interfejsami. Ponadto, z uwagi na ogólną popularność środowiska Microsoft Windows, większość użytkowników końcowych preferuje pracę z interfejsem graficznym Windows. Co więcej, zmuszanie różnych aplikacji do pracy z tymi usługami katalogowymi jest zajeciem nudącym. Trzeba zaimplementować szereg idei i interfejsów na poziomie programistycznym, a każda usługa katalogowa musi zostać obsługiwana osobno.

Usługa Active Directory Microsoftu została zaprojektowana tak, aby umożliwić standaryzowany dostęp do powszechnie używanych różnorodnych usług katalogowych. Dostępne w niej narzędzie Active Directory Services Interface udostępnia wysoki poziom współpracy z innymi usługami katalogowymi, dzięki czemu użytkownicy Active Directory mogą uzyskać dostęp do informacji z innych usług katalogowych.

Active Directory

Active Directory oferuje standardowy interfejs dla wszystkich istniejących usług katalogowych za pomocą narzędzia ADSI (*Active Directory Services Interface — Interfejs usług Active Directory*), które implementuje faktyczny dostęp do danych poprzez interfejs (tzw. *dostawce — provider*), pozwalający skontaktować się z wybranym katalogiem. Dla każdej usługi katalogowej obsługiwanej przez Active Directory istnieje osobny dostawca. Gdy klient Active Directory wysyła zadanie informacji o użytkowniku, komputerze, aplikacji lub zasobach w innej usłudze katalogowej, do odpytania tej usługi stosowany jest ADSI. Następnie wywołuje on określonego dostawcę, który szuka zadań informacji. Po przetworzeniu zapytania użytkownik otrzymuje informacje albo komunikat, iż szukany zasób nie został znaleziony. Cały proces jest niewidoczny dla klienta, który nie musi znać szczegółów implementacji i położenia usługi katalogowej.

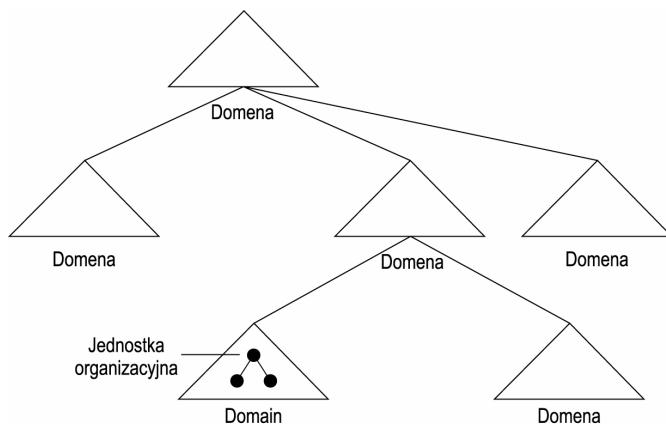


Active Directory również opiera się na standardzie LDAP.

Ponieważ Active Directory opiera się na LDAP, model danych używany przez tę usługę jest bardzo podobny do modelu LDAP. Wpis definiowany jest przez zestaw atrybutów zwanych właściwościami. W przeciwieństwie do płaskiej struktury domen w NTDS (*NT Directory Service*), Active Directory tworzy hierarchiczne drzewo domen. Wszystkie obiekty (użytkownicy, komputery, urządzenia periferyjne i aplikacje) w domenie są traktowane jak wpisy i mogą być zorganizowane w strukturę jednostek organizacyjnych. W związku z tym domena również posiada strukturę drzewa, co zwiększa skalowalność przestrzeni nazw. Każda domena zawiera serwer katalogowy zwany *kontrolerem domeny*, który umożliwia dostęp do wszystkich obiektów w domenie. Rysunek 17.6 przedstawia strukturę domen Active Directory.

Rysunek 17.6.

Drzewo domen
Active Directory



Cała komunikacja wewnętrzna i na zewnątrz domeny odbywa się poprzez LDAP. Różne typy klientów „widzą” drzewo domen w różny sposób. Na przykład, klienci Windows NT „zobaczą” strukturę domen NT, natomiast inne klienci korzystające z LDAP — interfejs katalogu LDAP, który może być przeglądany w taki sam sposób, jak DIT X.500.

Do szybkiego i wydajnego wyszukiwania zdalnych zasobów służy wykaz globalny (GC — *Global Catalog*). Kazdy obiekt w drzewie domen musi być zarejestrowany w GC, który jest zoptymalizowany pod względem szybkości. Gdy użytkownik lub aplikacja szuka obiektu w innej domenie drzewa, odpisywany jest GC zamiast zdalnego katalogu. Pozwala to użytkownikom łatwo i szybko znajdować zasoby.

Oprócz wyszukiwania danych oraz zarządzania użytkownikami i zasobami, Active Directory upraszcza zarządzanie siecią. Do zalet AD należą:

- ◆ *Scentralizowane zarządzanie* — możliwe jest kontrolowanie całej sieci z jednego miejsca.
- ◆ *Wspólny interfejs* — dostępny jest jeden typ interfejsu, za pośrednictwem którego można zarządzać różnymi usługami katalogowymi, na przykład książka adresowa lub poczta elektroniczna. Co więcej, wspólny interfejs pozwala na dodatkową współpracę operatywną pomiędzy innymi usługami katalogowymi.
- ◆ *Zintegrowane zabezpieczenia* — gdy użytkownik loguje się, jego dane zostają wprowadzone do pojedynczego systemu uwierzytelnienia. Pomaga to administratorom ustawiając parametry jednokrotnie dla każdego użytkownika lub obiektu. Ponadto uprawnienia dostępu można definiować dla całych obiektów, jak również dla ich właściwości. Zapewnia to bezpieczeństwo ważnych informacji.
- ◆ *Rozszerzalność* — administratorzy mogą dodawać nie tylko nowe obiekty i atrybuty do drzewa domen, lecz również nowe typy obiektów i atrybutów do istniejącego schematu Active Directory. W ten sposób można dostosować drzewo domen do wymagań przedsiębiorstwa.
- ◆ *Replikacja danych* — w drzewie domen możliwa jest replikacja multi-master. Obecność kilku serwerów zwiększa dostępność sieci i udogodnia cały system na bieżąco przez wyeliminowanie pojedynczego punktu awarii.
- ◆ *Elastyczne zapytania* — użytkownicy mogą wyszukiwać informacje na podstawie słowa, nazwy lub adresu. Zastosowanie wykazu globalnego skraca czas reakcji i zwiększa wydajność sieci.

Podstawowa wada Active Directory jest fakt, że funkcjonuje jedynie na platformie Windows i nie jest zgodna z innymi platformami. Aby więc korzystać z zalet Active Directory, użytkownik musi zaimplementować usługę tylko na platformie Windows.

Czesc IV

Tworzenie i utrzymanie sieci TCP/IP

W tej czesci:

- ◆ Rozdzial 18. Wybór schematu adresowania
- ◆ Rozdzial 19. Projektowanie trasowania dla sieci
- ◆ Rozdzial 20. Planowanie rozmieszczenia serwerów
- ◆ Rozdzial 21. Wprowadzenie do lacznosci
- ◆ Rozdzial 22. Planowanie bezpieczenstwa sieci
- ◆ Rozdzial 23. Rozwiazywanie problemów z siecia i lacznoscia
- ◆ Rozdzial 24. Monitorowanie sieci TCP/IP
- ◆ Rozdzial 25. Planowanie na przyszlosc

Ta czesc poswiecona jest budowaniu sieci TCP/IP. Po lekturze dotychczasowych rozdziałów Czytelnik powinien wiedziec, jak zainstalowac i skonfigurowac TCP/IP, oraz miec dobre pojecie o dostepnych narzedziach. Teraz pora polaczyc wszystkie skladniki w sieci, ktora bedzie funkcjonalna, bezpieczna i niezawodna.

Rozdzial 18. pozwoli okreslic potrzeby zwiazane z adresowaniem i zdecydowac, jak otrzymac potrzebne adresy. Po ustaleniu tego bedziemy mogli zdecydowac, jak podzialic posiadana przestrzen adresowa na potrzeby topologii.

Nastepnie nalezy zidentyfikowac strukture trasowania, jakiej potrzebuje nasza siec — ten temat zostal omówiony w rozdziale 19. Dalej musimy okreslic, ile ruterow bedzie potrzebnych, i jak beda aktualizowac informacje o trasach. Po skonfigurowaniu rutingu mozemy ustalic, gdzie umiescic serwery, aby najlepiej spelniali potrzeby klientow w sieci, czym zajmie sie rozdzial 20.

Gdy siec fizyczna bedzie juz zaprojektowana, mozemy zaczac przygladac sie potrzebom uzytkownikow laptopow i uzytkownikow pracujacych z domu w zakresie dostepu zdalnego do zasobow. Rozdzial 21. zajmie sie dostepem zdalnym i lacznoscia. Tematem zwiazanym z powyzszymi jest bezpieczenstwo sieci, a zwlaszcza uwierzytelnianie i szyfrowanie — zagadnienia omowane w rozdziale 22.

Teraz, majac w reku plan sieci, mozemy przejsc do jego implementacji; w rozdziale 23. zajmiemy sie strategiami rozwiazywania problemow z lacznoscia i innych, wystepujacych w sieci. Monitorowanie sieci zostało omówione w rozdziale 24.

Rozdział 18.

Wybór schematu adresowania

W tym rozdziale:

- ◆ Szacowanie potrzeb dotyczących adresów
- ◆ Używanie adresów sieci prywatnych

Teraz, gdy Czytelnik poznał już funkcjonowanie TCP/IP, pora zacząć składając wszystkie elementy razem, aby zbudować sieć. W tym celu musimy przyjrzeć się wymogom adresowania w sieci, aby ustalić najlepszy schemat adresowania.

Sieci zazwyczaj nie są budowane od zera; najprawdopodobniej zadaniem Czytelnika będzie przekonstruowanie istniejącej sieci. Pierwszym krokiem w obu przypadkach jest określenie schematu adresowania, który będzie zastosowany.

Szacowanie potrzeb dotyczących adresów

Zanim wybierzemy schemat adresowania, musimy oszacować szereg czynników, miedzy innymi:

- ◆ fizyczna konfiguracja sieci,
- ◆ lokalizacje, które sieć musi obsłużyć,
- ◆ wymogi wydajności.

W następnych kilku punktach przyjrzymy się tym czynnikom dokładniej.

Fizyczna konfiguracja sieci

Dostępnych jest kilka typów sieci; każdy z nich przynosi określone korzyści. Na przykład, Token Ring pozwoli umieścić w jednym segmencie znacznie więcej systemów, niż jest to możliwe w sieci Ethernet, jednak szybkość transmisji jest w nim niższa. Wielkość topologii można rozbudować za pomocą mostów lub innych urządzeń sieciowych, które pozwala zwiększyć liczbę systemów w pojedynczym segmencie.

Aby skutecznie zaplanowac liczbę stacji, która pomiesci pojedynczy segment, musimy ustalić, jak duży ruch sieciowy każdy system będzie generować. W sieci może wystąpić kilka różnych typów ruchu sieciowego, w tym:

- ◆ transfery plików,
- ◆ odwiedzanie stron WWW,
- ◆ sprawdzanie poczty elektronicznej,
- ◆ aplikacje sieciowe typu SQL Server,
- ◆ adresowanie grupowe,
- ◆ aplikacje biurowe uruchamiane z serwera.

Każdy z tych typów przesyłu danych musi zostać wliczony w całkowity ruch sieciowy, generowany przez stacje robocze w sieci. Firma może posiadać działy, które będą generować więcej ruchu określonego typu, niż pozostałe. Na przykład, użytkownicy w dziale finansowym mogą przede wszystkim korzystać z aplikacji sieciowej, zaś dział graficzny może przesyłać więcej plików. W takim razie trzeba będzie przeprowadzić analizę ruchu sieciowego według typów stacji roboczych i ustalić, w którym segmencie te stacje będą się znajdować.

Dodatkowo możemy liczyć na utratę 25 – 50% przepustowości sieci na ruch tła, czyli na przykład usługi DNS, DHCP, WINS i (lub) replikacje katalogów, w zależności od typów serwerów w sieci. Ustalenie tych wartości pozwoli określić liczbę hostów, jaka będzie możliwa przyłączenie do segmentów, a co za tym idzie — ile segmentów będzie potrzebnych.

Planując podział sieci na segmenty (które będą dalej nazywać podsiećmi), musimy dodatkowo przemyśleć rozmieszczenie serwerów i ruch sieciowy pomiędzy różnymi użytkownikami. Znając ruch generowany przez użytkowników, możemy właściwie zaplanować rozmieszczenie serwerów, przez co uzyskamy kontrolę nad obciążeniem sieci. Kontrola ruchu sieciowego daje użytkownikom najszybsze możliwe czasy reakcji w danej topologii. W rozdziale 20. przyjrzymy się dokładniej planowaniu serwerów.

Oprócz znajomości ruchu sieciowego, który będzie obecny w segmencie, czytelnik musi również wiedzieć o ruchu pomiędzy segmentami. Nie stanowi to większego problemu, gdy segmenty podłączone są bezpośrednio do sieci szkieletowej; jeśli jednak segmenty rozrzucone są po budynku lub osrodku, a nawet po całym świecie, trzeba będzie rozważyć odpowiednie rozmieszczenie tych segmentów.

Lokalizacje obsługiwane przez sieć

W małych firmach zazwyczaj mamy do czynienia z niewielką liczbą komputerów. Zazwyczaj wszystkie systemy są umieszczone w jednej lokalizacji i możemy połączyć wszystkie sieci razem przez proste podłączenie wszystkich segmentów do sieci szkieletowej. W tym przypadku nie musimy przejmować się różnymi lokalizacjami, które trzeba obsłużyć.

Jednakże wraz ze wzrostem skali organizacji wzrasta również prawdopodobienstwo pojawienia się dużych odległości pomiędzy fragmentami sieci. W takim przypadku trzeba będzie zbudować sieć większą od prostej LAN. Sieci większe od LAN mają różne nazwy:

- ♦ *Campus Area Network (CAN)* — ten typ sieci łączy dwa lub więcej budynków stojących blisko siebie. Zazwyczaj użytkownik takiej sieci sam prowadzi kable pomiędzy budynkami i ma pełną kontrolę nad siecią.
- ♦ *Metropolitan Area Network (MAN)* — w tym typie sieci (miejscowym) budynki nie są blisko siebie, lecz nadal znajdują się w obrębie jednej metropolii. W tym przypadku trzeba uzyskać połączenie od lokalnego dostawcy usług internetowych lub operatora telefonii. Oznacza to, że część okablowania będzie poza kontrolą administratora.
- ♦ *Wide Area Network (WAN)* — sieci tego typu (rozległe) łączą systemy nie znajdujące się w tym samym mieście. W tym przypadku sieć może objąć zasięgiem stan (województwo), kraj lub nawet cały świat. Sieci WAN mogą korzystać z usług większego dostawcy, który będzie mógł udostępnić linie dzierżawione; możemy również w roli sieci szkieletowej wykorzystać Internet. W obu przypadkach dane są bardziej wystawione na widok publiczny i mamy mniejszą kontrolę nad warunkami na łączu. W gospodarce także większe koszty, gdy będziemy chcieli, na przykład, połączyć biura na dwóch kontynentach.

Dla większych sieci, typu CAN, MAN lub WAN, trzeba użyć schematu adresowania, który pozwoli ograniczyć ruch do podsieci w każdej lokalizacji, lecz zapewni równoczesnie łatwe połączenie pomiędzy różnymi oddziałami. Drugim problemem, który musimy wziąć pod uwagę, jest skuteczne przesyłanie pakietów nawet w przypadku awarii łączu pomiędzy biurami.

Wymogi wydajności

Wprawdzie miło byłoby połączyć wszystkich użytkowników w całej firmie gigabitowa sieć Ethernet, lecz jest to niepraktyczne z uwagi na koszty. Jednym z zagadnień, które obejmuje planowanie sieci, jest ustalenie realistycznego poziomu wydajności. Z punktu widzenia schematu adresowania, dwa czynniki wpływające na analizowaną przez użytkownika wydajność.

Pierwszy czynnik jest oczywisty — jeśli umieścimy zbyt wiele hostów w jednym segmencie sieci, wydajność na tym ucierpi. Dotyczy to szczególnie sieci z rywalizacją o dostęp do nosnika (np. CSMA/CD). W takich typach sieci wszystkie hosty „nasłuchują” w sieci, czekając na chwilę ciszy (gdy żaden inny system nie nadaje), a następnie korzystają z okazji do nadawania. Wraz ze wzrostem liczby systemów w sieci jest coraz więcej „szumu”, mniej okazji do nadawania i więcej kolizji. Z drugiej strony, im więcej segmentów sieć posiada, tym więcej routów trzeba do przesyłania ruchu.

Drugim czynnikiem, na który musimy zwrócić uwagę, jest typ ruchu generowanego przez hosty i faktyczny czas, jaki poswiecają na komunikacje. Jeśli użytkownicy przez większość czasu pracują lokalnie i tylko okazjonalnie korzystają z sieci, wówczas segment może zawierać więcej hostów. Natomiast jeśli użytkownicy używają pakietu biurowego z serwera, to klienci komunikują się z serwerem cały czas i trzeba będzie ograniczyć liczbę hostów w podsieci.

Istnieja dwa sposoby na zwiększenie liczby systemów w sieci bez stosowania kosztowych technologii. Do zmniejszenia ruchu w sieci moga posluzyc mosty lub przelaczniki. Prosze pamietac, ze zadna z tych metod nie jest doskonala, zas dopuszczalna liczba systemów w podsieci bedzie nadal ograniczona.

Stosowanie mostów

Most (*bridge*) moze polaczyc dwa segmenty w warstwie fizycznej, w której kazdy pakiet wysylany do sieci Ethernet lub Token Ring jest typu rozgloszeniowego — to znaczy, kazda stacja w magistrali lub pierscieniu „widzi” ten pakiet i ustala adres docelowy. Most również „widzi” ten adres. Jesli wiadomo, ze adres znajduje sie na innym porcie, most przesyła pakiet na ten port. Jesli adres docelowy jest na tym samym porcie, co adres nadawcy, pakiety sa ignorowane przez urzadzenie. Dzieki temu jedynie ruch skierowany na inny port jest przepuszczany przez most.

Korzystna z zastosowania mostu jest mozliwosc podlaczenia np. 100 systemów po kazdej stronie mostu i przesywanie z jednej strony na druga tylko ruchu, który tego wymaga. Oznacza to mozliwosc podlaczenia 200 hostów do jednej podsieci. Oczywiscie most moze posiadac wiecej niz dwa porty — na przyklad, mozemy polaczyc mostem piec segmentow po 50 systemów i zwiększyć liczbe komputerów w podsieci do 250.

Ujemna strona tej konfiguracji jest zaleznosc systemów od funkcjonowania koncentratora — inaczej mówiąc, gdy koncentrator przestanie dzialac, wiele systemów nie bedzie zdolnych do komunikacji. Ponadto most musi poznac adresy fizyczne (MAC) nalezace do kazdego portu, co moze troche potrwac. Jesli segmenty nie beda dobrze zaplanowane, most moze zostac przeciazony.

Stosowanie przelaczników

Przelaczniki, podobnie jak mosty, funkcjonuja w warstwie fizycznej. Jednakze w przeciwienstwie do mostów, które lacza segmenty, przelaczniki zwykle lacza poszczególne systemy. Gdy system wysyla dane, dochodza one do portu w przelaczniku. Przelacznik sprawdza docelowy adres MAC i przekazuje dane do portu, do którego przylaczony jest adresat, otwierajac wirtualny obwód pomiedzy dwoma urzadzeniami. Pozwala to urzadzeniom wysylac i odbierac równoczesnie. Inaczej mówiąc, przelacznik umozliwia lacznosc pełnodupleksowa, podwajajac ilosc danych, jaka mozemy przesylic.

Poniewaz kazdy port jest izolowany, stacje rozpoznaja siebie jako jedyne urzadzenie w sieci. Oznacza to, ze zawsze sa w stanie nadawac, poniewaz naleza do własnej domeny kolizji. Technologia przelaczania (komutacji) jest bardzo popularna, lecz i ona nakłada ograniczenia na liczbe stacji, które mozna polaczyc ze sobą. Ponadto, przelaczniki sa drozsze od zwyklych koncentratorów, które zastepuja w strukturze sieci fizycznej.

Reasumujac, wydajnosc sieci sprowadza sie do liczby stosowanych podsiec, miejsca przeznaczenia ruchu (lokalny lub nie), oraz liczby systemów w kazdej podsieci. Technologie przelaczania i mostów moga byc przydatne w zwiększeniu liczby systemów w jednym segmencie, lecz technologie te nadal maja ograniczenia. Ominiecie tych ograniczen wymaga trasowania, a co za tymidzie, wlasciwego schematu adresowania.

Adresy publiczne i prywatne

Jedna z najbardziej oczywistych decyzji, które musimy podjąć, jest wybór typu adresu sieci — prywatnej lub publicznej (internetowej). W większości przypadków będzie używany jeden z adresów sieci prywatnych w połączeniu z wybrana formą translacji adresów, aby zyskać dostęp do Internetu. Pozwoli to ukryć schemat adresowania przed Internetem i zredukować ryzyko włamania; pozwoli też zmniejszyć wydatki, ponieważ za używanie adresów prywatnych nie trzeba płacić. Jednym przypadem, gdy potrzebne będą poprawne internetowe adresy IP, są dostawcy usług internetowych (ISP), którzy muszą udostępnić klientowi poprawny adres. Wówczas używane są poprawne adresy publiczne.

Trzy grupy adresów zostały zarezerwowane w RFC 1918 dla sieci prywatnych. Adresy te nie są nigdy używane w Internecie. Oto te zakresy:

- ◆ od 10.0.0.0 do 10.255.255.255,
- ◆ od 172.16.0.0 do 172.31.255.255,
- ◆ od 192.168.0.0 do 192.168.255.255

Powód, dla którego te adresy nie są używane w Internecie, jest stosunkowo prosty. Gdyby na przykład serwer *mail.ditdot.com* posiadał adres 10.25.26.35 i próbowałibyśmy skontaktować się z nim z sieci prywatnej, która używa przestrzeni adresów 10.0.0.0, adres wyglądałby na lokalny a nie internetowy.

Wiele organizacji używa adresu 10.0.0.0, ponieważ daje on największą elastyczność. Można w tej sieci łatwo otrzymać dwa poziomy hierarchii, co pozwala na skonfigurowanie rytuału w obiektie lokalizacji i pomiędzy lokalizacjami. Mniejsze organizacje mogą wybrać adresy 172.16.0.0 lub 192.168.0.0 jeśli posiadają albo mała liczbę sieci, albo mała liczbę hostów w podsieci. Ogólnie mówiąc, wybór klasy adresu nie jest istotny, o ile podzielimy adres poprawnie.

Najprawdopodobniej potrzebna będzie jeszcze przestrzeń prawdziwych adresów. W większości przypadków kilka adresów będzie wykorzystanych dla wyeksploatowanych serwerów, na przykład pocztowego, DNS i WWW.

Uzyskanie adresu i połączenia z Internetem

Przestrzeń adresów dla Internetu jest pod kontrolą IANA (Internet Assigned Numbers Authority). Organizacja ta przydziela adresy różnym dostawcom usług internetowych i duzym organizacjom, które połączone są z Internetem bezpośrednio. Poza przypadkami, gdy sieć przylaczana jest do sieci szkieletowej, w istocie stojąc się jej częścią, otrzymamy adres IP od dostawcy usług internetowych (ISP).

Liczba potrzebnych adresów zależy od liczby systemów, które będziemy musieli udostępnić w Internecie. Dobra wiadomość może być fakt, iż zazwyczaj ISP jest w stanie dostarczyć dodatkowe adresy internetowe. Koszt adresów zwykle mieści się w usłudze świadczonej przez ISP. Jeśli jednak potrzebujemy dużej liczby adresów, być może będziemy musieli dodatkowo zapłacić.

Przy wyborze ISP nalezy wziac pod uwage kilka czynnikow. Ponizej przedstawilismy czesc pytan, ktore warto zadac:

- ♦ Jakie jest polaczenie ISP z Internetem? W wiekszosci przypadkow firma powinna brac pod uwage dostawce z sieci szkieletowej — na przyklad, MCI, UUNET lub BellNexxia — zamiast mniejszego dostawcy. Mniejsze firmy ISP i tak musza kupowac uslugi od tych wiekszych i mozemy wyeliminowac posrednika, udajac sie od razu do dostawcy w sieci szkieletowej. W niektórych przypadkach jednak warto wybrac ISP, który laczy sie z wieksza liczba dostawców w sieci szkieletowej (pierwszego poziomu) — poniewaz wiele polaczen zapewnia nadmiarosc i moze dac lepszy dostep wiekszej grupie osob.
- ♦ Ilu subskrybentow posiada ISP? Jednym z popularnych sposobow zarabiania, stosowanych przez malych dostawcow, jest nadsubskrypcja uslug. Zakladajac, ze nie wszyscy uzytkownicy caly czas beda korzystac z polaczenia, ISP moze bez problemow sprzedac 110% lub wiecej swojego pasma. Wraz ze wzrostem liczby subskrybentow zmniejsza sie pasmo dostepne dla przecietnego klienta.
- ♦ Jaki typ rozwiazan zapasowych posiada ISP? Nawet w przypadku awarii zasilania chcemy zachowac dostep do Internetu, wobec tego nalezy sprawdzic, czy ISP posiada system zasilania awaryjnego. Jesli dostawca (lub dowolne ognisko lancucha) nie ma zapasowego zasilania, lacze bedzie niedostepne.
- ♦ Jak wygladaja dodatkowe uslugi u ISP? W pewnych przypadkach moze przydac sie umieszczenie serwera WWW u dostawcy uslug, co zmniejszy potrzebna przepustowosc lacza do sieci lokalnej, poniewaz dostep do WWW nie bedzie musiał przez to lacze przechodzic. Jest to dobre rozwiazanie dla malych i srednich firm.
- ♦ Co ISP moze zaoferowac w dziedzinie polaczen przez linie telefoniczne? Wieksosc dostawców swiadczy również te uslugi. Wprawdzie polaczenie telefoniczne moze nie byc idealnym rozwiazaniem dla biura firmy, lecz w rzeczywistych warunkach ten typ dostepu potrzebny bedzie uzytkownikom pracujacym w domu i uzytkownikom laptopow. Jesli wszyscy uzytkownicy beda laczyc sie telefonicznie z dostawca, który przylacza nasze biuro, dane beda przechodzic po drodze przez mniejsza liczbe sieci..

Sa jeszcze inne pytania, które warto zadac przy wyborze ISP, zalezne od konkretnej sytuacji. Obecnie wieksosc ISP oferuje podobne uslugi i często wybór dostawcy zależy od ceny uslug. Prosze jednak pamietac, ze jakosc uslug i gotowosc ISP do współpracy z nasza organizacja sa rownie vazne.

Wybór wlasciwego ISP moze zredukowac naklady pracy i liczbe odbieranych zazalen. Dobry dostawca uslug internetowych moze uwolnic od zarzadzania siecia — zaczynajac od szacowania potrzeb adresowych oraz ustalenia potrzeb w zakresie podsieci i strategii rutingu.

Obliczanie potrzeb adresowych

Przedstawilismy jak dotad teoretyczne zagadnienia adresowania i pokrotce omówiliśmy, czego oczekiwac od ISP, wobec tego pora przejsc do faktycznych obliczen. Mu-

simy pogodzic sie z faktem, ze nie mozna przeciagnac jednego kabla pomiedzy wszystkimi posiadanimi systemami i nazwac wynik siecia (o ile nie uzywamy 50 lub mniej systemów).

Wprawdzie obliczanie potrzeb w zakresie adresów jest dosc late, lecz jednoczesnie niezwykle wagne. Zmiana schematu adresowania IP po uruchomieniu sieci jest duzym przedsięwzięciem, a poniewaz zadanie takie wymaga okresowego odłączania użytkowników, mozemy miec do czynienia z szeregiem zazaleń.

Jak juz wspomnialiśmy, adres IP sklada sie z dwóch części: adresu sieci i adresu hosta. W rzeczywistych warunkach jednakze trzeba dodac identyfikator podsieci w adresie IP, aby mozna bylo stosowac wewnętrzny ruting. Jesli pracujemy wylacznie w sieci LAN, potrzebny bedzie adres podsieci, który posluzy do ustalenia, w której sieci znajduje sie host. Jesli mamy do czynienia z wieloma lokalizacjami, potrzebny bedzie dodatkowo adres lokalizacji. W rezultacie, 32 bity skladowace sie na adres IP, moga zawierac cztery informacje: adresy sieci, lokalizacji, podsieci i hosta.

Aby ustalic, ilu adresów uzyc, bedziemy musieli przyjrzec sie sieci. Z ilu lokalizacji sklada sie teraz, a ile moze posiadac w przyszlosci? Znajac te liczby mozemy ustalic, ile bitów zarezerwowac na adres lokalizacji. Jesli siec posiada tylko jedna lokalizacja i razej nie bedzie nigdy miala wiecej, ten krok mozna pominiac.

Aby ustalic liczbe bitów potrzebnych na adres lokalizacji:

1. Ustal maksymalna liczbe lokacji, jaka bedzie kiedykolwiek potrzebna.
2. Przelicz wynik na system dwójkowy.
3. Policz liczbe zapisanych bitów.

Na przyklad, jesli obecnie dysponujemy 7 lokalizacjami i byc moze powstanie 5 kolejnych, musimy opracowac plan dla 12 lokalizacji. 12 w kodzie dwójkowym wynosi 1100 i liczbe te mozna zapisac w czterech bitach. W tym przypadku na adres lokalizacji potrzebne beda 4 bity.

W nastepnym kroku musimy ustalic maksymalna liczbe podsieci, jaka bedzie kiedykolwiek potrzebna w którejkolwiek lokalizacji. Do tego zadania mozemy podejsc w dwojak sposob. Pierwszy polega na arbitralnym ustaleniu liczby podsieci na podstawie fizycznego rozkladu sieci i miejsc, gdzie mozna by polaczyc systemy. Metoda ta czasami sie sprawdza, czasami nie. Drugi sposob polega na analizie ruchu w sieci (jak pokazalismy wczesniej) i uzywanej topologii, aby ustalic maksymalna liczbe systemów, które chcemy umiescic w kazdej podsieci. Liczba ta moze zostac nastepnie wzieta pod uwage w fizycznym rozkladzie podsieci i logicznym rozmieszczeniu użytkowników w podsieciach.

Przy ustalaniu maksymalnej liczby systemów w podsieci, musimy dodatkowo oszacowac poziom ruchu generowanego przez klienty (jak juz pokazalismy) i przyjrzec sie topologii.

Aby ustalic maksymalna dopuszczalna liczbe klientow w podsieci:

1. Ustal maksymalna przepustowosc w danej topologii. Dla sieci Ethernet 100 Mb/s bedzie to po prostu 100 megabitow na sekunde. Jesli jednak wszystkie stacje robocze sa przylaczone bezposrednio do przelacznikow, wartosc te mozemy podwoic, poniewaz system moze dzialac pelnodupleksowo do 200 Mb/s.
2. Podziel wynik przez 10, co da przybliziona przepustowosc w megabajtach na sekunde. Prawda, bajt ma tylko 8 bitow, lecz dzielnik 10 pozwala na wliczenie sekwencji wstepnej i CRC oraz pewnego poziomu kolizji (a takze jest wygodniejszy).
3. Pomnóż liczbe megabajtow na sekunde przez 3600 (liczbe sekund w godzinie). Tyle danych topologia jest w stanie przeslac w ciagu godziny.
4. Ustal objetosc ruchu sieciowego, jaka uzytkownik generuje w ciagu dnia. Mozesz to zrobic za pomoca monitora sieci lub po prostu szacujac wartosc wedlug tabeli 18.1. Zanotuj wartosc w megabajtach.
5. Pomnóż oszacowany wynik przez dwa, aby wziac poprawke na ruch „tla” — mozna go zmierzyc, lecz przy zalozeniu, ze sieciowe systemy operacyjne sa odpowiedzialne za 25% – 50% ruchu sieciowego, mozemy przyjac takie uproszczenie. W najgorszym przypadku zalozymy zbyt duza przepustowosc, co zawsze jest mile widziane.
6. Podziel liczbe z kroku 4. przez liczbe godzin roboczych w ciagu dnia (10). Teraz podziel liczbe z kroku 3. (objetosc ruchu, jaka siec moze przeslac w ciagu godziny) przez liczbe z kroku 5. (objetosc ruchu generowana prze uzytkownika w ciagu godziny). Wynikiem bedzie maksymalna liczba hostow, jaka w idealnych warunkach mozna bedzie przylaczyc do jednej podsieci.

Do oszacowania objetosci danych wygenerowanych przez uzytkownika w ciagu dnia moze posłuzyc tabela 18.1.

Suma tych szacunkowych wartosci da pewne pojecie o generowanej objetosci ruchu sieciowego. Ruch sieciowy wywolywany przez aplikacje nie jest tu wliczony, poniewaz kazda aplikacja generuje inne objetosci przesypanych danych i wartosci te trzeba zmierzyc. Ponadto, obliczenie nie bierze pod uwage ruchu sieciowego pochodzacego od aplikacji na pulpicie uruchamianych z serwera. Przy szacowaniu objetosci strony WWW prosze pamietac, ze wieksosc stron obecnie jest typu .ASP i jest tworzona dynamicznie. Strony takie nie sa buforowane po stronie klienta i musza byc odswiezane przy kazdych odwiedzinach.

Na potrzeby przykladu zalozmy, iz siec uzywa okablowania 100 Mb/s. Oszacujemy ruch sieciowy dla przecietnej stacji zakladajac 100 wiadomosci na dzien, z których 10% posiada zalaczniki. Przecietny rozmiar wiadomosci wynosi 750 bajtow (prosze pamietac o wliczeniu naglowka), zas przecietny zalacznik ma 35000 bajtow. W naszej sieci niech profile beda zapisywane lokalnie, a statystyczny uzytkownik bedzie uzywal 75 plikow dziennie, o przecietnej objetosci 80 kB. Kazdy uzytkownik odwiedza dziennie 150 stron WWW o typowej objetosci.

Najpierw obliczymy w tabeli 18.2 ruch sieciowy dla przecietnej stacji roboczej.

Tabela 18.1. Obliczenia ruchu sieciowego z jednej stacji roboczej

Poczta elektroniczna
A) liczba listów w ciągu dnia
B) przeciętna objętość listu w bajtach
C) odsetek listów z załącznikami
D) przeciętna objętość załącznika
E) objętość wiadomości w ciągu dnia ($A*B$)
F) objętość załączników w ciągu dnia ($A*C*D$)
G) ruch sieciowy generowany przez poczty elektroniczne, w megabajtach ($(E+F)/1024$)
Przesył plików (tylko jeśli użytkownicy składają pliki w serwerze)
H) objętość przeciętnego profilu w MB (jeśli użytkownicy korzystają z profili sieciowych)
I) przeciętna liczba plików przesyłanych dziennie
J) przeciętna objętość pliku w MB
K) ruch sieciowy wywoływany podczas przesyłania plików w MB ($H+(I*J)$)
Korzystanie z WWW
L) liczba stron odwiedzonych w ciągu dnia
M) przeciętna objętość strony w bajtach (domyślnie 10240)
N) dzienny ruch generowany przez WWW ($L*M/1024$)
O) całkowity ruch generowany przez stację roboczą ($G+K+N$)

Teraz możemy ustalić maksymalną liczbę systemów w podsieci, zgodnie z opisana powyżej procedura. Używana topologia jest Ethernet 100 Mb/s, który daje po podziale przez 10 około 10 megabajtów na sekundę. Jeśli teraz przemnożymy to przez 3600 (sekundy w godzinie), otrzymamy zdolność do przesyłania 36 000 megabajtów danych w ciągu godziny.

Biorąc 4143 MB wyliczone w tabeli 18.2 i mnożąc przez dwa (poprawka na ruch generowany przez serwery), otrzymamy 8 286 megabajtów na dzień (roboczy), czyli 828,6 megabajta na godzinę.

Teraz możemy wziąć objętość ruchu, jaką sieć może przesłać w ciągu godziny — 36 000 MB — i podzielić przez otrzymane 828,6 megabajta na godzinę dla pojedynczej stacji roboczej. Wyjdzie nam liczba stacji, jaką dana topologia może obsłużyć — w naszym przykładzie nieco ponad 43 stacje.

Następnie możemy otrzymać wymaganą liczbę podsieci, dzieląc liczbę posiadanych systemów przez 43. Na przykład, jeśli organizacja posiada 2 394 systemy, powinnismy planować około 55 podsieci.

Tabela 18.2. Przykładowe obliczenie ruchu sieciowego

Poczta elektroniczna	
A) liczba listów w ciągu dnia	100
B) przeciętna objętość listu w bajtach	750
C) odsetek listów z załącznikami	10%
D) przeciętna objętość załącznika	35 000
E) objętość wiadomości w ciągu dnia ($A \cdot B$)	75 000
F) objętość załączników w ciągu dnia ($A \cdot C \cdot D$)	2 625 000
G) ruch sieciowy wytworzony przez poczty elektroniczne, w megabajtach ((E+F)/1024)	2 637
Przesył plików (tylko jeśli użytkownicy składają pliki w serwerze)	
H) objętość przeciętnego profilu w MB (jeśli użytkownicy korzystają z profili sieciowych)	n/d
I) przeciętna liczba plików przesyłanych dziennie	75
J) przeciętna objętość pliku w MB	0,08
K) ruch sieciowy generowany podczas przesyłania plików w MB ($H + I \cdot J$)	6
Korzystanie z WWW	
L) liczba stron odwiedzonych w ciągu dnia	150
M) przeciętna objętość strony w bajtach (domyślnie 10240)	10 240
N) dzienny ruch generowany przez WWW ($L \cdot M / 1024$)	1 500
O) całkowity ruch generowany przez stację roboczą ($G + K + N$)	4 143

Następnym krokiem jest ustalenie liczby bitów, wymaganych do zaadresowania tej liczby podsieci. Wynik dodamy do liczby bitów zajętych już na adresy lokacji. Ponieważ potrzeba nam 55 podsieci, możemy zapisać 55 w systemie dwójkowym (110111) i policzyć bity (7).

Za pomocą tej samej procedury, z której właśnie skorzystaliśmy dla części adresu IP zarezerowanej na podsieć, możemy ustalić ile bitów potrzeba na ID hosta. Jak wyszło z obliczeń, każda podsieć powinna zawierać nie więcej niż 43 hosty. Zapisując 43 w systemie dwójkowym otrzymamy 101011, czyli 7 bitów.

Obliczyliśmy już wszystkie potrzebne składniki adresu IP: 4 bity na lokalizację, 7 na ID podsieci i 7 bitów dla hosta. Gdy dodamy te liczby, okazuje się, że do funkcjonowania naszej sieci potrzeba 18 bitów w części adresu IP przeznaczonej na adres hosta. Oznacza to, że musimy użyć adresu klasy A, który udostępnia 24 bity na adres hosta (klasa B pozwala na 16 bitów, a klasa C tylko na 8).

Musimy dla każdej lokalizacji użyć odrebnego adresu klasy B. Bez czterech bitów dla lokalizacji będziemy potrzebować tylko czternastu. Do tego wystarczy 16 bitów udostępnianych przez adres klasy B; jednakże każda lokalizacja wymaga osobnego adresu klasy B — w przeciwnym razie nie byłoby możliwe trasowanie pomiędzy oddziałami. Gdyby wszystkie używały tego samego adresu sieci, ich rozróżnienie nie byłoby możliwe.

Nastepnym krokiem bedzie utworzenie schematu podsieci klasy A (masek podsieci), które obsluga nasza siec.

Podzial na podsieci

Podzial na podsieci (*subnetting*) jest zagadnieniem, które przez lata dezorientowalo i zadziwialo wielu ludzi. Zadaniem niniejszego podrozdzialu jest objasnienie podzialu na podsieci w prosty, mamy nadzieje, sposob. Ostrzegamy: jesli Czytelnik nie jest dobrze zaznajomiony z systemem dwójkowym, moze troche zabolec.

Jak mówilismy w rozdziale 5., z adresu IP mozna wydobyc identyfikator hosta za pomocą maski podsieci. ID sieci wydobyty z adresu pozwala ustalic, czy adres docelowy jest lokalny czy zdalny — na podstawie tej informacji pakiety sa różnie traktowane. Pokazalismy tez trzy standardowe maski podsieci: 255.0.0.0, 255.255.0.0 i 255.255.255.0.

Te standardowe maski podsieci sluza do zasloniecia czesci adresu IP przypadajacej na hosta, aby mozna bylo ustalic adres sieci i odpowiednio przeslac do niej pakiet. I poniewaz te maski odpowiadaja wykorzystaniu do identyfikacji adresu sieci 8, 16 lub 24 bitów adresu IP, zalaczane lub wylaczane sa całe oktety, dzieki czemu mozemy pracowac z liczbami łatwymi do przeliczania.

Obliczanie ID lokalizacji

W naszym przykladzie uzyliśmy 4 bitów dla lokalizacji, 7 dla podsieci i jedynie 7 dla hosta. Oznacza to, ze podzial na podsieci nie zostanie przeprowadzony w oktetach (grupach 8 bitów). Musimy wiec ustalic własna maske podsieci, która bedzie nadawac sie dla naszej organizacji. W rzeczywistosci beda nam potrzebne dwie maski — jedna pomiędzy lokalizacjami i jedna uzywana we wszystkich lokalizacjach.

Przyjrzyjmy sie procesowi ustalania, czy adres jest lokalny, czy zdalny. Tabela 18.3 przedstawia obliczenia dla hosta o adresie IP 158.35.64.7 i masce podsieci 255.255.0.0 (standardowa maska podsieci klasy B), próbujacego skomunikowac sie z hostem o adresie IP 158.35.80.4.

Tabela 18.3. Ustalenie, czy host jest lokalny, czy zdalny

Pozycja	Notacja dziesietna rozdzielona kropkami	Postać dwójkowa
Lokalny adres IP	158.35.64.7	10011110 00100011 01000000 00000111
Maska podsieci	255.255.0.0	11111111 11111111 00000000 00000000
ID sieci	158.35.0.0	10011110 00100011 00000000 00000000
Docelowy adres IP	158.35.80.4	10011110 00100011 01010000 00000100
Maska podsieci	255.255.0.0	11111111 11111111 00000000 00000000
ID sieci	158.35.0.0	10011110 00100011 00000000 00000000

W tabeli 18.3 dwa uzyskane identyfikatory sieci sa takie same, wiec system jest lokalny. Jak widac, funkcja AND bardzo łatwo wydobywa identyfikator sieci. W naszym przypadku musimy jednakże utworzyć wiele różnych sieci, wiec nie da się zastosować tej standardowej maski podsieci.

Przejdzmy teraz do poprzedniego przykładu, który wymaga 4 bitów na lokalizację, 7 bitów na podsiec i 7 bitów na hosta. Musimy najpierw podjąć decyzje o wyborze adresu sieci prywatnej. Wracając do dostępnych możliwości przypomnijmy, że dostępne są: klasa A i grupy klas B i C (o adresach odpowiednio zaczynających się od oktetu 10, 172 i 192).

W naszym przykładzie potrzeba 7 bitów na hosta, 7 na podsiec i 4 na lokalizację — w sumie 18 bitów. Ponieważ nie możemy zmienić bitów w danym adresie bez zmiany samego adresu, musimy utworzyć podsieci w części adresu przypadającej na hosta. Przestrzeń adresowa klasy B udostępnia 16 bitów na hosta (2 oktety), zaś klasa C tylko 8. Oznacza to, że musimy użyć adresu klasy A — adresu sieci prywatnej 10.0.0.0.



Uwaga

W rzeczywistych warunkach większość organizacji używa adresu sieci klasy A, ponieważ wybór daje największe możliwości rozbudowy. W większości przypadków drugi oktet jest używany na lokalizację, trzeci na ID podsieci, a ostatni na adres hosta. Oznacza to, że każda firma posiadająca nie więcej niż 256 lokalizacji z 256 (lub mniej) podsieciami w każdej lokalizacji może korzystać z adresu 10.0.0.0.

W rzeczywistości każda lokalizacja jest odrebną siecią, zaś w każdej lokalizacji wszystkie podsieci również stanowią odrebną sieć. Oznacza to, że musimy zachować większą część adresu na sieć, a mniejszą na hosta.

W standardowej masce podsieci klasy A bit, które reprezentują adres sieci, są złączone (1), zaś bity hosta wyłączone (0).

11111111	00000000	00000000	00000000
10	0	0	0

Jesli więc potrzebujemy więcej sieci, kolejne bity w masce podsieci zostaną użyte na sieci — to znaczy, więcej kolejnych bitów będzie złączonych (jedynki). Gdy dodamy cztery bity przeznaczone do określenia lokalizacji, maska podsieci będzie wyglądała tak:

11111111	11110000	00000000	00000000
----------	----------	----------	----------

Po przeliczeniu wartości dwójkowej z powrotem na dziesiętną, nowa maska podsieci będzie miała wartość 255.240.0.0. Następnym etapem będzie znalezienie dla każdej lokalizacji początkowego adresu IP, który możemy nazwać identyfikatorem lokalizacji. ID lokalizacji będzie zaczynać się od 10. Ponieważ 10. jest przydzielona części adresu, w przypadku prawidłowych internetowych adresów IP wystarczy zastąpić 10 przydzielonym adresem sieci. W naszym przykładzie ID lokalizacji mieści się cały w drugim oktacie, dzięki czemu wiemy, że tylko wartości drugiego oktetu będą się zmieniać. Dla wszystkich ID lokalizacji dwa ostatnie oktety będą miały wartość 0.0.

Wielu ludzi ma kłopoty ze zrozumieniem, że nie każda zmiana wartości drugiego oktetu będzie oznaczać nową podsieć. W naszym przykładzie 10.14.0.0 i 10.15.0.0 mieszczą się w jednej lokalizacji, lecz 10.16.0.0 już będzie należać do innej. Ostatni adres należy

do innej lokalizacji, poniewaz wzór pierwszych czterech bitów drugiego oktetu zmienia sie z 0000 w przypadku 10.14.0.0 i 10.15.0.0 na 0001 w przypadku 10.16.0.0.

Oznacza to, ze musimy ustalic, jakie liczby powodują zmiany w pierwszych czterech bitach. Kazda wartosc, która powoduje te zmiany, bedzie osobnym ID lokalizacji. Oczywiście mozemy zapisac wszystkie liczby od 0 do 255 w kodzie dwójkowym i wyszukac zmiany pierwszych czterech bitów w otrzymanej liscie, lecz zajeloby to troche czasu! Jest szybsza metoda: znalezc, gdzie jest ostatni bit „1” w masce podsieci i ustalic wartosc tej kolumny. Przy wykorzystanych czterech bitach ostatni znajduje sie w czwartej kolumnie oktetu. Wartosci kolumn w oktecie od lewej wynosza 128, 64, 32, 16, 4, 2 i 1, wobec tego czwarta kolumna ma wartosc *przyrostem*.

Dysponujac wartoscia przyrostu mozemy szybko ustalic wszystkie ID lokalizacji. Za- czniemy od 0 i bedziemy zwiększac oktet o 16. Oznacza to, ze identyfikatorami lokalizacji beda 10.0.0.0, 10.16.0.0, 10.32.0.0, 10.48.0.0, 10.64.0.0 i tak dalej. Powód tego skrótu jest prosty. Istnieje ograniczona liczba kombinacji pierwszych czterech bitów oktetu. W zapisie dwójkowym wygladaja tak:

```
0000  
0001  
0010  
0011  
0100  
0101  
0110  
0111  
1000  
1001  
1010  
1011  
1100  
1101  
1110  
1111
```

W istocie jest to lista liczb od 0 do 15, zapisanych dwójkowo. Kazda liczba jest wieksza o 1 od poprzednika — inaczej mówiac, kazda kolejna liczba przerasta o 1. Jednakze te cztery bity leza na poczatku oktetu, nie na koncu, na którym mieści sie kolumna jedynek. Faktycznie wiec bedziemy szukac takich liczb:

```
0000 0000  
0001 0000  
0010 0000  
0011 0000  
0100 0000  
0101 0000  
0110 0000  
0111 0000  
1000 0000  
1001 0000  
1010 0000  
1011 0000  
1100 0000  
1101 0000  
1110 0000  
1111 0000
```

Prosze zwracic uwage, ze dodalismy jedynie zera na koniec. Zmiany w czterech pierwszych bitach sa dokladnie takie same. Jedyna różnica w porównaniu z pierwsza lista jest to, ze nie zwiększymy kolumny jedynek, lecz czwarta kolumna o wagę 16. Gdybysmy musieli zamiast czterech bitów uzyć trzech, szukalibysmy tych liczb:

```
000 00000
001 00000
010 00000
011 00000
100 00000
101 00000
110 00000
111 00000
```

Używając trzech bitów w rzeczywistości jedynie zwiększymy o 1 trzecią kolumnę. Wobec tego, dla trzech bitów przyrost wynosi 32, zas ID lokalizacji wynoszą 10.0.0.0, 10.32.0.0, 10.64.0.0, 10.96.0.0 i tak dalej. Ta uproszczona metoda nadaje się dla dowolnej liczby użytych bitów.

Niezależnie od liczby bitów, które zostały użyte, możemy wykorzystać tabelę 18.4 do skojarzenia maski podsieci z przyrostem. I bardzo dobrze, bo oszczędzi to nam grzebania się w liczbach dwójkowych.

Tabela 18.4. Możliwe maski podsieci i związane z nimi przyrosty

Maska podsieci	Wartość dwójkowa	Wartość kolumny (przyrost)
255.0.0.0	11111111 00000000 00000000 00000000	nie dotyczy
255.128.0.0	11111111 10000000 00000000 00000000	128
255.192.0.0	11111111 11000000 00000000 00000000	64
255.224.0.0	11111111 11100000 00000000 00000000	32
255.240.0.0	11111111 11110000 00000000 00000000	16
255.248.0.0	11111111 11111000 00000000 00000000	8
255.252.0.0	11111111 11111100 00000000 00000000	4
255.254.0.0	11111111 11111110 00000000 00000000	2
255.255.0.0	11111111 11111111 00000000 00000000	1

Czytelnik powinien już rozumieć, jak obliczane są ID lokalizacji. Oto pełna lista identyfikatorów lokalizacji dla przykładowej sieci, której używamy:

```
10.0.0.0
10.16.0.0
10.32.0.0
10.48.0.0
10.64.0.0
10.80.0.0
10.96.0.0
10.112.0.0
10.128.0.0
10.144.0.0
10.160.0.0
10.176.0.0
```

```

10.192.0.0
10.208.0.0
10.224.0.0
10.240.0.0

```

Obliczanie ID podsieci

Mamy juz różne ID lokalizacji, wiec mozemy dodac kolejne 7 bitów, które podziela kazda lokalizacje na podsieci. Oznacza to zmiane ostatnich czterech bitów drugiego oktetu i pierwszych trzech bitów trzeciego oktetu w masce podsieci na 1. Nowa maska podsieci bedzie miala wartosc 11111111.11111111.11100000.00000000, czyli 255.255.224.0 w notacji dziesietnej z kropkami.

Zalozmy, ze ID lokalizacji sieci 10.32.0.0 przydzielimy do biura w Atenach w Grecji. Adresy wszystkich podsieci beda zaczynac sie od 10.32.0.0. Ustalimy ID podsieci zwiększąc ten adres o 32 w trzecim oktacie (jak widac w tablicy 18.4, przyrost dla maski podsieci 224 wynosi 32).

Stosując taka sama strategie jak uprzednio, uzyskamy podsieci 0, 32, 64 i tak dalej. Oznacza to, ze pierwsza podsiec w biurze w Atenach bedzie miala adres 10.32.0.0, co moze byc mylące — poniewaz 10.32.0.0 jest również ID lokalizacji. Nie wykorzystamy wiec 10.32.0.0 jako ID podsieci; zamiast tego zaczniemy od 10.32.32.0. Aby utrzymać kolejne ID podsieci, bedziemy dodawac przyrost do adresu. Kolejnych kilka podsieci to 10.32.64.0, 10.32.96.0, 10.32.128.0 i tak dalej.

Jaki wiec adres nastapi po 10.32.244.0 Aby odpowiedziec na to pytanie, musimy zignorowac granice narzucone przez kropki w notacji dziesietnej. Po prostu bedziemy dalej zwiększać wartości dwójkowe, jak przedtem (patrz tabela 18.5).

Tabela 18.5. Przekraczanie granic między oktetami

Przydzielona podsiec	ID lokalizacji	ID podsieci	ID hosta
00001010	0010	0000 111	00000 00000000
00001010	0010	0001 000	00000 00000000
00001010	0010	0001 001	00000 00000000

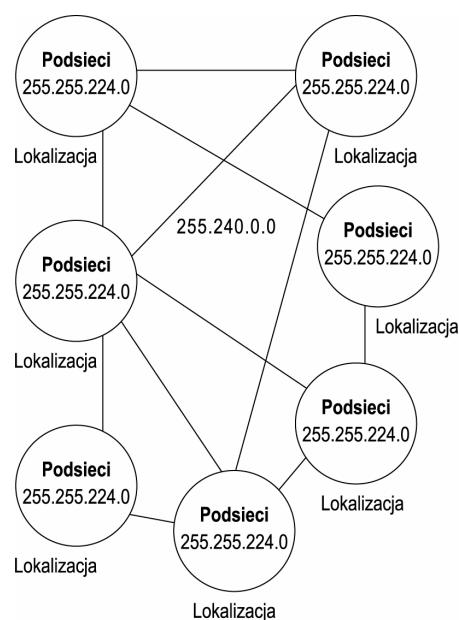
Następnymi ID podsieci po 10.32.224.0 bedą 10.33.0.0 i 10.33.32.0. Prosze pamietac, ze nastepna wartosc, która zmieni wzór pierwszych czterech bitów drugiego oktetu — czyli w istocie zmieni ID lokalizacji — wynosi 48. Bedziemy wiec wykorzystywac w Atenach ID podsieci posiadajace w drugim oktacie wartosci 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46 i 47. Dopiero dla adresów od 48 bedziemy miec do czynienia z inną lokalizacją.

Prosze zwrócić uwagę, ze w tabeli 18.5 adres jest podzielony na cztery różne fragmenty, z którymi tu pracujemy: pierwsze 8 bitów jest oktetem przydzielonego adresu, cztery pierwsze bity drugiego oktetu sa przeznaczone na ID lokalizacji, 7 bitów z drugiego i trzeciego oktetu jest przeznaczonych na ID podsieci. Pozostale 13 bitów pozostaje na ID hosta. Jest to o wiele wiecej, niz nam trzeba, lecz i tak musimy wykorzystac wszystkie 32 bity.

W przykładowej sieci mogliśmy po prostu wykorzystać drugi oktet na lokalizację, trzeci na ID podsieci i ostatni na ID hosta. Zaoszczędziliśmy czasu na obliczenia i znacznie uprościło adresowanie, ponieważ granice pomiędzy różnymi składnikami adresu zgadzały się z granicami oktetów. Na dodatek takie rozwiązanie zostało wiecej miejsca na rozbudowę sieci. W chwili obecnej jedynym miejscem na rozbudowę jest ID hosta. Lepiej zostawić wolne miejsce dla nowych ID lokalizacji i podsieci, ponieważ liczba hostów w pojedynczej podsieci jest ograniczona fizycznie.

Mamy teraz dwie maski podsieci, których możemy używać: jednej dołączenia ze sobą biur oraz drugiej w każdym biurze, jak na rysunku 18.1.

Rysunek 18.1.
Uproszczony schemat
przykładowej sieci



Jak widać, maska podsieci 255.255.224.0 jest używana w każdej lokalizacji, natomiast 255.240.0.0 pomiędzy lokalizacjami. Zapewnia to, że każdy host w sieci będzie „widzialny” pozostałe podsieci z tej lokalizacji jako zdalne. Ponadto taki schemat adresowania powoduje, że jedna lokalizacja rozpoznaje wszystkie pozostałe jako sieci odległe.

Ustalenie adresów hostów

Obliczyliśmy już ID lokalizacji i wiemy jak obliczyć ID podsieci w każdej lokalizacji. Teraz musimy ustalić zakres adresów hostów dla każdej sieci. Wracając do biura w Atenach, użyjemy jako przykładu podsieci 10.32.32.0.

Adres 10.32.32.0 ma pełne 32 bitów. Nie możemy jednak uzyc ich wszystkich dla hosta, ponieważ będą potrzebne ruterom do zbudowania tablic tras w sieci. Adres 10.32.32.0 jest w rzeczywistości „nazwą” całej podsieci, w której wszystkie bity ID hosta wynoszą 0. Zastosujemy ponownie metodę przyrostów, tym razem jednak tylko proste zwiększenie wartości o 1. Ponieważ ID hosta zawsze mieści się na koncu adresu IP, zawsze będzie kończyć się w ostatniej kolumnie, której waga wynosi 1.

Dla pierwszego hosta dodamy 1 do identyfikatora podsieci równego 10.32.32.0, co da w wyniku 10.32.32.1. Kolejny host otrzyma adres 10.32.32.2 i tak dalej. Podobnie jak poprzednio, gdy dojdziemy do adresu 10.32.32.255, przekroczymy granice oktetu. Następny host bedzie posiadal adres 10.32.33.0, kolejny 10.32.33.1 i tak dalej.

Adres, w którym czesc przypadajaca na hosta sklada sie z samych zer, jest ID podsieci i nie moze zostac wykorzystany na adres hosta, poniewaz posiada specjalne znaczenie. Istnieje jeszcze jeden adres w czesci hosta, którego nie mozna wykorzystac — adres rozgloszeniowy skladowy sie z samych jedynek. W naszym przykładzie jest to:

00001010 001000001 00111111 11111111

W notacji dziesietnej rozzielonej kropkami bedzie to 10.32.63.255. Prosze zwrócić uwagę, ze dodajac 1 do tej liczby otrzymamy ID nastepnej podsieci — 10.32.64.0. Tak bedzie zawsze i mozemy w ten sposob łatwo znalezc adres rozgloszeniowy, odejmujac 1 od kolejnego adresu podsieci.

Dla podsieci 10.32.32.0 mamy ID podsieci równy 10.32.32.0, adres rozgloszeniowy 10.32.63.255 oraz pierwszy poprawny adres hosta 10.32.32.1. Poniewaz tylko pierwszy adres hosta (same zera) i ostatni (same jedynki) maja specjalne znaczenie, ostatni poprawny adres hosta jest mniejszy od rozgloszeniowego dla podsieci o jeden. W naszym przykładzie bedzie to 10.32.63.254.

Rzut oka na nadsieci

Tworzenie *nadsieci* (*supernetting*), inaczej *bezklasowy ruting domen internetowych* (CIDR — *Classless Internet Domain Routing*) jest w zasadzie odwróconym procesem podzialu na podsieci. Przy ograniczonej liczbie internetowych adresów w klasach A (216 adresów) i B (16 384 adresy) pojawiły się problemy z przydzieleniem adresów IP firmom, które posiadają wiecej hostów niż 254, dopuszczalne w klasie C.

Problem został w duzym stopniu rozwiązany przez wykorzystanie prywatnych adresów sieciowych i serwerów proxy. Te dwie techniki daly przedsiębiorstwom dowolna potrzebna liczbę adresów wewnętrznych z wykorzystaniem poprawnych adresów internetowych. Zdarza się jednak przypadki (na przykład dostawcy usług internetowych lub duże przedsiębiorstwa), które wymagają przydzielenia dużych bloków poprawnych adresów internetowych. Bez CIDR te firmy musiałyby obyć się adresami z klasy C.

CIDR pozwala na połączenie wielu małych sieci w jedną dużą. Na przykład, jeśli firma potrzebuje 620 adresów internetowych, to potrzebne jej będą przynajmniej 3 adresy klasy C.

Aby zrozumieć CIDR, musimy ponownie porzucić sztuczne granice, narzucone przez notację dziesiętną rozzieloną kropkami, i spojrzeć na adresy IP jak na 32-bitowe liczby dwójkowe. Jeśli potraktujemy grupę sieci klasy C jak podsieć adresu klasy B, problem zostanie uproszczony.

Jeśli firma potrzebuje 620 poprawnych adresów IP, możemy na to popatrzyć po prostu jak na 620 hostów. W notacji dwójkowej liczba ta wynosi 10 01101100 i ma długość dziesięciu bitów. W adresie klasy B byłoby to proste: jeśli potrzebujemy 10 bitów na hosty, to ID podsieci będzie miał 6 bitów. Maska podsieci w tym przypadku będzie 255.255.252.0.

Poniewaz potrzeba nam poprawnych adresów internetowych, musimy skorzystac z uslug ISP, któremu organizacja IANA przyznala duzy blok internetowych adresów IP. Dostawca w rzeczywistosci wykona te obliczenia i znajdzie zakres adresów klasy C, które beda funkcjonowac razem jak podsiec klasy B.

Jesli, na przykład, ISP otrzymał zakres adresów IP od 207.236.0.0 do 207.236.255.255, potraktuje go jak adres sieci klasy B — 207.236.0.0. Od tego momentu proces jest podobny do znajdowania ID podsieci dla tej sieci „klasy B” z maska podsieci 255.255.252.0. W naszym przypadku przyrost wynosi 4, wiec ISP poszuka na przykład 207.236.48.0, 207.236.49.0 i 207.236.50.0. Jesli wszystkie adresy w tym zakresie beda wolne, ISP bedzie mógł przydzielic adres 207.236.48.0 z maska podsieci 255.255.252.0. Oznacza to dokladnie mówiac 1022 poprawne adresy, lecz na potrzeby trasowania musi zostać przydzielony caly blok.

Teraz pakiety usilujace znalezc nasza siec beda wysylane do sieci 207.236.0.0 klasy B naszego ISP, który ustali, ze adres nalezy do podsieci 207.236.48.0 i przesle pakiet do naszego głównego rutera. W nim przydzielony adres nalezy juz do podsieci i ruter przesle go we właściwe miejsce.

CIDR zasadniczo pozwala uzywac dowolnej klasy adresu w roli dowolnej innej klasy adresu i dzielic (lub laczyc) adresy w sposób najlepiej pasujacy do naszych potrzeb.

Rozdział 19.

Projektowanie trasowania dla sieci

W tym rozdziale:

- ◆ Podstawy trasowania
- ◆ Tworzenie struktury trasowania
- ◆ Trasowanie dynamiczne

W rozdziale 18. omówiono podział na podsieci i wydobywanie identyfikatora podsieci z adresu IP i maski podsieci. Teraz zobaczymy, jak można zastosować te mechanizmy. Kilka razy już kładliśmy nacisk na fakt, iż nie da się umieścić nieograniczonej liczby komputerów w jednym segmencie sieci — wobec tego musimy podzielić sieć na mniejsze, łatwiejsze do opanowania fragmenty.

Niniejszy rozdział omawia podstawy trasowania i wyniki dodawania coraz większej liczby segmentów do sieci. Do innych zagadnień omówionych tutaj należą: trasowanie klasowe i bezklasowe, maski podsieci o zmiennej długości i różne metody automatycznej wymiany informacji o trasach pomiędzy różnymi routery w sieci, zamiast recznej konfiguracji informacji.

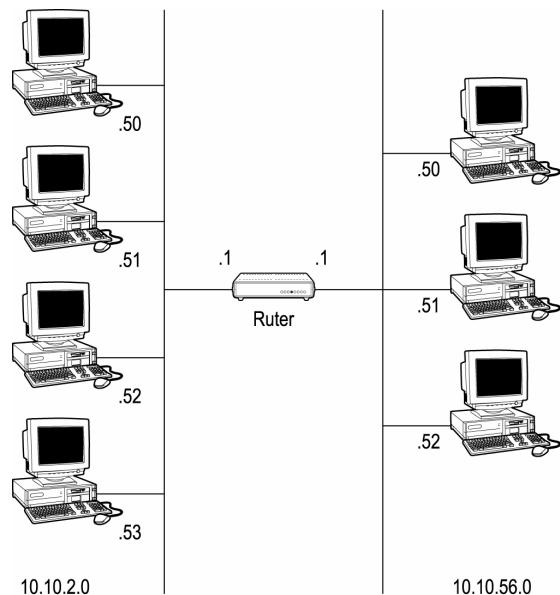
Podstawy trasowania

Trasowanie (*routing*) jest procesem przesyłania informacji z sieci źródłowej do docelowej poprzez dowolne urządzenie, które posiada dwa lub więcej interfejsów sieciowych i stos IP. Inaczej mówiąc, urządzenie „przygląda się” każdemu odebranemu pakietowi i ustala, czy ten pakiet wedruje do sieci, do której dane urządzenie jest przylądkowane fizycznie. Jeśli tak, to pakiet zostanie wysłany na dany interfejs lokalny. W przeciwnym razie ruter szuka trasy, której może użyć. Ogólnie mówiąc, ruter sprawdza, czy jeden z routerów, z którym jest połączony, potrafi przesłać pakiet dalej.

Ten mechanizm jest kamieniem wegielnym trasowania i sieci z wyborem tras: routery przesyłają pakiet po jednym przeskoku na raz (przeskok taki nazywany jest *hopem*), aż do osiągnięcia miejsca przeznaczenia. Sam ruter jest bardzo prostym urządzeniem. W rzeczywistości dowolne urządzenie, posiadające dwa lub więcej interfejsów sieciowych i warstwę protokołu IP, można zmusić do trasowania pakietów.

Załóżmy, że sieć została podzielona na dwie części (na potrzeby przykładu wybierzymy prosty podział na podsieci). Pierwsza sieć ma adres 10.10.2.0, zas druga 10.10.56.0. W obu maskach podsieci jest 255.255.255.0 (patrz rysunek 19.1).

Rysunek 19.1.



Jak widac, dwie sieci sa rozdzielone ruterem, który posiada dwa interfejsy sieciowe; po jednym dla kazdej sieci: 10.10.2.1 i 10.10.56.1. Gdy host, na przyklad 10.10.2.51, próbuje wyslac pakiet do 10.10.2.53, wówczas warstwa internetowa z IP lokalnego hosta i maski podsieci otrzymuje ID sieci 10.10.2.0. Poniewaz host docelowy moze miec inną maskę podsieci, warstwa internetowa uzywa IP adresata i własnej maski podsieci (jednej, która zna) i uzyskuje prawdopodobny ID sieci hosta docelowego. W naszym przykładzie uzyskamy ID sieci 10.10.2.0, zgodny z własnym ID sieci nadawcy. Poniewaz identyfikatory sieci dla nadawcy i odbiorcy sa takie same, IP rozpoznaje, ze host docelowy jest lokalny, wiec za pomoca protokolu ARP (*Address Resolution Protocol* — protokół rozwiązywania adresu) znajduje adres sprzetowy karty interfejsu sieciowego adresata i wysyla pakiet bezposrednio do hosta.

Jak dotad ruter nie bral udzialu w transmisji i nie wystapilo trasowanie. Gdyby adres IP adresata nalezel do drugiej podsieci, na przyklad 10.10.56.52, wówczas ruter zostalby zaangazowany w transmisje. Ponownie IP i maska podsieci lokalnego hosta sluzą do ekstrakcji ID lokalnej podsieci — 10.10.2.0. Jednakze po nalozeniu lokalnej maski podsieci na IP adresata uzyskany zostanie ID sieci 10.10.56.0. Ten identyfikator nie jest zgodny z lokalnym, wobec czego pakiet trzeba bedzie trasowac. Proces trasowania za- czyna sie w lokalnej stacji roboczej za pomoca tablicy tras.

Tablica tras

Trasowanie w rzeczywistości zaczyna się już w lokalnym komputerze, który posiada specjalna tablice, zwana tablica tras. Jest ona tworzona przy każdym uruchomieniu systemu i służy IP do trasowania pakietów. W przypadku komputera lokalnego tablica tras

zwykle zawiera wpisy lokalne i jeden wpis przesyłający cały pozostały ruch do lokalnego rutera. Poniżej przedstawiliśmy przykład tablicy tras, jaką może znajdować się w hostie 10.10.2.51:

Metryka	Adres sieciowy	Maska sieci	Adres bramy	Interfejs
1	0.0.0.0	0.0.0.0	10.10.2.1	10.10.2.51
1	10.10.2.0	255.255.255.0	10.10.2.51	10.10.2.51
1	10.10.2.51	255.255.255.255	127.0.0.1	127.0.0.1
1	10.255.255.255	255.255.255.255	10.10.2.51	10.10.2.51
1	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1
1	224.0.0.0	224.0.0.0	10.10.2.51	10.10.2.51
1	255.255.255.255	255.255.255.255	10.10.2.51	10.10.2.51
1				

Na pierwszy rzut oka ta tablica może wyglądać na zagmatwana, lecz tak naprawdę kiedy to ją zrozumieć. Docelowy adres IP jest łączony z maską sieci, aby wydobyć ID sieci. Jeśli uzyskany ID sieci jest zgodny z docelową siecią, to znajdziemy trasę i możemy wysłać dane za pomocą lokalnego interfejsu sieciowego, wymienionego w kolumnie *Interfejs*, pod adres z kolumny *Adres bramy*.

W tablicy tras pierwszy wpis od góry to 0.0.0.0 zarówno w kolumnie *Maska sieci*, jak i w docelowym adresie sieciowym. Użycie tej maski z dowolnym adresem IP zawsze da jako sieć docelową 0.0.0.0. Umieszczenie tego wpisu na początku tablicy może wyglądać dziwnie, jednakże tablica tras jest czytana w odwrotnej kolejności (z dolu do góry). W tym przypadku system na początku sprawdza, czy pakiet jest wysłany na globalny adres rozgłoszeniowy (255.255.255.255), następnie sprawdza adres grupowy (224.0.0.0), a na koniec adres petli zwrotnej (127.0.0.1).

Po sprawdzeniu tych adresów system wymienia adresy dla każdego interfejsu lokalnego (jeśli komputer posiada więcej kart sieciowych, wiersze te będą powtarzane dla każdego interfejsu). Dla każdego lokalnego interfejsu najpierw szuka adresu rozgłoszeniowego podsieci (.255), następnie wszelkich pakietów kierowanych do lokalnego hosta, a na koniec wszystkiego, co wychodzi do lokalnej podsieci. Sprawdzanie lokalnego adresu IP może wydać się nadmiarowe, biorąc pod uwagę, że system już sprawdził, iż pakiet przeznaczony jest dla sieci lokalnej; jednakże system z wieloma kartami sieciowymi wymaga sprawdzania lokalnych interfejsów, aby pakiety odebrane na jednym interfejsie (gdzie została porównana z IP i maską podsieci tego interfejsu) mogły być wysłane z innego interfejsu.

Wpis 0.0.0.0, będący tzw. *trasą domyslną (default route)*, wychwytuje wszystko i pojawia się tylko wtedy, gdy system posiada skonfigurowaną bramę domyslną. Jeśli nie zostanie znaleziona żadna trasa podczas porównywania przez system docelowego adresu IP z każdą maską sieci w tablicy tras, to trasa domyslna posłuży do wysłania pakietu do skonfigurowanego routera.

Wracając do naszego przykładu (w którym 10.10.2.51 chce skomunikować się z 10.10.56.52), system lokalny używa pokazanej przed chwilą tablicy tras, aby ustalić

kolejny przeskok (hop). W tym przypadku zostanie uzyta trasa domyslna, a system za pomoca protokolu ARP znajdzie adres sprzetowy karty 10.10.2.1. Pakiet zostanie nastepnie przeslany do hosta o adresie IP 10.10.2.1, który w naszym przypadku jest ruterem.

Ruter odbierze pakiet i porowna go z adresem IP i maska podsieci lokalnego interfejsu. W tym przypadku ruter porowna pakiet skierowany do 10.10.56.52 z adresem IP 10.10.2.1. za pomoca maski podsieci 255.255.255.0. Podobnie jak przed chwila, wynik porownania oznacza, ze host docelowy jest zdalny (dla danego interfejsu) i zmusi ruter do sprawdzenia swojej tablicy tras.

Metryka	Adres sieciowy	Maska sieci	Adres bramy	Interfejs
1	10.10.2.0	255.255.255.0	10.10.2.1	10.10.2.1
1	10.10.2.1	255.255.255.255	127.0.0.1	127.0.0.1
1	10.255.255.255	255.255.255.255	10.10.2.1	10.10.2.1
1	10.10.56.0	255.255.255.0	10.10.56.1	10.10.56.1
1	10.10.56.1	255.255.255.255	127.0.0.1	127.0.0.1
1	10.255.255.255	255.255.255.255	10.10.56.1	10.10.56.1
1	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1
1	224.0.0.0	224.0.0.0	10.10.2.51	10.10.2.51
1	255.255.255.255	255.255.255.255	10.10.2.51	10.10.2.51

W tym przykladzie IP odkryje, ze trasa 10.10.56.0 wspolpracuje z maska 255.255.255.0. System przesle wiec pakiet do bramy 10.10.56.1 (do siebie samego), aby dostarczyc go do 10.10.56.52. Do znalezienia adresu karty sieciowej hosta koncowego posluzyl protokol ARP i pakiet zostanie dostarczony.

Podsumowujac proces: zrodlowy host na podstawie swojego adresu IP i maski podsieci oraz docelowego adresu IP ustala, czy adresat jest lokalny. Jesli tak, pakiet zostaje przeslany bezposrednio do hosta pod jego adres sprzetowy (okreslony za pomoca ARP). Jesli adresat nie jest lokalny, wówczas nadawca wyszukuje w swojej tablicy tras trase do niego. Jesli trasa zostanie znaleziona, pakiet bedzie wyslany do bramy skonfigurowanej dla tej trasy, na podstawie jej adresu sprzetowego (ponownie ustalonego przez ARP).

Brama odbiera pakiet i w ten sam sposob porownuje adres docelowy ze swoim adresem IP i maska podsieci dla interfejsu, na którym pakiet został odebrany. Jesli pakiet jest lokalny (co swiadczy o wystapieniu bledu gdzies po drodze, poniewaz host juz dokonał tego samego porównania), pakiet zostaje wyslany do hosta docelowego z wykorzystaniem jego adresu sprzetowego. Jesli porównanie wykaze, iz adresat jest zdalny (powinien byc), do ustalenia kolejnego hopu zostaje wykorzystana tablica tras bramy. Jesli kolejny hop jest lokalny dla dowolnego z interfejsow bramy, pakiet zostaje przeslany do kolejki dla tego interfejsu i wyslany bezposrednio do celu za pomoca adresu sprzetowego. Jesli okaze sie, ze trasa nie jest lokalna, brama przesyła pakiet dalej. W przypadku braku trasy ruter zwraca do nadawcy pakietu komunikat „uplynal limit czasu zadania”, „host docelowy nie został znaleziony” lub „host docelowy jest nieosiagalny” za pomoca protokolu ICMP (*Internet Control Message Protocol*).

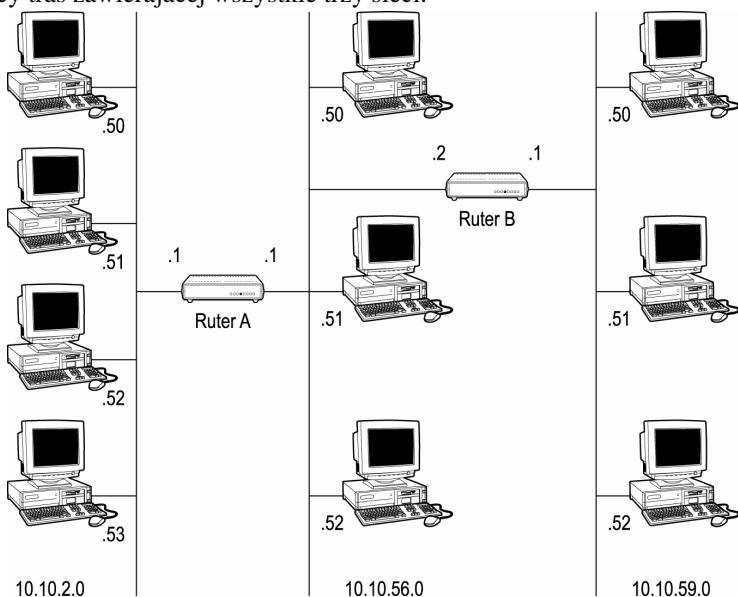
Budowanie tablicy tras

Czytelnik może zastanawiać się, skąd bierze się tabela tras. Czy jest ładowana z pliku? Czy jest wyliczana? Czy użytkownik musi coś zrobić, aby utworzyć tablice tras? Odpowiedz na wszystkie trzy pytania jest twierdząca.

Lecz zacznijmy po kolei. W przykładzie używanym w tym podręczniku trasowanie funkcjonowało, ponieważ ruter był podłączony fizycznie do każdej sieci i mógł dzięki temu budować swoja tablice tras na podstawie tych sieci. Na rysunku 19.2 przedstawione zostały dwa routery i trzy sieci, zas zaden z routera nie jest bezpośrednio podłączony do wszystkich trzech sieci. Oznacza to, że ani jeden, ani drugi ruter nie może zbudować tablicy tras zawierającej wszystkie trzy sieci.

Rysunek 19.2.

Przykład rozbudowy prostej sieci



Rysunek 19.2 przedstawia dwa routery, A i B. Ruter A „wie” o sieciach 10.10.2.0 i 10.10.56.0, ponieważ posiada fizyczne połączenie z każdą z nich. Podobnie, ruter B „wie” o sieciach 10.10.56.0 i 10.10.59.0.

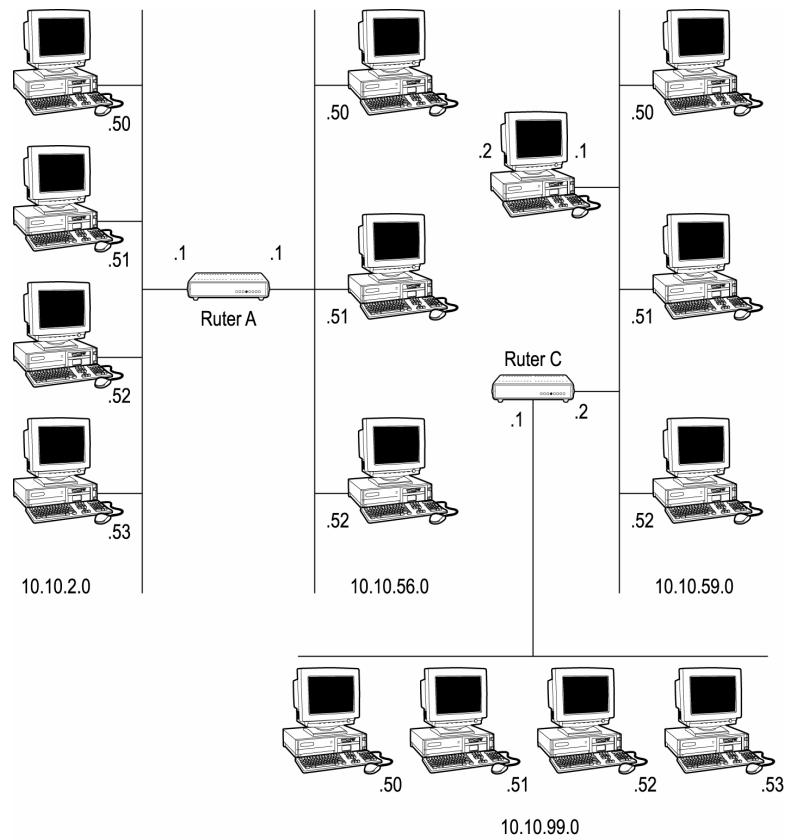
Gdy host 10.10.2.50 próbuje skomunikować się z hostem 10.10.59.51, mamy problem. System sprawdza kombinacje adresów IP i maski podsieci i ustala, że adresat jest zdalny. Dzięki wpisowi bramy domyślnej (0.0.0.0) w tablicy tras, pakiet zostanie wysłany do skonfigurowanego routera (A). Ten również sprawdza IP i maskę podsieci na interfejsie, na którym odebrał pakiet i stwierdza, że ma on zostać dostarczony do hosta zdalonego względem tego interfejsu. Następnie sprawdza swoja tablice tras, a ponieważ zawiera ona jedynie informacje zebrane z lokalnych interfejsów, nie znajdzie trasy. Ruter A nie „wie” nic o sieci 10.10.59.0, ponieważ nie ma z nia fizycznego połączenia. Na skutek tego zwraca komunikat „host docelowy jest nieosiągalny”.

Ruter A potrzebuje jakiegoś metody przekazywania pakietu do routera B. Najprostszym sposobem jest skonfigurowanie routera A tak, by przesyłał wszelkie pakiety o nieznanym sobie adresie do routera B — inaczej mówiąc, użycie routera B jako bramy do-

myslnej dla ruteru A. Wówczas ruter A mógłby uzyc trasy domyslnej, aby przeslac pakiet do ruteru B (poniewaz nie zna innej trasy). Ruter B „zna” siec 10.10.59.0, poniewaz jest z nia fizycznie polaczony. W wyniku tego ruter B dostarczy pakiet.

Pozostaje jednak pewien problem. Zalóżmy, ze host 10.10.2.50 wysyla zadanie echo (ping). Pakiet dociera do 10.10.59.1 i host ten zamierza teraz odeslac odpowiedz echo z powrotem do 10.10.2.50. Host dokonuje porównania, odkrywa ze adresat jest zdalny, znajduje pierwsza brame domyslna i wysyla pakiet do ruteru B. Ten kontroluje pakiet na lokalnym interfejsie i ustala, ze jest zdalny; zgodnie z tym sprawdza tablice tras i zwraca do 10.10.59.1 komunikat „host docelowy jest nieosiagalny”. Host docelowy nie moze zostac osiągniety, poniewaz ruter B nie „wie” o sieci 10.10.2.0 wiecej, niz ruter A wiedzial o sieci 10.10.59.0. Wobec tego najlepszym sposobem, by umozliwic ruterowi B przekazanie pakietu do ruteru A, jest skonfigurowanie A jako Bramy domyslnej w ruterze B. Rozwiazanie to jest late do wykonania, jednakze nie bedzie dzialac, jesli dodamy jeszcze jedna siec. Rysunek 19.3 pokazuje potencjalne komplikacje.

Rysunek 19.3.
Dodajemy kolejna siec



Statyczny wybór trasy

Wraz ze wzrostem rozmiarów sieci metoda Bramy domyslnej przestaje sie sprawdzac. Poniewaz tylko jeden wpis Bramy domyslnej ma znaczenie, mozemy wprowadzic kilka wpisów — lecz tylko pierwszy, który zostanie znaleziony przez system operacyjny, zosta-

nie uzyty, o ile nie jest nieczynny. W tym przypadku mozemy poradzic sobie z siecia za pomoca rutera z wieloma interfejsami (na przyklad czterema), aby wszystkie sieci byly dla niego lokalne; staje sie to jednak trudne, gdy musimy polaczyc 50 lub 60 podsieci.

Musimy „powiedziec” ruterom o istnieniu innych sieci, z którymi nie sa bezposrednio polaczone. Mozemy to uczynic, dodajac statyczne trasy do kazdego routera, aby rozpoznawal, gdzie wyslac pakiety przeznaczone dla nieznanych sieci. Jak widac na rysunku 19.3, ruter A „zna” sieci 10.10.2.0 i 10.10.56.0. Mozemy dodac informacje o 10.10.59.0 i 10.10.99.0, mówiac ruterowi A, aby wysylal adresowane do nich pakiety do routera B. Zasadniczo musimy dodac nastepujace wiersze do tablicy tras:

Adres sieciowy Metryka	Maska sieci	Adres bramy	Interfejs
10.10.59.0 2	255.255.255.0	10.10.56.2	10.10.56.1
10.10.99.1 2	255.255.255.0	10.10.56.2	10.10.56.1

Jesli dodamy te wiersze do tabeli tras w ruterze A i odpowiednie trasy w pozostalych dwóch ruterach, informacje beda mogly przeplywac przez cala siec. W malych organizacjach jest to metoda preferowana, poniewaz nie angazuje dodatkowego ruchu w sieci na wzór dynamicznego trasowania. Jednym problemem jest koniecznosc przeróbek wszystkich statycznych tras, jesli konfiguracja sieci ulegnie zmianie.

Dodanie trasy jest w wiekszosci systemów operacyjnych prosta czynoscia. Praktycznie wszystkie systemy operacyjne uzywaja do tego celu polecenia `route`. Podstawowe parametry sterujace tego polecenia sa nastepujace:

- ♦ `print` — wyswietla tablice tras,
- ♦ `add` — tworzy statyczny wpis w tablicy tras,
- ♦ `delete` — usuwa wpis z tablicy tras,
- ♦ `modify` — zmienia istniejaca trase,
- ♦ `flush` — usuwa wszystkie trasy z tablicy i przeladowuje z pliku lub rejestrzu.

Aby dodac trasy potrzebne w ruterze A do systemu Windows, nalezy wpisac:

```
route add 10.10.59.0 mask 255.255.255.0 10.10.56.2
route add 10.10.99.0 mask 255.255.255.0 10.10.56.2
```

W systemie Linux polecenia te beda wygladac nastepujaco:

```
route -A inet add -net 10.10.59.0 netmask 255.255.255.0 gw 10.10.56.2
route -A inet add -net 10.10.99.0 netmask 255.255.255.0 gw 10.10.56.2
```

Te polecenia dodaja wymagane trasy w komputerze dla uzywanej wlasnie konfiguracji. Jednakże jesli system zawiesi sie i bedzie wymagal przeladowania, trasy zostana utracone. Warto wiecek zapisac trasy w pliku (w srodowisku uniksowym) lub w Rejestrze (w systemach Windows).

Wraz ze wzrostem rozmiarów sieci rośnie tez liczba routerów. Predzej czy pózniej dojdziemy wiecek do punktu, w którym reczne aktualizacje routerów nie beda mozliwe i trzeba bedzie znalezc jakas metode automatycznej aktualizacji tablic tras. To zagadnienie

zostanie omówione w jednym z kolejnych podpunktów, dotyczącym dynamicznego wyboru tras.

Tworzenie struktury trasowania

Omówiliśmy już podstawy trasowania, więc pora skierować dyskusje na fizyczny rozkład sieci. Typ używanej sieci fizycznej i wymagania dotyczące wydajności, omówione w rozdziale 18., również wchodzą w grę przy tworzeniu sieci. W istocie czynniki te są scisłe ze sobą powiązane.

Maski podsieci, które omówiliśmy w rozdziale 18., będą używane jako podstawowe maski dla klientów w podsieciach. Teraz musimy połączyć ze sobą wszystkie klienty za pomocą wybranego rodzaju sprzętu sieciowego, aby mogły komunikować się ze sobą. Oznacza to planowanie użycia koncentratorów lub przełączników do utworzenia topologii magistrali lub jednostek dostępu do stacji wieloterminalowych (MAU), aby utworzyć pierścień.

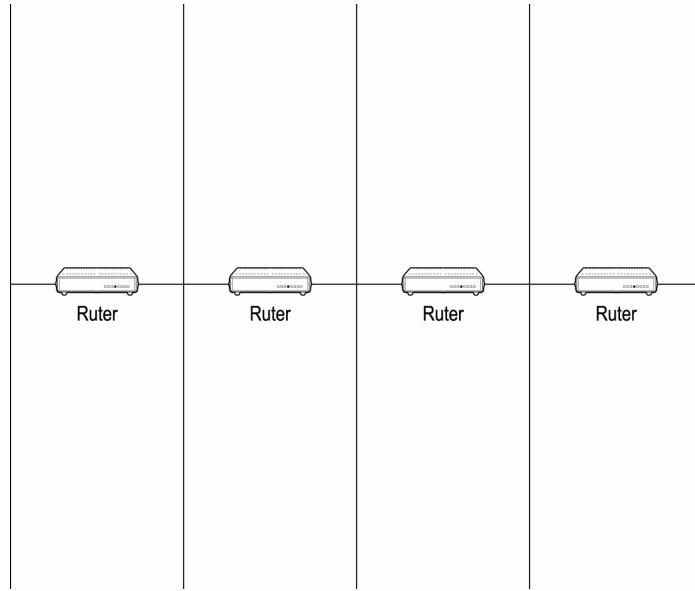
Ponieważ większość czytelników zastosuje Ethernet, który jest topologią magistrali, skoncentrujemy się na tej właśnie topologii. Użytkownicy sieci Token Ring mogą, jak pamiętamy, do segmentu przylączyć więcej hostów, niż jest w stanie obsłużyć Ethernet. W rozdziale 18. przedstawiliśmy podstawowy wzór do wyliczenia, ile hostów może zostać połączonych do podsieci.

Laczenie podsieci

Jak połączyć ze sobą podsieci, aby trasowanie nadal funkcjonowało? Narzucająca się metoda (użyta w poprzednich przykładach) jest doczepianie kolejnych podsieci na koniec sieci logicznej (podsiec — ruter — podsiec), aż uzyskamy liczbę podsieci wystarczającą dla liczby posiadanych użytkowników (patrz rysunek 19.4).

Rysunek 19.4.

*Szeregowe
laczenie podsieci*

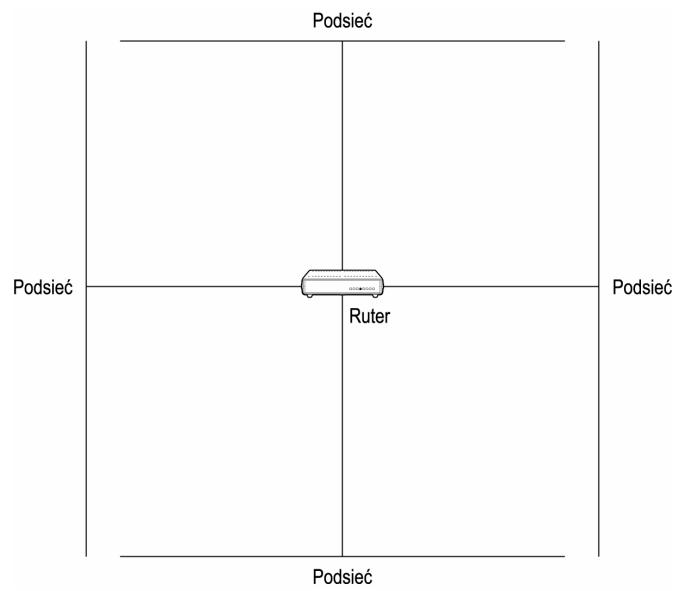


Ta metoda jest prosta i łatwa do zaimplementowania. Gdy jednak użytkownicy muszą komunikować się z innymi, oddalonymi o kilka hopów, routery mogą zostać mocno obciążone. Jeśli którykolwiek z nich zawiedzie, może to mieć wpływ na całą sieć, ponieważ struktura nie jest nadmiarowa, a serwery nie są skoncentrowane.

Kolejnym oczywistym rozwiązaniem jest rozmieszczenie wszystkich sieci dookoła centralnego routera. Równie łatwo wyobrazić sobie taką sieć, a wszystkie hosty powinny być w stanie łatwo komunikować się ze sobą (patrz rysunek 19.5).

Rysunek 19.5.

*Podsieci rozmieszczone
dookoła
centralnego
rutera*

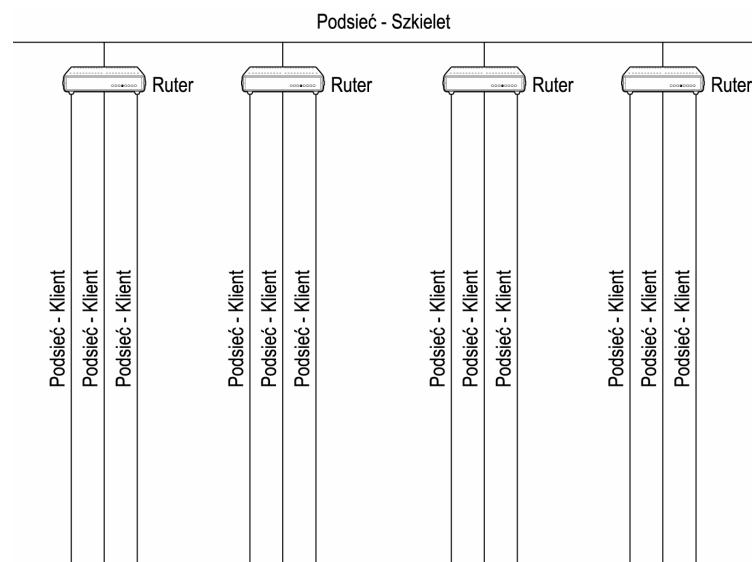


Centralny ruter dobrze sprawdza sie w malych biurach. W takiej strukturze mozemy tez latwo zapewnic nadmiarowosc, dodajac w centrum drugi ruter. Jedynym problemem jest duza objetosc danych, przepływajacych przez jeden ruter lub pare ruterów. A jesli nie zastosujemy zapasowego rutera, siec bedzie posiadac pojedynczy punkt awarii.

Struktura z centralnym ruterem to konstrukcja pozwalajaca rozbudowywac siec. Jesli wyznaczmy jedna z podsieci z rysunku 19.5 do roli „szkieletu”, otrzymamy ruter kazacy trzy podsieci z siecią szkieletową. Rozbudowujac siec, przylaczmy kolejny ruter do sieci szkieletowej, a nastepnie dolaczmy podsieci klienckie do tego rutera (patrz rysunek 19.6).

Rysunek 19.6.

Wykorzystanie ruterów do podlaczenia podsieci klienckich do sieci szkieletowej



Rysunek 19.6 przedstawia dosc typowy projekt, uzywany w wielu sieciach. Wspólne serwery (na przyklad serwer pocztowy) sa zwykle przylaczone do sieci szkieletowej, zas serwery klienckie (serwery plików i drukowania) moga byc przylaczone do podsieci razem z klientami. Awaria pojedynczego rutera nie wpływa na cala siec i jest dosc miejsca na rozwój. Patrzac na rysunek 19.6 przypomnijmy sobie obliczenia z rozdziału 18., gdzie kazda z podsieci mogla pomiescic 30 lub 126 komputerów. W dwunastu podsieciach moze pracowac 360 uzytkowników, a maksymalnie 1512. Dodawanie kolejnych uzytkowników nie jest trudne; wystarczy dolaczyc kolejny ruter i trzy kolejne podsieci lub przylaczyc 4 do 8 podsieci do kazdego rutera.

Taka struktura jest bardzo elastyczna; jednakże objetosc danych, z jaka moze poradzic sobie siec szkieletowa, jest ograniczona (nawet jesli jest zbudowana w technice FDDI). Mozemy zmniejszyc ten ruch, przenoszac wiecej serwerów do podsieci klienckich i zapewniajac, by uzytkownicy udostepniajacy sobie nawzajem dane i komunikujacy sie ze soba byli podlaczeni do jednej podsieci, a przynajmniej do jednego rutera.

Predzej czy pózniej trzeba bedzie jednak przekroczyć limit ruchu sieciowego, z jakim potrafi uporac sie pojedyncza siec szkieletowa. Mozna to zrobic, dzielac szkielet na czesci i konfigurujac trasowanie pomiedzy nimi. Prawdopodobnie nasza siec i tak juz

posiada wiecej szkieletów, poniewaz typowe sieci obejmują wiecej niż jedna lokalizacje, z osobna siecią szkieletową w każdej lokalizacji.

Maski podsieci o zmiennej długości

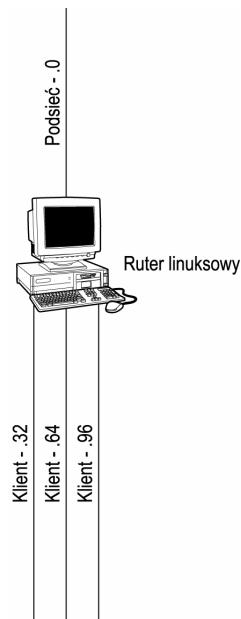
W strukturach posiadających jedną sieć szkieletową problemem jest ruch sieciowy. Jeśli wszystkie serwery podłączone są do jednego szkieletu lub użytkownicy intensywnie ze sobą współpracują, sieć szkieletowa będzie poradnie obciążona. Maski podsieci o zmiennej długości (VLSM — *Variable Length Subnet Masking*), zdefiniowane w RFC 1817, są rozwiązaniem pozwalającym tworzyć kilka poziomów sieci szkieletowej, a co za tym idzie, kilka różnych miejsc, w których dane mogą przechodzić z jednej sieci do drugiej. Implementacja VLSM jest jednakże trudna, ponieważ w różnych częściach sieci mamy do czynienia z różnymi maskami podsieci.

Na potrzeby wyjaśnienia tego zagadnienia rozważmy maskę podsieci 255.255.255.224 jako opcjonalną maskę podsieci dla naszych klientów. Oznacza to, że chcemy ograniczyć każdą podsieć do 30 adresów IP, czyli 29 komputerów i interfejsu routera. Możemy więc do budowania sieci wykorzystać 24-portowe przełączniki. Założymy dodatkowo, że stosujemy system Linux na potrzeby niskopoziomowych routerów, i że każdy z nich będzie obsługiwał cztery karty sieciowe. Wobec tego kolejne podsieci będą miały adresy .0, .32, .64 i .96. Interfejsy amerykańskie routerów będą przypuszczalnie pierwsze adresy IP z każdego segmentu — .1, .33, .65 i .97. Aby połączyć się z dużą siecią, potrzebne będzie jedno połączenie, więc wykorzystajmy podsieć .0. Rysunek 19.7 pokazuje, jak taki system może wyglądać.

Mamy teraz konfigurację, w której cały ruch do sieci .0, .32, .64 lub .96 będzie musiał przejść do .1. Dodajmy pozostałe oktety, aby nasze adresy wyglądały bardziej normalnie. Cały ruch do podsieci 10.10.10.0, 10.10.10.32, 10.10.10.54 i 10.10.10.96 musi być kierowany na interfejs 10.10.10.1. Router ten następnie przesyła do podsieci przeznaczone dla nich dane. Oznacza to jednak, że wszystkie dane dla adresów IP od 10.10.10.1 do

Rysunek 19.7.

*Ruter laczacy
cztery podsieci*



10.10.10.127 musza przejsc przez ten ruter. Wprawdzie dla trzech klienckich podsieci maska podsieci musi byc 255.255.255.22, lecz dla interfejsu laczacego je z wieksza siecia mozemy uzyc w masce sieci 255.255.255.128 lub mniej bitow.



W tym rozdziale uzywamy masek podsieci, lecz Czytelnik moze również spotkac sie z notacją /liczba_bitów, która jest łatwiejsza do czytania, gdy sie juz do niej przyzwyczaimy. Na przykład, adres 10.10.10.0 z maska podsieci 255.255.254.0 bedzie zapisany jako 10.10.10.0/23.

Wykorzystanie takiego schematu adresowania oznacza, ze inny ruter moze posiadac adres 10.10.10.129 i obslugiwac ruch dla trzech innych podsieci. W istocie, na rysunku 19.8 widac, ze jesli uzyjemy maski podsieci 255.255.254.0, bedziemy mogli polaczyc ze soba jeszcze wiecej ruterów.

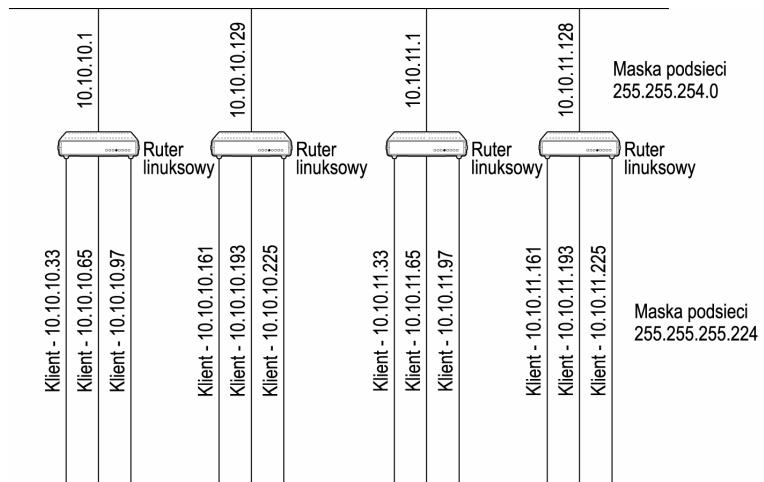
Zbudowalismy wiec strukture sieci szkieletowej z własnymi maskami podsieci. Duze osiągniecie, prawda? No coż, jesli przyjrzymy sie dokladnie adresom na górze rysunku 19.8, zauwazymy, ze *kazda* z tych dwunastu sieci moze byc opisana jako siec 10.10.10.0 z maska 255.255.254.0. Oznacza to, ze mozemy potraktowac cala strukture z rysunku 19.8 jako pojedynczy wezel i powielic kilka razy, tworząc wieksza siec. Rysunek 19.9 przedstawia nastepny poziom hierarchii.

Stosowanie masek o zmiennej dlugosci wymaga oczywiscie wiecej pracy podczas planowania, zaczynajac od decyzji o maksymalnej liczbie hostow. Jednakze skorzystanie z VLSM niesie ze soba kilka korzysci:

- ♦ Mozemy umiescic serwery na dowolnym poziomie hierarchii, dzieki czemu beda blizej uzytkownikow.
- ♦ Mozemy przydzielic oddzialom lub lokalizacjom bloki adresow o różnych wielkosciach, zaleznie od potrzeb.

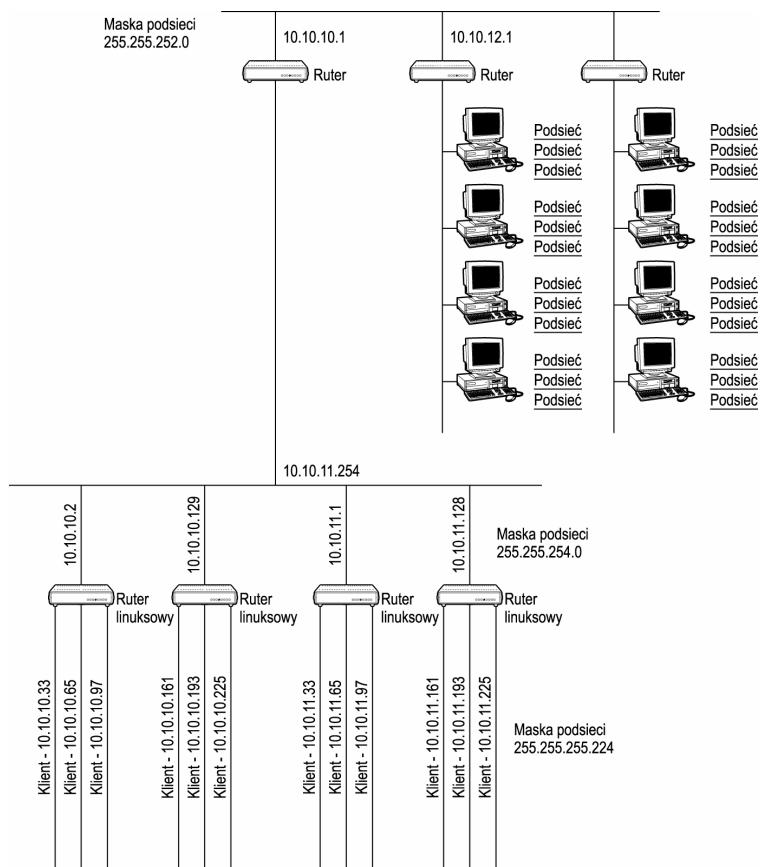
Rysunek 19.8.

Grupa czterech ruterów
przyłączająca 12
podsieci



Rysunek 19.9.

Trzy poziomy hierarchii
w schemacie
VLSM



♦ Zredukujemy ruch sieciowy dochodzący az do sieci szkieletowej.



Czytelnicy pracujacy w srodowisku Windows powinni pamietac, ze Windows NT 4.0 i starsze systemy operacyjne nie moga uzywac VLSM. Jest to związane ze sposobem, w jaki te systemy operacyjne porządkują wpisy w tablicach tras. Aby móc zastosować VLSM, wpisy o długich maskach podsieci muszą być sprawdzane w pierwszej kolejności.

Podlaczanie odleglych biur

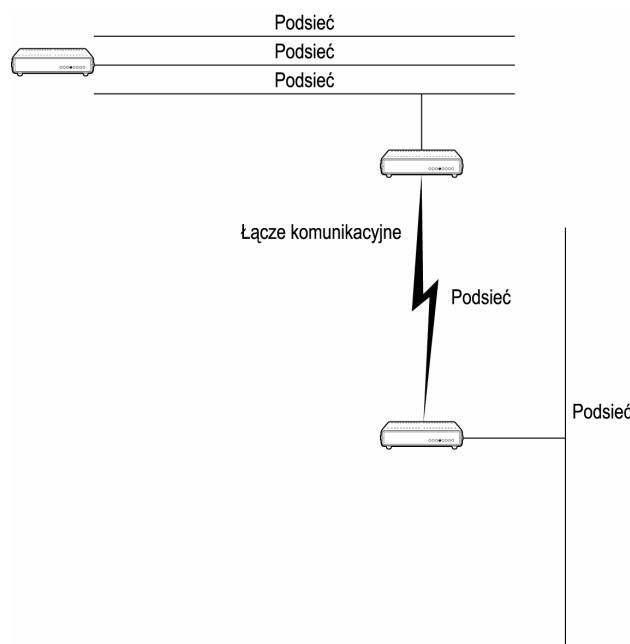
Sposób podlaczenia odległych jednostek będzie zależał od ogólnego schematu, według którego zbudujemy swoją sieć. Należy pamiętać o kilku podstawowych zasadach:

- ◆ Kazde biuro powinno posiadać ciągły blok adresów. Ułatwi to trasowanie pomiędzy biurami.
- ◆ Jesli połaczenie używa interfejsu wyboru trasy na zadanie (*dial-on-demand*), trzeba będzie dodać dla niego trasy statyczne.
- ◆ Pośród dwóch koncentrycznych komunikacyjnych trzeba uwzględnić podsieci. W przypadku połączenia dwupunktowego może to być podsieć fałszywa.
- ◆ Powinnismy posiadać połączenie zapasowe na wypadek awarii głównego połączenia. Dla typowego ruchu połączenie zapasowe może być typu L2TP (lub PPTP w srodowisku Microsoftu).

Ogólnie mówiąc, jesli używamy centralnego routera w małej sieci, router wychodzący powinien być połączony do podsieci zawierającej zasoby potrzebne zdalnym użytkownikom, jak na rysunku 19.10.

Rysunek 19.10.

Ruter łączący
dwa biura
w prostej sieci

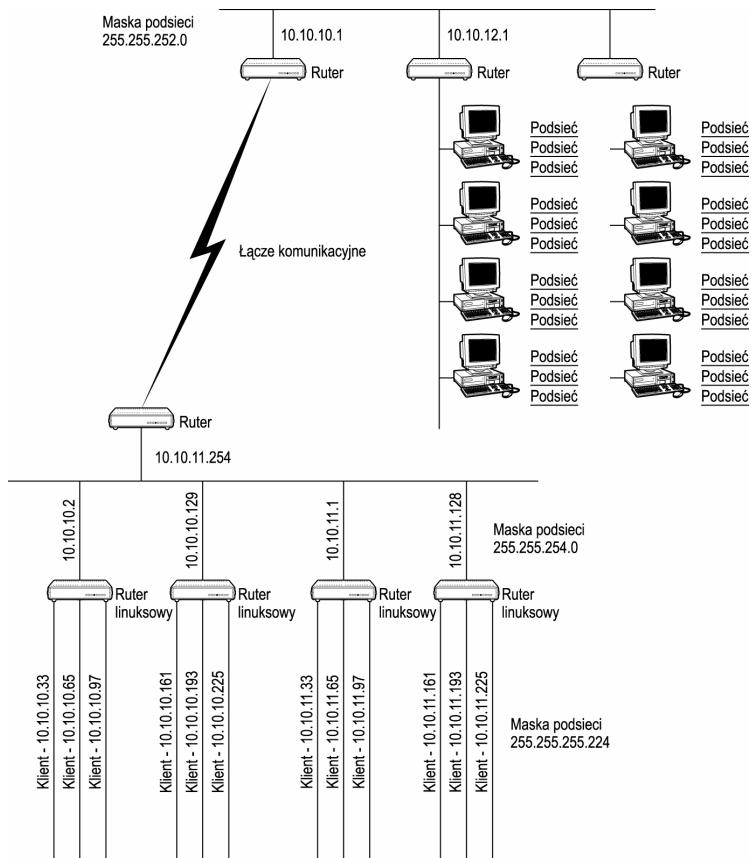


Jesli sieć jest zbudowana na podstawie sieci szkieletowych, w każdym biurze należy utworzyć szkielet i połączyć routery tworzące łącze ze szkieletami w każdym biurze.

W implementacji VLSM mozemy oderwac czesc przestrzeni adresowej potrzebna dla zdalnego biura i dodac ruter na odpowiednim poziomie. Rysunek 19.11 przedstawia przyklad takiej architektury.

Rysunek 19.11.

Lacze komunikacyjne w implementacji VLSM



Dynamiczny wybór tras

Jak pokazalismy wczesniej, rutery uzywaja tablic tras do ustalania, gdzie przekazac kazdy pakiet wymagajacy przeslania dalej. Oznacza to, ze tablica tras w kazdym ruterze musi zawierac wszystkie informacje niezbedne, by znalezc nastepny hop do dowolnego systemu w sieci. W przypadku malych sieci mozemy to osiągnac, wprowadzajac recznie trasy statyczne w ruterach. Dla sieci duzych — lub zmieniajacych sie często — reczne konfigurowanie niekoniecznie jest dobrym rozwiazaniem.

W sieciach duzych lub dynamicznych zwykle uzywac bedziemy protokolu dynamicznego wyboru tras, w którym ruter wymienia swoje informacje o trasach z innymi ruterami w sieci. Dostepnych jest kilka protokołów wyboru tras, przyjrzymy sie czterem z nich:

- ♦ *IRD (Router Discovery z protokołu ICMP)*
- ♦ *RIP (Routing Information Protocol)*

♦ *IGRP (Internet Gateway Routing Protocol)*

♦ *OSPF (Open Shortest Path First)*

Jednym z problemów z duzymi sieciami TCP/IP jest to, iz różne grupy w obrebie organizacji (lub nawet w różnych organizacjach) zarzadzaja różnymi obszarami sieci. Dla prostoty sieci TCP/IP sa zwykle dzielone na systemy autonomiczne (AS — *Autonomous System*). Kazdy system autonomiczny moze stosowac do zarzadzania trasami dla bram wewnętrznych osobny protokół trasowania z rodzaju IGP (*Interior Gateway Protocol* — wewnętrzny protokół bramowy). IGP odpowiada za znajdowanie wszystkich bram przez pozostałe w obrebie systemu autonomicznego, jednakże IGP nie pozwala na wymiane informacji o trasach pomiędzy różnymi systemami autonomicznymi. Do IGP naleza protokoly RIP i IGRP, które sluzą do wzajemnego udostepniania informacji o trasach wewnatrz systemu autonomicznego.

Gdy chcemy wymieniac informacje pomiędzy różnymi systemami autonomicznymi, potrzebny jest zewnętrzny protokół bramowy (EGP — *Exterior Gateway Protocol*). Do pewnego stopnia moze posłuzyc do tego IGRP, lecz takie protokoly, jak OSPF sa lepsze, poniewaz zostały zaprojektowane specjalnie w tym celu.

ICMP Router Discovery

Router Discovery z protokolu ICMP tak naprawde nie jest protokołem wyboru trasy. Jest narzedziem, za pomoca którego host znajduje lokalna brama domyslna, jesli nie jest dla niego skonfigurowana recznie. IRD wykorzystuje dwa polecenia protokolu ICMP (*Router Discovery* — wykrycie ruteru i *Router Advertisement* — ogłoszenie ruteru), pozwalajac klientowi wykryc ruter w swojej podsieci.

Ogłoszanie ruteru

Rutery uzywajace IRD okresowo ogłasza swoja obecnosć w sieci: albo za pomoca adresu grupowego 224.0.0.1, albo za pomoca rozgłoszeń sieciowych pod adresem 255.255.255.255. Gdy przychodzi na to czas (zwykle co 7 – 10 minut), ruter ogłasza na wszystkich lokalnych interfejsach adresy IP tych interfejsów. Podaje dodatkowo wartosc preferencji, aby w przypadku obecnosci kilku ruterów klient wybrał ten o najwyższej wartosci (liczbie) preferencji. Liczba ta jest stosowana, by dac administratorom kontrole nad tym, który ruter bedzie standardowo uzywany przez klienty.

Ogłoszenia zawieraja również czas zycia (TTL), który okresla, jak dlugo klient bedzie miał prawo korzystac z ruteru. Czas ten powinien byc dluzszy od okresów pomiedzy ogłoszeniami — domyslnie wynosi 30 minut.

Wykrycie ruteru

Wykrycie ruteru nastepuje zwykle, gdy host usiluje polaczyc sie z systemem spoza swojej podsieci niedlugo po swoim uruchomieniu (do 7 – 10 minut). Jesli host otrzymał już ogłoszenie ruteru, nie musi dokonywac wykrycia. Jesli jednak ruter w podsieci ulegnie awarii lub klient musi natychmiast polaczyc sie ze zdalna siecia, moze wyslac komunikat Route Discovery.

Ruter musi być tak skonfigurowany, aby pozwalał na wykrycia oraz aby używał 255.255.255 jako adresu docelowego lub 224.0.0.2 jako adresu grupowego. Rutery obsługujące IRD dolaczają się do grupy pod tym adresem i wysyłają komunikat Router Announcement (ogłoszenie ruteru), gdy odbiorą zadanie Router Discovery. Nie wszystkie systemy operacyjne obsługują IRD (nawet w roli klientów). Na przykład, starsze klienty Microsoftu były niezdolne do wykrywania ruterów.

Jesli nasza sieć ulega częstym zmianom i nie planujemy stosowania DHCP, IRD może być przydatnym protokołem. W większości przypadków jednak nie powinien być implementowany w środowisku przedsiębiorstwa.

Protokół RIP

Routing Information Protocol (RIP), zdefiniowany w standardzie internetowym STD 34, dobrze spełnia swoje zadania jako instrument, który daje ruterom możliwość dynamicznej wymiany informacji o trasach, w miarę zmian warunków w sieci. Istnieją dwie wersje RIP: 1. i 2. Wersja pierwsza jest implementowana już w bardzo niewielu miejscach, lecz warto się jej przyjrzeć, gdyż stanowi podstawę wersji 2.

Wersja 1.

Jak pamiętamy z podrozdziału dotyczącego tablic tras, na wpis składa się kilka elementów:

- ♦ docelowa sieć lub host,
- ♦ maska sieci,
- ♦ interfejs,
- ♦ brama,
- ♦ metryka.

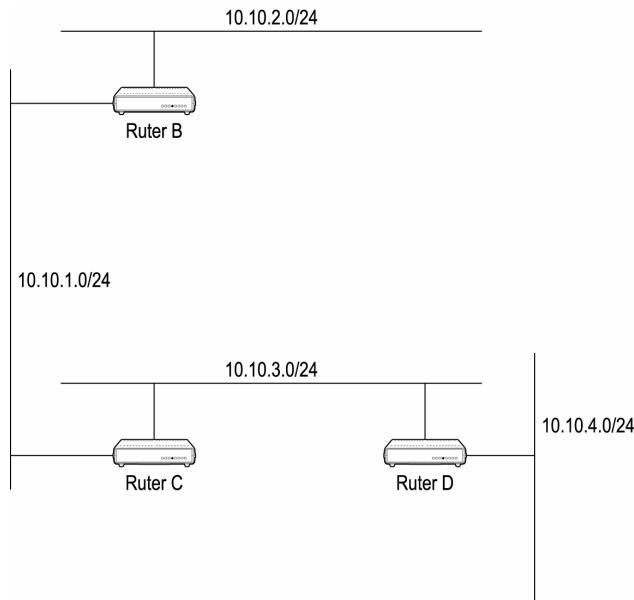
Wielkość tych elementów powinna już być znajoma. Jesli wynikiem jest docelowa sieć lub host, dane są wysyłane do interfejsu w celu dostarczenia do bramy. Należy zdać sobie sprawę, iż docelowy adres IP jest zestawiany z każdą maską sieci. Różne protokoły wyboru trasy używają metryki do zapisywania różnych wartości. W przypadku protokołu RIP metryka oznacza liczbę ruterów, przez które musi przejść pakiet.

Powód stosowania metryki jest stały — służy ona do ustalenia najlepszej trasy do zdalnej sieci, a więc do zdalnego hosta. Gdy mamy do wyboru sciezki zawierające dziesięć ruterów i sciezki złożone zaledwie czterech, wybierzemy te drugie. Rysunek 19.12 przedstawia prosty przykład sieci.

Rysunek 19.12 przedstawia cztery wewnętrzne podsieci z 24-bitowymi maskami podsieci. W trasowanie zaangażowane są trzy routery. Klienci w sieci 10.10.4.0 mogą mieć spore kłopoty ze skontaktowaniem się z klientami z sieci 10.10.2.0. Zakładając, że routery obsługują protokół RIP, i że zaczynamy od routera D, proces dynamicznego udostępniania tras za pomocą RIP przebiega następująco:

Rysunek 19.12.

Udostepnianie informacji o trasach za pomoca protokolu RIP w wersji 1.



1. Ruter D „zna” sieci 10.10.4.0 i 10.10.3.0, wiec w odpowiednich odstepach czasu (domyslnie 30 sekund) rozglasza swoja tablice tras.
2. Ruter C odbiera rozgloszenie i sprawdza zawarte w nim trasy.
3. Ruter C zwiększa wszystkie metryki w rozgloszeniu o wartosc metryki dla interfejsu, na którym odebral rozgloszenie (domyslnie o 1).
4. Ruter C sprawdza trasy. Znajduje trasę do sieci 10.10.4.0 o metryce równej dwa, której nie rozpoznał, wiec dodaje trasę do tablicy. Oprócz tego znajduje trasę do 10.10.3.0, lecz te posiada juz w swojej tablicy tras. Ponieważ posiada dwie trasy do tej samej sieci, porównuje ich metryki. W przypadku tras do 10.10.3.0 ruter ustala, że posiadana juz przez niego trasa ma metryke o jeden nizsza od odebranej w rozgloszeniu. Ponieważ metryki tras z rozgloszena sa zawsze zwiększone o jeden przed porównaniem tras, trasy do sieci lokalnej zawsze wygrywają z ogłaszanymi przez sasiadujacy ruter.
5. Ruter C w odpowiedniej porze rozglasza trasy które „zna”, lacznie z tymi, które wlasnie „poznał”.
6. Ruter B otrzymuje rozgloszenie i aktualizuje swoje informacje, dodajac trasy dla 10.10.4.0 i 10.10.3.0. Trasa dla 10.10.1.0 nie zostaje dodana, poniewaz istnieje juz trasa lokalna.

Na tym etapie wszystkie rutery „wiedza” o sieci 10.10.4.0; ruter B również wysle rozgloszenie, aktualizujac dane rutera C. Ten z kolei przez rozgloszenie zaktualizuje dane rutera B. Teraz wszystkie rutery znaja wszystkie trasy, a siec jest w stanie zbieznosci. Niestety RIP nie posiada tych informacji, wiec dalej powtarza rozgloszenia co 30 sekund. Transmisje te moga zajac troche pasma, zwlaszcza jesli siec zawiera wiecej routerów.

Z protokołem RIP są, poza ciągim rozmieszczaniem, jeszcze inne problemy. Bardzo poważnym problemem jest wielkość, jaką mogą przybrać pakiety rozmieszczonych. Założymy, na przykład, że mamy użyć protokołu RIP w Internecie. Jeśli ruter w San Diego w Kalifornii rozmieści pakiet RIP, a informacje będą przesyłane i uzupełniane od routera do routera, jak to rozmieszczenie będzie wyglądać w Glasgow w Szkocji? Oczywiście skumulowane po drodze informacje, które dotarły na drugi koniec świata, będą gigantyczne. Aby zapobiec niszczeniu sieci przez RIP, projektanci wbudowali mechanizm zabezpieczający. żaden ruter nie może mieć metryki wyższej niż 15, co ogranicza efektywne rozmiary sieci, w której można zastosować protokół RIP.

Reakcja na zmiany warunków w sieci

Jednym z głównych powodów zastosowania protokołu dynamicznego jest fakt, iż warunki w sieci zmieniają się od czasu do czasu. Ruter może się zawiesić lub uszkodzić, kable sieciowe mogą ktoś przeciąć, a łączność komunikacyjna może się zerwać. Wobec tego, każdy użyty protokół musi uporać się z takimi zmianami.

RIP rozmieszcza swoje tablice tras co 30 sekund, co pozwala na propagację nowych tras i usuwanie starych. W pewnych warunkach RIP może mieć jednak problemy z metrykami odliczającymi do nieskończoności.

Odliczanie do nieskończoności

Gdy trasa do sieci w RIP przestaje być dostępna, ruter pozostaje najbliżej punktu uszkodzenia przestaje otrzymywać aktualizacje od routera sąsiadującego. Po upływie 180 sekund bez aktualizacji trasa zostaje uznana za nieosiągalną i jej metryka otrzymuje wartość 16 (jak pamiętamy, najwyższa dopuszczalna metryka wynosi 15, więc oznacza to brak dostępu do sieci). Gdy routery udostępniają tablice tras sąsiadom, metryki przyrostają.

Niektórzy z Czytelników mogą zauważyć w tym problem. Na przykład, na rysunku 19.13 jedna z sieci jest nieczynna. Ruter C posiada do sieci 10.10.4.0 trasę o metryce 2 przez ruter D; ruter B posiada do sieci 10.10.4.0 trasę o metryce 3 przez ruter C. Metryka trasy do sieci 10.10.4.0 zostanie ustawiona na 16, jeśli ruter C nie odbierze nic od routera D przez okres trzech rozmieszczeń.

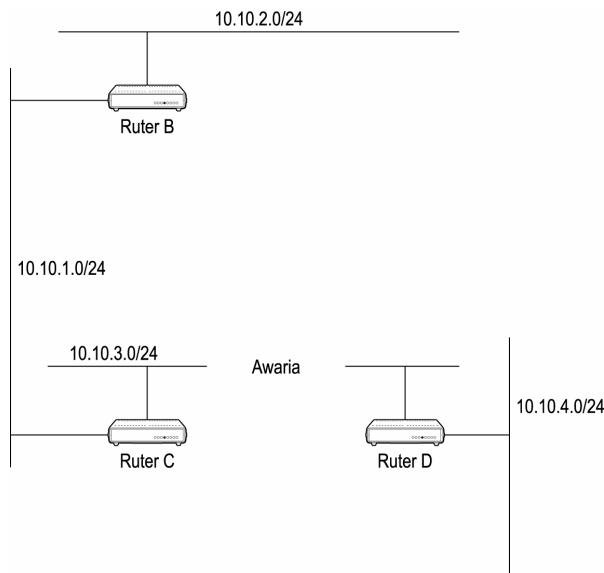
Jeśli ruter C jako pierwszy ogłoszi swoją trasę, problemu nie będzie. „Widzi” on trasę do 10.10.4.0 z metryką 3, która jest lepsza od 16, więc ją zaakceptuje. Teraz ruter C rozmieści swoje informacje o trasach, łącznie z trasą do 10.10.4.0 o metryce 4. Ruter B ustali, że jego trasa do 10.10.4.0 przechodzi przez ruter C i zaktualizuje swoja tablice, więc metryka będzie wynosić 5. Proces ten będzie trwał przez jakiś czas, dopóki metryka w obu routera nie osiągnie 15. Zjawisko to nazywane jest problemem odliczania do nieskończoności i wyjaśnia, dlaczego najwyższa metryka wynosi 15.

Metody podziału horyzontu i zatrucia zwrotu

Odliczanie do nieskończoności oczywiście marnuje czas i przepustowość sieci. Można jednak zastosować dwie metody, które pomagają zmniejszyć ten problem. W pierwszej aktualizacje tras nie są wysyłane do sąsiada, od którego informacje o danych trasach zostały otrzymane. Gdyby w poprzednim przykładzie ruter B nie odesłał trasę z powrotem do routera C, problem nie wystąpiłby. Ta metoda nosi nazwę *podziału horyzontu (split horizons)*.

Rysunek 19.13.

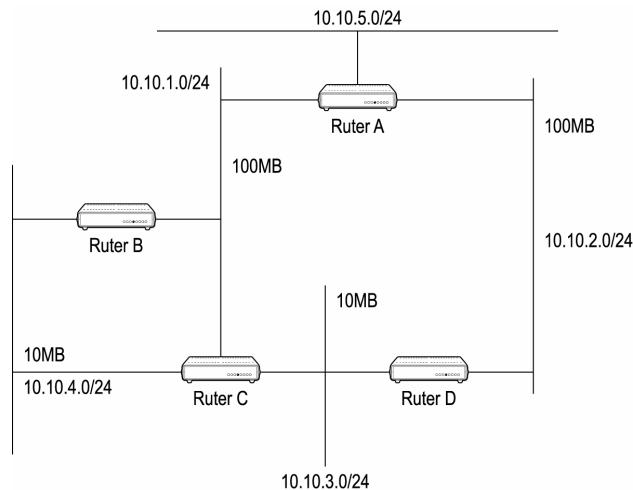
Odlaczanie do nieskonczonosci



Druga metoda, nazywana *zatruciem zwrotem (poisoned reverse)*, jest bardzo podobna do pierwszej, lecz inaczej traktuje aktualizacje. Odsyła je z powrotem do systemu, od którego poznala trasy, lecz z metryka 16. Jak widac, obie metody pozwalaja uniknac zapalenia pomiedzy dwoma systemami; spójrzymy jednak na rysunek 19.14, w którym pojawił sie dodatkowy ruter w sieci.

Rysunek 19.14.

Bardziej złożony problem odliczania



Aktualizacje wyzwalane

W tej sieci w przypadku analogicznej awarii, ruter C wysyla aktualizacje odebrane przez ruter A, lecz nie przez ruter B, z powodu zatoru lub podobnego problemu. Ruter A musi teraz oznaczyc trasę do 10.10.4.0 metryka 16. Ruter B wysyla aktualizacje i ruter A ustala, iz B posiada trasę do 10.10.4.0 z metryka 3. Ruter A wysyla swoje aktualizacje. Poniewaz uzyskal informacje o 10.10.4.0 od rutera B, przekazuje te trase ruterowi C.

Na tym etapie ruter A posiada trase przez ruter B, który posiada trase przez C, który z kolei posiada trase przez A. Ponownie konczymy w petli, zwiększać metryki az do osiągnięcia 15. Zadna z dwóch poprzednich metod nie zapobiegnie problemowi odliczania do nieskończoności.

Aby zatrzymać te pętle, musimy dodać kolejną funkcję do protokołu RIP — *aktualizacje wyzwalane (triggered updates)*. Funkcja ta pozwala ruterowi natychmiast wysłać aktualizację do innych w przypadku zmiany warunków.

Uprzatanie pamięci

Po całej tej dyskusji o przyrostaniu metryki do 16 Czytelnik przypuszczalnie zastanawia się, kiedy te trasy zostaną usunięte. Proces *uprzatania pamięci (garbage collection)* jest w rzeczywistości bardzo prosty. Po 180 sekundach bez aktualizacji trasa otrzymuje metrykę 16 i licznik czasu uprzatania pamięci zostaje ustawiony na 120 sekund; po odliczeniu do 0 trasa zostaje usunięta.

Wersja 2.

W prawdziwej wersji RIP wersji pierwszej był zdany do użytku, lecz sprawiał pewne problemy — na przykład, nie wysyłał razem z trasą maski sieci. RIP w wersji 2. (zdefiniowany w RFC 2453) to aktualizacja wersji 1., zawierająca zmiany w następujących dziedzinach:

- ◆ uwierzytelnianie,
- ◆ znaczniki tras,
- ◆ maski podsieci,
- ◆ adresy ruterów dla następnego hopu,
- ◆ obsługa adresowania grupowego.

Uwierzytelnianie

W wersji 1. nie istnieje żadna forma uwierzytelniania, co oznacza, że każdy ruter w sieci jest w stanie odczytać trasę i poznać strukturę sieci. Doprowadziło to do powstania „cichych” RIP — stacji, które jedynie nasłuchują rozmów RIP. W wersji 2. pierwszy wpis RIP może być oznaczony typem adresu 0xFFFF, co oznacza, że wpis ten jest informacją uwierzytelniającą. Wybór takiego sposobu uwierzytelnienia pozwala na użycie więcej niż jednego typu uwierzytelniania.

Jedyna obecnie zaimplementowana metoda jest uwierzytelnianie otwartym tekstem. Hasło może mieć do 16 znaków i jest przesyłane w postaci nie zaszyfrowanej. Nie wprowadza to jakiegos wyjątkowego poziomu bezpieczeństwa, lecz stanowi krok we właściwym kierunku.

Znaczniki tras

Znacznik trasy (*route tag*) jest nowym atrybutem, który został dodany do drugiej wersji RIP i może zostać ustawiony dla dowolnej trasy w pakiecie RIP. Jego zadaniem jest umożliwienie systemom oznaczenia tras, poznanych przez inne protokoly. Na przykład,

system uzywajacy OSPF moze „dowiedziec” sie o trasie do innej sieci, ktora musi oglosic wewnetrznie w systemie autonomicznym. Od ruterow stosujacych oba protokoly oczekuje sie oznaczania tras w RIP, aby nie byly traktowane jak zwykłe trasy RIP.

Maski podsieci

Dodanie do pakietow RIP maski podsieci pozwala pracowac z różnymi typami schematów podzialu na podsieci, lacznie z VLSM i nadsieciemi. Pewne problemy moga nadal pozostac — na przyklad, jesli pomieszamy rutery uzywajace RIP w wersjach 1. i 2.

Na przyklad, dla VLSM RIP w wersji 1. nie rozpoznaje różnicy pomiedzy 10.10.10.0/23 a 10.10.10.0/27, wobec tego potraktuje obie jako trasy do tej samej sieci. Jesli planujemy wykorzystanie VLSM, RIP w wersji 1. nie powinien byc uzywany.

Ten sam scenariusz wystepuje w przypadku bezklasowego trasowania domen interemetowych (CIDR). W tej technologii mozemy na przyklad polaczyc dwie sieci klasy C — 192.14.2.0 i 192.14.3.0 — w jedna podsieci klasy B 192.14.2.0 z maska 255.255.254.0. Poniewaz RIP w wersji 1. nie wysyla maski podsieci, ten scenariusz rowniez nie zadziala.

Adres rутera dla nastepnego hopu

Wpisy w pakietach RIP wersji 2. powinny zawierac pole nastepnego hopu, czyli kolejnego rутera, do którego pakiet bedzie przeslany. Ten adres ma za zadanie zmniejszyc liczbe nieistotnych hopow w sieciach, w których nie wszystkie rutery obsluguja RIP. Dla trasy do sieci, z ktora ruter jest bezposrednio polaczony, pole zawiera 0.0.0.0.

Adresowanie grupowe

Zostalo dodane adresowanie grupowe (*multicasting*), by zmniejszyc obciążenie systemów w sieci. Redukuje ono obciążenie innych systemów, poniewaz adres 224.0.0.9 bedzie interesujacy jedynie dla systemów, które nasluchuja tej transmisji. Pozostale systemy odrzuca pakiet juz w warstwie fizycznej, a nie w warstwie internetowej.

RIP jest protokolem prostym i skutecznym. Ograniczaja go jedynie rozmiary sieci, jaka jest w stanie obsluzyc. Ograniczenie to czyni z protokołu RIP dobry wybór dla malych sieci. W sieciach duzych powinny byc uzywane inne protokoly, na przyklad *Internet Gateway Routing Protocol* (IGRP).

Protokół IGRP

Internet Gateway Routing Protocol (IGRP), opracowany przez Cisco w roli zastepcy protokołu RIP, jest również protokolem opartym na wektorach. Jest jednak znacznie lepszy, poniewaz:

Zapewnia stabilne trasowanie, nawet w duzych lub złożonych sieciach.

- ♦ Unika zapetlania tras, spotykanego w protokole RIP.
- ♦ Szybko reaguje na zmiany w topologii sieci.
- ♦ Mniej obciąża zasoby obliczeniowe niż RIP.
- ♦ Może równoważyć obciążenie pomiędzy trasami o mniejszej podobnej przydatności.

- ♦ Moze reagowac na warunki na lacz i poziomy ruchu sieciowego.
- ♦ Moze reagowac na rózne typy uslug.

Jedna z zalet IGRP, w porównaniu z RIP, jest zdolosc do rozpoznawania różnych typów ruchu sieciowego i tego, ze wymagaja one różnych typów sieci. Na przyklad, jesli przenosimy 300 – 400 MB plików z jednej lokalizacji do innej, pozadana jest jak największa jednostka transmisji (najwieksze pakiety, jakie mozna przeslac dana trasa), lecz dopuszczalne sa niewielkie opóznienia. Natomiast wideokonferencje wymagaja znacznie mniejszych objetosci danych, lecz wszelkie opóznienia w transmisji beda zauważalne.

Aby przystosowac sie do tych potrzeb, IGRP tworzy metryke reprezentujaca rózne wartosci, w tym:

- ♦ czas opóznienia topologii,
- ♦ przepustowosc najwolniejszego segmentu trasy,
- ♦ dostepnosc kanalów w trasie,
- ♦ niezawodnosc trasy.

Czas opóznienia topologii oznacza czas, jaki zajmuje pakietowi dotarcie do miejsca przeznaczenia w nie obciazonej sieci. Parametr ten pozwala na wziecie pod uwage w licznej metryce takich laczy, jak satelitarne, które mogą przesyłac duze objetosci danych, lecz powodują opóznienia, gdy dane przesyłane są pomiędzy naziemnymi stacjami przez satelite. Na fakt, iz wartosc opiera sie na topologii nie obciazonej, brana jest poprawka *wartosci dostepnosci kanalu*, bedacej zasadniczo biezacym wykorzystaniem pasma w procentach. Przepustowosc musi uwzgledniac trasy, które obejmują wolniejsze lacza, na przykład linie 56 kb/s. Niezawodnosc również jest ważna, wiec brana jest pod uwage liczba retransmisji. Dostepnosc kanalu i niezawodnosc są mierzone podczas komunikacji pomiędzy ruterami.

Cztery wymienione wczesniej wartosci sluza do utworzenia pojedynczej metryki, która reprezentuje jakosc trasy. Pojedyncza metryka decyduje o tym, jakie informacje zachowac w ruterze i sluzy do ustalenia, jak wysylac dane z ruterem. Poza ta metryka pomiędzy ruterami przesyłane są dodatkowo liczba hopów (liczba bram w trasie) oraz MTU (*maximum transmission unit* — maksymalna jednostka transmisji), aby rutery te mogły dokonywać optymalnych wybórów tras.

Podobnie jak w przypadku RIP, kazdy ruter uzywajacy protokolu IGRP rozglasza określowo (domyslnie co 90 sekund) cala swoja tablice tras do sasiadujacych ruterów. Tutaj również uzywana jest metoda podzialu horyzontu, omówiona wczesniej przy okazji RIP, aby zapobiegac zapetlaniu tras. Rutery, które odebraly rozwloszenie, sprawdzaja trasy i dodaja do swoich tablic tras wszelkie nowe (lub lepsze) trasy.

Rutery uzywaja tych informacji, aby ustalic najlepsza trase dla wszelkich danych, jakie musza przeslac. Kalkulacja opiera sie na wartosciach z tablicy tras i dwóch wartosciach wagowych. Pierwsza jest waga przepustowosci (WP), która sluzy do ustalenia wagosci dostepnego pasma. Druga to waga opóznien (WO), która ustala wagosc opóznien. Wzór, na podstawie którego wyliczana jest najlepsza trasa, wyglada nastepujaco:

$$((WP / (MP * (1 - ZK)) + (WO * OT)) * NT$$

W tym wzorze MP oznacza minimalna przepustowosc, ZK dostepnosc kanalu, OT opózniecie topologii i NT niezawodnosc trasy.

Trasa, dla której wyliczona wartosc bedzie najnizsza, zostanie wybrana. Jesli do sieci docelowej istnieje wiecej tras i dla dwóch lub wiecej wartosc bedzie równa i najnizsza, ruter wykorzysta obie, dzielac dane pomiedzy nie. Za pomoca tego wzoru ruter moze tez dostosowac sie do różnych typów uslug, dobierajac różne wartosci wagowe. Dodatkowo mozna uzyskac równowazenie obciazenia przez zapisywanie pełnych tras.

Dzieki wymienionym udoskonaleniom IGRP jest protokolem lepszym od RIP. Jednakże IGRP jest również wewnętrzny protokolem bramowym, czyli jest przeznaczony do użytku w pojedynczym systemie autonomicznym. W następnej kolejnosci przyjrzymy się zewnętrznemu protokołowi bramowemu OSPF.

OSPF

Na skutek ciąglego rozwoju sieci osiąga w pewnym momencie skalę, przy której nie można jej traktować jako pojedynczego systemu autonomicznego. Protokół OSPF (*Open Shortest Path First* — najpierw najkrótsza otwarta trasa) został zaprojektowany, by rozwiązać ten dylemat dzieląc sieć na obszary trasowania. OSPF używa lepszych metryk niż RIP w wersji 2. (zdefiniowanych w RFC 2238) i dodatkowo utrzymuje informacje o stanie wszystkich interfejsów we wszystkich ruterach. Dzięki temu każdy ruter OSPF może ustalić ze swojej perspektywy optymalną trasę dla danych. Protokół OSPF ma miedzy innymi następujące właściwości:

- ◆ brak ograniczeń dotyczących liczby hopów,
- ◆ obsługa VLSM,
- ◆ użycie adresowania grupowego do aktualizacji stanów łączy,
- ◆ szybsza zbieżność, ponieważ aktualizacje stanów łączy są wysyłane natychmiast,
- ◆ metryki obejmują informacje o opóźnieniach łączy, nie tylko o liczbie hopów,
- ◆ obsługa równowazenia obciążenia,
- ◆ podział sieci na obszary, które mogą zawierać do ok. 50 ruterów,
- ◆ uwierzytelnianie otwartym tekstem lub za pomocą algorytmu mieszania *message-digest*,
- ◆ znaczniki tras zewnętrznych.

Jak Czytelnik zapewne się domyślił, z funkcjami tymi są związane pewne koszty — po pierwsze, ruch w sieci pochodzący od stosowanego OSPF (ponieważ każde łącze w każdym routerze musi być monitorowane przez wszystkie routery w danym obszarze, obsługa infrastruktury trasowania może wymagać dodatkowych transmisji). Po drugie — nakłady pracy na planowanie, które jest wymagane przy konfiguracji wdrożenia OSPF. Na koniec, ponieważ routery obliczają dla siebie „najlepsze trasy”, same routery muszą posiadać wiele zasobów — szybsze procesory i więcej pamięci.

Stany i koszty lacz

Kazdy ruter zajmuje się zarządzaniem i propagacją informacji o stanie lacz, co oznacza, że dla każdego połączenia logicznego gromadzi kilka informacji (adres IP, maska podsieci, topologia używana w laczu, inne dostępne routery używające tego lacz itp.). Informacje te są propagowane do wszystkich routerów w obszarze za każdym razem, gdy stan lacz ulega zmianie.

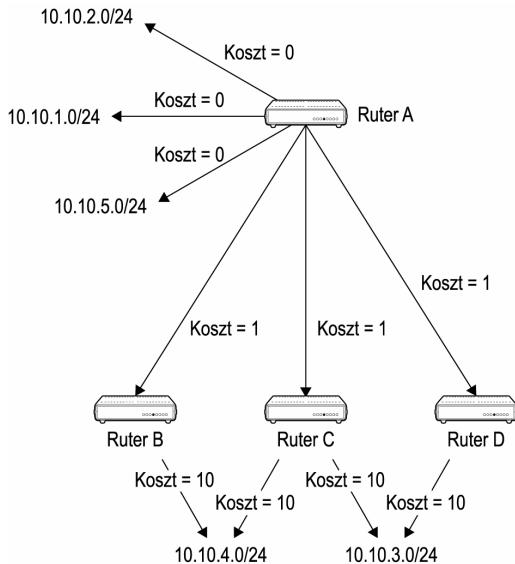
Gdy ruter jest inicjowany lub stan jednego z jego lacz ulega zmianie, wówczas tworzy ogłoszenie stanów lacz, które zostaną wysłane do wszystkich sąsiednich routerów, na przykład RIP lub IGRP. Sąsiadujące routery zapisują kopie tych informacji i przekazują ogłoszenie do wszystkich innych znanych sobie sieci — proces ten nosi nazwę *trasowania rozpływowego (flooding)*. Po aktualizowaniu bazy danych każdy ruter przelicza ponownie swoje drzewo najkrótszych tras, które stanowią listę sieci docelowych, skojarzonych kosztów i kolejnych hopów.

Podczas obliczania drzewa najkrótszych tras musi zostać wzięty pod uwagę koszt każdego lacz. Ponieważ każdy ruter w rzeczywistości zna typ połączeń, jakie mają pozostałe routery w obszarze, OSPF może obliczyć koszt całej trasy, nie tylko koszt oparty na wartości z sąsiedniego routera. Koszt jest wyliczany wyłącznie na podstawie odwrotności przepustowości lacz. Za szybkie lacz uznaje się 100 Mb/s, więc dla połączenia 100 Mb/s koszt wynosi 1. Koszt połączenia 10 Mb/s wynosi $100/10 = 10$, zaś koszt lacz T1 o przepustowości 1,544 Mb/s wynosi $100/1,544 = 64$.

Podczas tworzenia drzewa najkrótszych tras ruter „zakłada”, iż jest centrum sieci i że znajdzie najkrótszą trasę do dowolnej sieci poprzez swojego sąsiada. Rozważmy sieć z rysunku 19.15.

Rysunek 19.15.

Diagram przedstawiający drzewo najkrótszych tras dla routera A



Na tym rysunku ruter buduje drzewo, aby znalezc wszystkie mozliwe trasy i przydzielic wartosci kosztow do wszystkich laczy w trasie. Koszt dla sieci lokalnych wynosi 0, zas dla pozostalych — jest wyliczany na podstawie przepustowosc, jak omowilismy to przed chwila. Ten proces doprowadzi do utworzenia drzewa, ktore bedzie wygladalo jak na rysunku 19.15.

Korzystajac z drzewa najkrótszych tras, ruter zbuduje teraz swoja tablice tras. W naszym przykladzie beda dwie trasy do 10.10.3.0 i 10.10.4.0, co pozwoli ruterowi równowazyc obciaszenie pomiedzy dwie nadmiarowe trasy.

W stabilnej sieci OSPF sprawuje sie dobrze i nie uzywa nadmiernych ilosci pasma. Jednakze w sieci niestabilnej OSPF moze sprawic problemy z wydajnoscia, z powodu zalewów (trasowania rozplywowego) i stalego odtwarzania drzew najkrótszych tras i tablic tras.

Laczanie obszarów w siec

Jak juz wspomniano, OSPF pozwala na podzial systemu autonomicznego na obszary, co oznacza mozliwosc rozbudowy sieci do olbrzymich rozmiarow. W zaleznosci od topologii i stabilnosci, „realistyczna” liczba ruterow w dowolnym obszarze wynosi okolo 50. Powyzej tej liczby „wrodzona” niestabilnosc sieci i liczba ruterow, ktore trzeba aktualizowac, zaczynaja powaznie wpływanie na wydajnosc.

Wprawdzie 50 ruterow moze wydawac sie duza liczba, lecz wiele sieci znacznie ja przekracza. W ich przypadku trzeba podzielic siec na osobne obszary. Podzialu moze my dokonac tez z innych powodow, na przyklad z uwagi na odrebnie administrowane obszary.

W gruncie rzeczy obszar ogranicza zasieg zalewów (*flooding*). Aktualizacje stanu laczy sa wysylane tylko do ruterow w tym samym obszarze, co uniemozuyla znalezienie pelnej trasy do hosta w innym obszarze. Zamiast tego rutery obliczaja najlepsze trasy do *ruterow brzegowych obszaru*. Te rutery sluzsa nie tylko do przesyłania danych pomiedzy obszarami, lecz również do wymiany informacji o trasach w obszarach, z którymi sie lacza. Rutery wewnatrz obszaru mozna nazwac *ruterami wewnetrznymi*, aby odróżnic je od ruterow brzegowych.

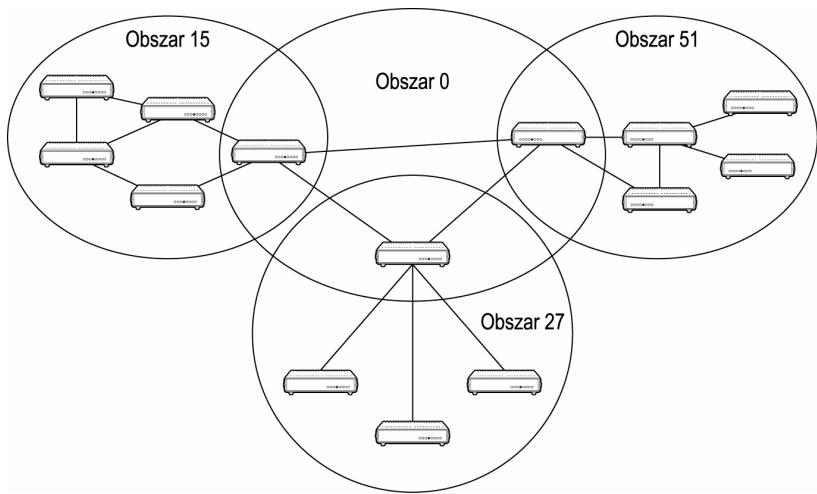
Jeden obszar w sieci OSPF jest wyznaczony do roli szkieletu (Obszar 0), z którym powinny laczyc sie wszystkie rutery brzegowe obszarow. Ten wspólny obszar pozwala ruterom brzegowym utworzyc sume informacji o laczach w obsługiwanych przez nie obszarach; te informacje moga zostac udostepnione innym ruterom brzegowym obszarow poprzez Obszar 0. Rysunek 19.16 przedstawia, jak moze wygladac Obszar 0.

W niektórych przypadkach podlaczenie wszystkich ruterow brzegowych obszarow do Obszaru 0 jest niemozliwe — na przyklad, jesli siec jest geograficznie rozproszona na wiele różnych regionow, z niewielka liczba szybkich polaczen pomiedzy grupami lokalizacji. W takich przypadkach tworzone sa lacza wirtualne.

Gdy polaczenie fizyczne nie jest mozliwe, lacza wirtualne moga posluzyc do polaczenia ze soba odleglych fragmentow Obszaru 0. Lacze wirtualne moze przebiec przez inny obszar, polaczony z wszystkimi fragmentami Obszaru 0 — posiadajacy w czesci wspólnej z kazdym z tych fragmentow ruter brzegowy. Lacze wirtualne jest wówczas tworzone pomiedzy tymi ruterami brzegowymi obszaru.

Rysunek 19.16.

Sieć OSPF moze zostać podzielona na obszary, aby zmniejszyć obciążenie sieci przez trasowanie rozpływowe, oraz aby zmniejszyć obciążenie ruterów spowodowane przeliczaniem tabel tras



Ten sam proces moze posluzyc do polaczenia odlegiego obszaru z Obszarem 0. W tym celu pomiedzy obszarem odleglym i drugim, polaczonym z Obszarem 0, zostaje umieszczony ruter brzegowy, który moze byc wykorzystany do utworzenia wirtualnego lacza z ruterem brzegowym, który jest podlaczony bezposrednio do Obszaru 0.

Inne funkcje ruterów

Czytelnik powinien rozumieć juz różnice pomiędzy ruterami wewnętrznyimi i brzegowymi obszaru. Jednakże ruter moze grac jeszcze inne role, które nie zostały tu omówione. Na przykład, moze grac role rутera brzegowego systemu autonomicznego (ASBR — *Autonomous System Border Router*). ASBR sluzi do laczenia systemu autonomicznego z innymi systemami autonomicznymi, które uzywaja innych protokolów.

Trasy, które pozna ASBR, sa rozprowadzane w sieci. Trasy sa streszczone przez ASBR, poniewaz rutery te moga byc podlaczone do bardzo duzych sieci. Proces umieszczania tych zewnetrznych laczy w sieci OSPF nosi nazwe redystrybucji. Informacje sa gromadzone z innych protokolów, na przykład RIP lub IGRP, a nastepnie redystrybuowane w sieci OSPF.

Rutery musza również znajdowac inne rutery, z którymi dziela polaczenia. Inaczej mówiąc, ruter powinien „wiedzieć” o wszystkich pozostałych ruteraach we wspólnych segmentach fizycznych. W OSPF proces znajdowania innych ruteraów nosi nazwe *procesu znajdowania sąsiadów* (*neighbour process*). Proces ten wykorzystuje pakiety Hello, które sa prostymi pakietami, zawierajacymi liste znanych sąsiadów i inne informacje o ruterze wysylajacym pakiet. Pakiety Hello sa wysylane okresowo do adresów grupowych. Po odebraniu pakietu Hello dodawane sa wszelkie systemy nie znajdujące sie na liscie sąsiadów. Aby dwa rutery mogły zostać sąsiadami, muszą znajdować się w jednym obszarze (aby mogły uwierzytelniac sie u siebie nawzajem). Ponadto rutery te uzgadniają interwally *Hello* (interwał przywitania) i *Dead* (czas zwłoki). Interwał przywitania oznacza częstotliwość, z jaką ruter wysyla pakiety Hello. Czas zwłoki oznacza czas, przez który ruter zachowuje sąsiada na liscie, jeśli nie otrzymuje od niego wiadomości.

Rutery stosuja proces znajdowania sasiadów do wzajemnej identyfikacji, a nastepnie wybieraja „rzecznika”, inaczej ruter desygnowany (DR — *Designated Router*) dla danej grupy sasiadów. DR koordynuje aktualizacje informacji w calym obszarze. Inaczej mówiac, kazdy ruter przesyła aktualizacje do DR, który nastepnie rozsyła je do wszystkich pozostalych ruterów. DR jest ruterem o najwyzszym priorytecie OSPF. Istnieje jeszcze *zapasowy ruter desygnowany* dla danej grupy sasiadów, który jest uzywany w razie awarii DR.

Po wyborze DR wszystkie pozostałe rutery z danego obszaru usiluja uformowac przyleglosc z tym ruterem. Ten proces idzie dalej niz zwykłe pakiety Hello i pozwala ruterom wymieniac miedzy soba bazy danych. Gdy proces ten zostaje uruchomiony, przyleglosc jest oznaczona jako nieczynna. W trakcie procesu znajdowania sasiadów przyleglosc jest oznaczona jako *Init* — co oznacza, ze pakiety Hello zostaly zauważone — a nastepnie jako *Two-Way* (dwukierunkowa), gdy sasiedzi zostana znalezione. Przyleglosc naprawde zaczyna sie w procesie znajdowania sasiadów. Rutery przechodza nastepnie w stan wymiany (*Exchange*), w którym wysylaja do siebie nawzajem bazy danych. Dwa rutery pobierajace od siebie informacje wchodza w stan *Loading* (ladowania). Na koniec przyleglosc zostaje oznaczona jako pelna (*Full*) i kazda para baz danych zawiera wszystkie informacje o obu ruterach.

Oczywiscie proces znajdowania sasiadów i budowania przyleglosci wymagaja stosowania mediów, które przenosza rozgloszenia. Niektóre nosniki, na przyklad frame relay i ATM, nie uzywaja rozgloszen, co oznacza, ze w kazdym ruterze trzeba skonfigurowac adresy IP sasiadów. Konfiguracja dwupunktowego podinterfejsu lub interfejsu punkt-grupa również nadaje sie do tego, poniewaz podaja one ruterom adresy sasiadów. Przez skonfigurowanie dwupunktowego podinterfejsu pomiedzy kazdym ruterem i DR oraz zapasowym ruterem desygnowanym, mozemy zapewnic aktualizowanie wszystkich baz danych. Wykorzystanie interfejsu punkt-grupa zmniejsza liczbe podinterfejsów, które trzeba skonfigurowac.

Rozdział 20.

Planowanie rozmieszczenia serwerów

W tym rozdziale:

- ◆ Ustalenie usług potrzebnych w sieci
- ◆ Planowanie nadmiarowości i równowżenia obciążenia

Dwa poprzednie rozdziały zajmowały się budowaniem sieci — nie fizyczna instalacja okablowania, lecz zagadnieniami adresowania i trasowania TCP/IP, które umożliwiają komunikacje. W niniejszym rozdziale zwróciśmy uwagę na serwery, których klienci będą używać w sieci.

Omówimy trzy typy serwerów, jakie można znaleźć w sieciach, oraz zagadnienia wydajności dla każdego typu. Ponadto przyjrzymy się różnym metodom równowżenia obciążenia i uzyskiwania nadmiarowości. Lecz na początek zobaczymy, jak ustalić listę usług potrzebnych w naszej sieci.

Ustalenie usług potrzebnych w sieci

W najprostszej formie sieć jest środkiem, za pomocą którego użytkownicy korzystają z dostępu do wspólnych zasobów. Zasobem takim może być, na przykład, udostępniona drukarka lub serwer plików. Krótko mówiąc, program kliencki zgłasza zadania pod adresem innego procesu, który jest uruchomiony w innym systemie. Można powiedzieć, że sieć pozwala dzielić prace dwóm procesom w dwóch różnych komputerach.

Gdy bedziemy przyglądać się różnym serwerom, które mogą być potrzebne w sieci, czytelnik powinien pamiętać, po co mają one zostać w sieci zainstalowane. Serwery udostępniają użytkownikom usługi, wobec tego powinniśmy skoncentrować się na serwerach, z których użytkownicy będą korzystać. Wielkość użytkowników wystarcza po prostu możliwość napisania listu, wydrukowania go i skorzystania z poczty elektronicznej. Inaczej mówiąc, jeśli chcemy uzyskać spójność społeczności użytkowników, koncentrowanie się na serwerze potokowej transmisji danych, który umożliwi emisję prezentacji wideo w sieci, nie jest przypuszczalnie najlepsza strategia.

Wobec tego musimy realistycznie ustalić typy serwerów, jakie będą potrzebne w sieci, aby zaspokoić potrzeby użytkowników. Do typowych potrzeb użytkowników sieci, oprócz najbardziej podstawowych — drukowania i korzystania z poczty elektronicznej,

nalezy zdolnosc otrzymania adresu IP przez komputer i rozwiazywania nazw (DNS lub NetBIOS) na adresy IP. Zapewniajac, by uslugi niezbedne dla uzytkownikow i sieci byly zawsze dostepne, zmniejszymy liczbe reklamacji i prosb o pomoc techniczna ze strony uzytkownikow.

Dysponujac podstawowa lacznoscia, uzytkownicy moga zidentyfikowac sie w sieciowym serwerze uwierzytelniania, a nastepnie laczyc z serwerami oferujacym faktyczne uslugi. Serwery, ktore udostepniaja uzytkownikom uslugi dziela sie na dwie glowne kategorie: *serwery plikow* (tu zaliczane sa rowniez serwery drukowania) oraz *serwery aplikacji*. Oznacza to, ze tak naprawde istnieja trzy typy serwerow:

- ◆ *Serwery infrastruktury sieciowej* — do tej grupy naleza serwery DHCP, BOOTP, DNS, WINS, NIS, NDS i kontrolery domen.
- ◆ *Serwery plikow i drukowania* — definicja tej grupy jest oczywista — serwery plikow i serwery drukarek.
- ◆ *Serwery aplikacji* — do tej grupy moga nalezac serwery baz danych, serwery pocztowe, serwery pracy grupowej i tak dalej.

Dobór serwerów w sieci zalezy przede wszystkim od dwóch czynników: typu posiadanych klienckich systemów operacyjnych i wymagan uzytkowników. Ogólnie mówiąc, w praktycznie kazdej sieci potrzebne sa pewne podstawowe uslugi, obejmujace:

- ◆ *DHCP* — *Dynamic Host Configuration Protocol* (protokół dynamicznej konfiguracji hosta) sluzy do przydzielania adresów IP i różnych szczegółów konfiguracji, w zależności od systemu operacyjnego klienta. Ogólnie mówiąc, kazdy klient musi otrzymac przynajmniej adres IP, a wiekszość wymaga maski podsieci, adresu serwera DNS i adresu bramy lub rutera. Zapewnia to podstawowa lacznosc, potrzebna wszystkim systemom, i oszczedza administratorom pracy zwiazanej z indywidualnym konfigurowaniem wszystkich klientów.
- ◆ *DNS* — *Domain Name System* (system nazw domen) zapewnia odwzorowanie nazw pełnych złożonych (FQDN) hostów na adresy IP. Oznacza to, że klienci mogą łatwo znajdować serwery na podstawie prostych do rozpoznania nazw, niezależnie od położenia.
- ◆ *WINS* — usługa *Windows Internet Name Service* jest potrzebna, aby rozwiazywać nazwy NetBIOS na adresy IP.



Wprawdzie można sobie wyobrazić sieć nie zawierającą klientów NetBIOS, lecz są to rzadkie przypadki. Jeśli nie posiadamy serwerów NetBIOS, serwer WINS jest niepotrzebny.

- ◆ *Serwery plików i drukowania* — niezależnie od używanego sieciowego systemu operacyjnego, uzytkownicy zwykle chcą udostępniać sobie nawzajem pliki, a przynajmniej pobierać potrzebne pliki z serwera. Wobec tego serwer plików jest niezbędny.
- ◆ *Poczta* — zdolność do wysyłania i odbierania poczty stała się absolutnie niezbędna w środowiskach biurowych. Pracownicy oczekują, iż otrzymają możliwość

wymiany listów elektronicznych z pracownikami innych biur i przedsiębiorstw. Wobec tego utworzenie wydajnego systemu poczty elektronicznej jest istotne.

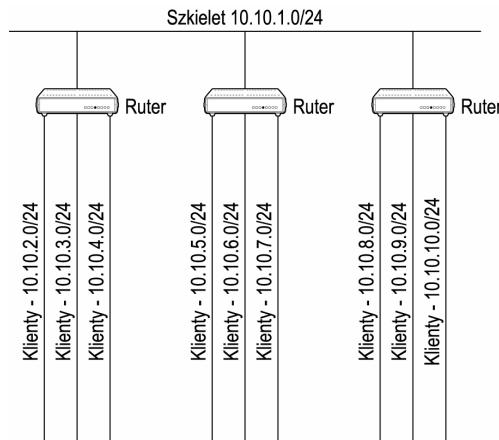
Powyzsze usługi stanowią podstawowy poziom wsparcia funkcjonowania udanej sieci. Musimy szukać sposobów, by te usługi poradziły sobie z obciążeniem powodowanym przez użytkowników. Wymaga to dobrego zrozumienia każdej usługi, w tym używanych przez nią protokołów oraz poznania objętości ruchu sieciowego, jaki generują te protokoly.

Przyjrzymy się teraz usługom, których potrzebuje sama sieć.

Instalowanie usług w sieci

Na początek analizy usług wymaganych w funkcjonalnej sieci, spójrzmy na przykład podstawowej sieci. Rysunek 20.1 przedstawia prostą sieć z dziewięcioma podsieciemi klienckimi i jedna szkieletowa. W tym przypadku możemy założyć, że sieć obsługuje typowe małe biuro i około 450 użytkowników. Oznacza to przeciętnie 50 klientów w każdej podsiedzi.

Rysunek 20.1.
Przykładowa sieć
potrzebująca serwerów



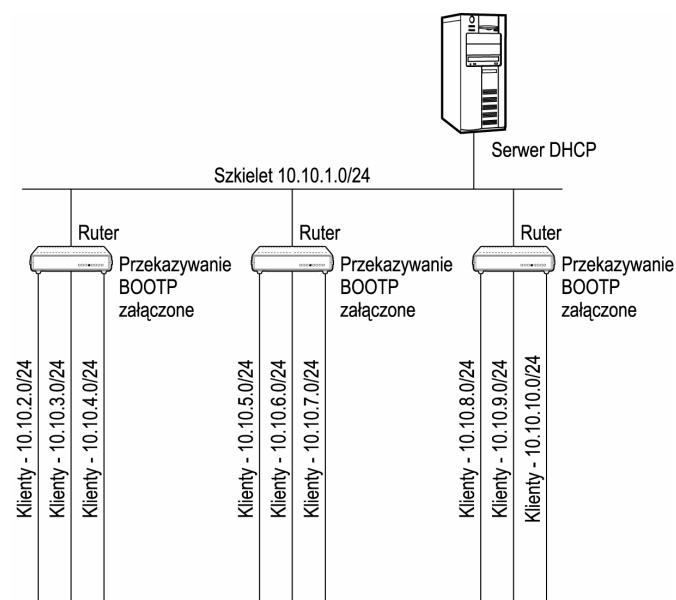
Zaczniemy od zapotrzebowania na usługę DHCP. Jak Czytelnik zapewne pamięta, DHCP jest prostym protokołem, który zużywa bardzo niewiele przepustowości sieci. Ponadto do uruchomienia tej usługi potrzeba bardzo niewiele zasobów, więc serwer nie musi być zbyt duży. W naszym przypadku wymagany jest tylko jeden serwer DHCP, ponieważ pojedynczy serwer DHCP jest w stanie obsłużyć 450 klientów. Musimy jednak uwzględnić fakt, że DHCP używa adresu rozgłoszeniowego; w przeciwnym razie klienci nie byliby w stanie uzyskać adresu od DHCP, ponieważ sieć zawiera routery, a te domyślnie nie przekazują ruchu rozgłoszeniowego.

Dostępne są dwie metody pozwalające skorzystać z usługi DHCP w takim środowisku z routerami. Pierwsza polega na użyciu agentów przekazujących DHCP jako pośredników pomiędzy klientami i serwerem DHCP. Ponieważ agent przekazujący potrafi skierować rozgłoszenie do serwera DHCP, ten możemy bez problemów umieścić w innej podsiedzi. Metoda druga (prostsza) polega na załączeniu przekazywania BOOTP w routeraх, co pozwoli ruterom przesyłać zadania do serwera DHCP w innej podsiedzi.

W naszym przypadku zalaczmy przekazywanie BOOTP w ruterach i umiescimy pojedynczy serwer DHCP w sieci szkieletowej, jak na rysunku 20.2. Standardowo funkcje przekazywania BOOTP musimy w ruterach zalaczyc. Domyslnie jest ona wylaczena, poniewaz przekazywanie BOOTP zwiększa liczbę pakietów, z którymi ruter musi sobie poradzic oraz moze doprowadzic do zatorów w ruterze.

Rysunek 20.2.

Przykladowa siec
z dodanym
serwerem DHCP



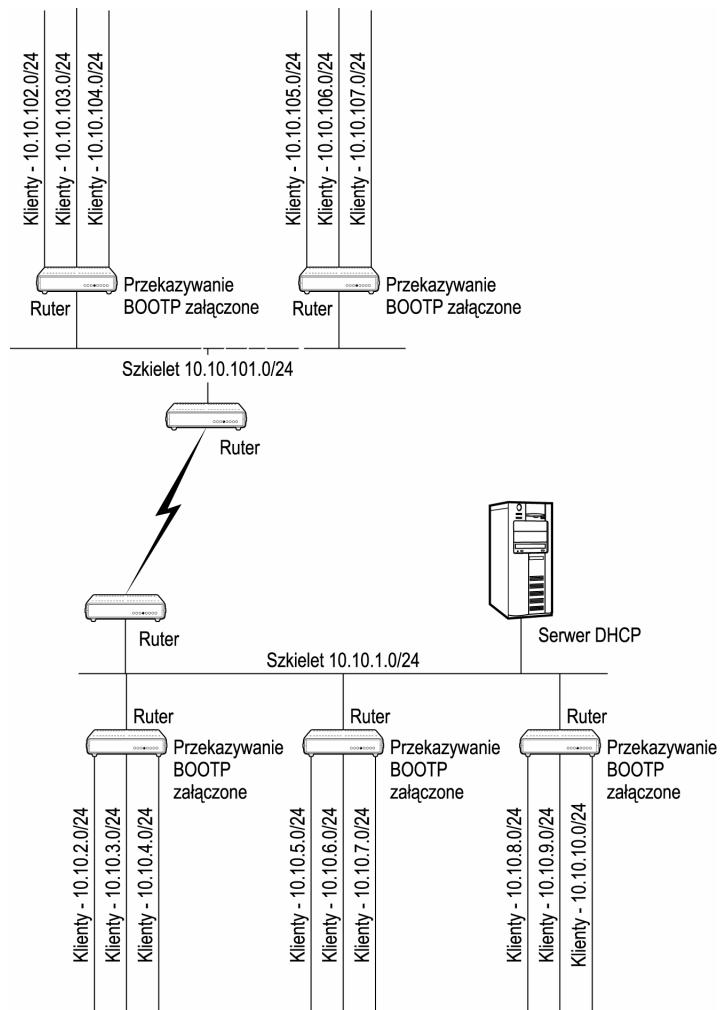
Teraz wszystkie klienty beda mogly zazadac od DHCP adresu, zas routery beda przekazywac te zadania do szkieletu, gdzie serwer DHCP bedzie w stanie je obsluzyc (pod warunkiem, ze posiada zakres odpowiedni dla klienta).

To byl prosty przyklad. Jednak podczas projektowania sieci musimy przyjrzec sie wszelkim scenariuszom typu „co by bylo, gdyby...”. Oznacza to pytania typu: Co by sie stalo w razie awarii serwera? A w wypadku awarii rutera? A gdyby dodac kolejna podsieć lokalnie lub zdalnie? Wprawdzie rozwiazanie z rysunku 20.2 jest funkcjonalne, lecz wezmy pod uwage bardziej zlozona siec z rysunku 20.3, w której dodano zdalna lokalizacje.

Na rysunku 20.3 serwer DHCP nie jest w stanie udostepnic adresów dla dwóch zdalnych podsieci, poniewaz ich routery nie przekazują pakietów BOOTP. Nawet gdyby to czynily, musimy zastanowic sie, czy chcemy przesyłac pakiety tam i z powrotem przez laczne komunikacyjne, co zwiększyloby w nim ruch i zwolnilo caly proces. Te czynniki wskazuja na potrzebe zainstalowania drugiego serwera DHCP po zdalnej stronie polaczenia, obsługujacego klienty po tamtej stronie. Gdyby jednak pomyslec o opisanych przed chwilą scenariuszach, zobaczymy, ze w razie awarii jednego z pieciu ruterów czesc klientów nie bylaby w stanie w ogole uzyskac adresu IP.

Rysunek 20.3.

*Przykładowa sieć
o wiekszej złożoności*



Jak widać, samo umieszczenie usługi w sieci nie wystarczy, by zapewnić jej stałą dostępność, niezależnie od dowolnego punktu awarii. Jeśli chcemy uodpornić sieć na pojedyncze punkty awarii, musimy zaplanować nadmiarowość usług sieciowych. Skala trudności tego zadania rośnie z liczbą potrzebnych nam usług.

Laczenie usług

Na rysunku 20.3 trzeba dodać drugi serwer w zdalnej sieci, aby obsługiwał DHCP. Musimy dodać także dodatkowe serwery sieciowe, na przykład DNS, WINS i uwierzytelniające serwery sieciowe.

Najprostsza sieć zawiera serwery DHCP, DNS, plików i drukowania. Aby zapewnić nadmiarowość dla tych usług, potrzebnych byłoby sześć serwerów: dwa DHCP, dwa DNS i dwa plików i drukowania. Dodając kolejne sieci zdalone, jak na rysunku 20.3, musimy dodać trzy kolejne serwery, udostępniające te podstawowe usługi w sieci zdalonej.

Dodawanie serwera dla kazdej kolejnej uslugi w sieci jest kosztowne, w coraz wiekszym stopniu utrudnia zarzadzanie i zwiększa ruch sieciowy tla w sieci. Aby zmniejszyc licze serwerów w sieci, które wymagaja zarzadzania i generuja ruch sieciowy tla, mozemy polaczyc kilka uslug w jednym systemie. W rzeczywistych warunkach jest to normalne postepowanie i w praktyce mozemy obsluzyc cala siec przez pojedynczy serwer, jesli tylko siec jest wystarczajaco mala. Jedynym problemem jest tu umieszczenie wszystkich uslug razem. Inaczej mówiac, gdy ten pojedynczy serwer zawiedzie, wszystkie uruchomione w nim uslugi przestana byc dostepne.

Decydujac o tym, jakie uslugi polaczyc w jednym systemie, musimy wziac pod uwage, jakich zasobow wymaga kazda z uslug. Dla kazdej uslugi system potrzebuje czterech podstawowych zasobow:

- ♦ *Procesor* — uslugi typu DHCP, DNS, WINS i podobne uslugi zarzadzajace krótkimi listami nie wymagaja zbyt duzej mocy obliczeniowej. Uslugi obsługujace duże listy (bazy danych), jak np. serwery Oracle i MS SQL, wymagaja sporej mocy procesorów.
- ♦ *Pamiec* — wieksza ilosc pamieci w kazdym przypadku pomoze serwerowi. Czesto nie zauważa sie faktu, iz CPU nie komunikuje sie z dyskiem, klawiatura lub z czymkolwiek innym — tylko z pamiecia. Dla systemu przeniesienie danych w pamieci, na przyklad z listy adresów do bufora, w którym budowany jest pakiet odpowiadajacy na zapytanie DNS, jest czynoscia szybka i prosta.
- ♦ *Dysk* — wydajnosc podsystemu dyskowego jest w niektórych przypadkach bardzo wzazna — zwlaszcza w serwerach baz danych, na przyklad Oracle lub MS SQL, które mogą posiadac terabajty danych. Dla serwerów typu DNS, WINS lub DHCP objetosc danych jest tak mala, ze przestrzen dyskowa nie jest czynnikiem krytycznym
- ♦ *Siec* — zadaniem sieci jest przesyłanie danych, wobec tego zdolosc do przenoszenia danych pomiedzy serwerem i siecią jest bardzo wzazna. Musimy zastosowac dobrą karte sieciowa, poniewaz to za jej pomoca serwer bedzie wlaczony do sieci. Jesli przylaczmy serwer do koncentratora o przepustowosci 10 Mb/s, to bedzie musial rywalizowac o pasmo z wszystkimi innymi urzadzeniami przylaczonymi do tego koncentratora. Natomiast, jesli przylaczmy serwer do przejacznika 100 Mb/s, wówczas nie bedzie mial praktycznie z czym rywalizowac. Wobec tego polaczenie, typ uzytej karty i liczba kart sieciowych w systemie wpływaja na zdolosc do przesyłania danych, a co za tym idzie, na wydajnosc serwera.



Prosze pamietac, ze elektrony w przewodach poruszaja sie z predkoscia porównywalna z predkoscia swiatla, natomiast glowica zapisu-odczytu dysku twardego nie moglaby poruszac sie równie szybko bez zlamania praw fizyki. W teorii, gdyby glowica spróbowala zrobic cos takiego, jej masa wzrosłaby do masy planety, powodując kolaps grawitacyjny i powstanie czarnej dziury w systemie. Wniosek: aby zwiększyć wydajnosc, potrzebujemy dodatkowej pamieci — a zwiększenie pamieci wirtualnej (pliku wymiany na dysku) to nie to samo, co dokupienie RAM-u.

- ♦ *Siec* — zadaniem sieci jest przesyłanie danych, wobec tego zdolosc do przenoszenia danych pomiedzy serwerem i siecią jest bardzo wzazna. Musimy zastosowac dobrą karte sieciowa, poniewaz to za jej pomoca serwer bedzie wlaczony do sieci. Jesli przylaczmy serwer do koncentratora o przepustowosci 10 Mb/s, to bedzie musial rywalizowac o pasmo z wszystkimi innymi urzadzeniami przylaczonymi do tego koncentratora. Natomiast, jesli przylaczmy serwer do przejacznika 100 Mb/s, wówczas nie bedzie mial praktycznie z czym rywalizowac. Wobec tego polaczenie, typ uzytej karty i liczba kart sieciowych w systemie wpływaja na zdolosc do przesyłania danych, a co za tym idzie, na wydajnosc serwera.

Podczas laczenia uslug w serwerze nalezy wyszukiwac takie, które nie rywalizuja ze soba o zasoby. W wiekszosci przypadkow mozemy łatwo polaczyc DNS, DHCP, WINS i uwierzytelnianie sieciowe. W niektórych przypadkach polaczenie uslug jest posunięciem bardzo oplacalnym. Na przyklad, usluga DDNS (dynamiczny DNS) współpracuje

z serwerem DHCP, rejestrując pary nazwa - adres IP, jeśli więc zainstalujemy obie usługi w jednym systemie, ich współpraca nie będzie generować żadnego ruchu w sieci. W systemach Microsofta można skonfigurować DNS tak, by nie znalezione adresy sprawdzały w usłudze WINS, więc połączenie tych dwóch usług też jest sensowne.

Po podjęciu decyzji, jakie usługi połączyc, musimy po prostu ustalić charakterystyki obciążenia systemu, który będzie je obsługiwać. W tym celu możemy korzystać z różnych narzędzi, zależnie od używanej platformy. Generalnie jednak powinniśmy przyjąć się omówionym poprzednio czterem kluczowym zasobom systemu, najpierw dla jednego klienta, następnie dla coraz większych grup klientów. W pewnym momencie jeden z zasobów osiągnie swój limit.



Proszę pamiętać, że liczba jednocześnie używających się użytkowników może przekroczyć maksimum, z którym mógłby poradzić sobie system. Ponieważ bardzo rzadko (lub nigdy) wszyscy użytkownicy będą połączeni z serwerem jednocześnie, możemy do pewnego stopnia nadmiarowo wykorzystać serwery. Przy planowaniu maksymalnej liczby użytkowników należy zwracając uwagę na wzory wykorzystania serwera — a przynajmniej pamiętać o nich. Weźmy na przykład serwer plików i drukowania: jest bardzo mało prawdopodobne, iż wszyscy użytkownicy będą równocześnie zapisywać w nim plik przez sieć.

Spójrzmy na przykład na serwer DHCP. W danej sieci klienci zwykłe pozostają w jednym miejscu, więc przedłużymy okres dzierżawy do osmu dni. Jak Czytelnik pamięta z opisu DHCP w rozdziale 9., ustawienie okresu dzierżawy na osiem dni oznacza, iż klienci muszą odnawiać dzierżawy co cztery dni. Następnie musimy przyjrzeć się serwerowi i ustalić liczbę równoczesnych połączeń, które jest w stanie obsłużyć. Załóżmy, że serwery, które chcemy wykorzystać są zdolne do równoczesnej obsługi 6000 zadań klientów.

Aby oszacować liczbę klientów DHCP, jaka serwer DHCP jest w stanie obsługiwać, musimy ustalić ile czasu zajmie serwerowi DHCP obsługa zadania odnowienia dzierżawy (wybraлиmy proces odnowienia dzierżawy, ponieważ jest najczęściej używany). Odnowienie dzierżawy DHCP wymaga odbioru pakietu, kontroli dzierżawy, obliczenia nowej daty wygasnięcia dzierżawy i odesłania pakietu do klienta. Do ustalenia dokładnego czasu mogą posłużyć analizatory pakietów sieciowych lub monitor wydajności systemu. W praktyce jednak ten proces nie powinien zająć wiecej niż jedna sekundy. Wobec tego, na potrzeby przykładu założymy taką wartość.

Mamy już wszystkie dane, więc musimy połączyc je w celu ustalenia maksymalnej liczby klientów, jaka może obsługiwać jeden serwer DHCP. Wzór wygląda tak:

$$\begin{aligned} \text{procesy/h} &= 3600 \text{ sekund / czas wykonania procesu} \\ \text{klienci/h} &= \text{procesy/h} * \text{liczba równoczesnych zadań} \\ \text{suma klientów} &= \text{okres odnawiania [h]} * \text{klienci/h} \end{aligned}$$

Jeśli liczby z naszego przykładu wstawimy do równania, otrzymamy:

$$\begin{aligned} \text{procesy/h} &= 3600/1 \\ \text{klienci/h} &= 3600 * 6000 \\ \text{suma klientów} &= 96 * 21\,600\,000 \end{aligned}$$

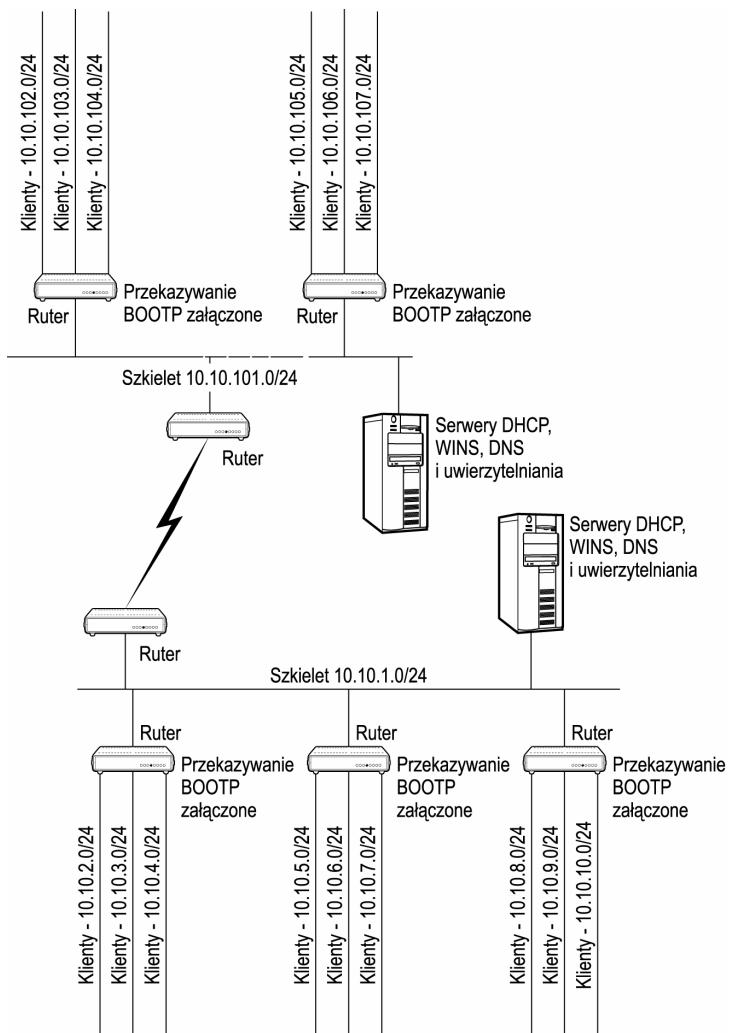
Maksymalna liczba klientów, jaka może obsługiwać serwer DHCP w naszym przypadku wynosi 2 073 600 000. Oczywiście w rzeczywistych warunkach nigdy nie dojdziemy do takiej wartości. Jeśli jednak zakładana liczba klientów używających serwera DHCP po-

dzielimy przez powyzszy wynik, otrzymamy w przyblizeniu odsetek zasobów systemu, jakiego usluga (w tym przypadku DHCP) bedzie uzywac. Szacujac przyblizony odsetek dla kazdej uslugi, bedziemy mogli ocenic, jak obciążony bedzie serwer.

Na rysunku 20.4 do przykładowej sieci zostały dodane wszystkie podstawowe usługi sieciowe: DHCP, DNS, WINS i uwierzytelnianie. Na tym etapie awaria routera lub jednego z systemów usług sieciowych spowoduje, iż część użytkowników sieci nie będzie mogła skorzystać z tych usług. Co więcej, gdyby nastąpiła awaria zasilania, a wszystkie klienty podczas uruchamiania pobierały adres z DHCP, rejestrowały się w usługach WINS i DDNS, a następnie uwierzytelniali w serwerze, serwery te mogłyby zostać przeciążone po przywróceniu zasilania. Inaczej mówiąc, gdy mamy już w sieci wszystkie podstawowe usługi, pora zaczynać planować równowagę obciążenia i nadmiarowość.

Rysunek 20.4.

Przykładowa sieć zainstalowanymi wystarczającymi usługami sieciowymi



Planowanie równowazenia obciążenia i nadmiarowości

Wprawdzie są to dwa odrebnego zagadnienia, lecz zwykle planowane wspólnie — zazwyczaj osiągnięcie jednego równoczesnie zapewnia drugie. Równoważenie obciążenia jest procesem podziału obciążenia ze strony klientów na wiele serwerów, tak by wydajność widziana przez każdego klienta była mniej więcej taka sama. Nadmiarowość zapewnia dostępność usługi niezależnie od awarii urządzenia lub segmentu sieci.

Na przykład, jednym z najprostszych sposobów zapewnienia nadmiarowości jest dodanie drugiego systemu, udostępniającego tę samą usługę, dla której chcemy uzyskać nadmiarowość. Ponieważ możemy sprawić, aby połowa klientów korzystała z jednego serwera, a reszta z drugiego, dodatkowo zaimplementujemy równoważenie obciążenia.

W celu otrzymania nadmiarowości i (lub) równowagi obciążenia możemy zrobić cztery rzeczy:

- ◆ dodać więcej systemów,
- ◆ dodać więcej kart sieciowych do jednego komputera, czyniąc z niego system wieloadresowy,
- ◆ utworzyć hierarchię serwerów,
- ◆ zastosować grupowanie (klastry) serwerów.

W następnych punktach przyjrzymy się dokładniej tym metodom.

Dodawanie kolejnych systemów

Plan minimum zakłada, że każda usługa potrzebna w sieci musi być uruchomiona w przynajmniej dwóch różnych komputerach. Dyktuje to po prostu zdrowy rozsadek — gdyby sieć polegala całkowicie na usłudze, a ta by zawiodła, wówczas sieć również przestanie działać, a telefony zaczynały się urywać. Dodanie kolejnych systemów jest korzystne i zwykle zapewnia wymaganą nadmiarowość. Jednakże dodanie następnego systemu wymaga również zapewnienia metody synchronizacji nowego serwera z pozostałymi.

Załóżmy na chwilę, iż posiadamy serwer pocztowy o nazwie Mercury i adresie IP 10.10.1.58. Decydujemy się przenieść go o jedną podsieć bliżej użytkowników i otrzymuje adres IP 10.10.52.100. W sieci znajdują się dwa serwery DNS, a zmieniliśmy adres IP tylko w głównym serwerze DNS. Wszystkie klienci znajdują się teraz nowe położenie serwera i są w stanie wysłać poczty. Ale co się stanie, gdy podstawowy serwer DNS będzie niedostępny? Brak poczty = niezadowoleni użytkownicy.

Tej sytuacji można uniknąć przez skonfigurowanie transferów stref. Jak pamiętamy z opisu usługi DNS w rozdziale 10., możemy tak skonfigurować tę usługę, aby pliki stref były przesyłane okresowo do serwera wtórnego działającego w innym systemie. Proszę jednak pamiętać, że takie rozwiązanie zwiększa ruch w sieci. To samo dotyczy większości usług, ponieważ wszystkie muszą być od czasu do czasu aktualizowane.

Jednakze w przypadku DNS-u mozemy kontrolowac odstepy czasu pomiedzy transferami strefy. Jesli spodziewamy sie czestych zmian, ustawimy krótszy okres. Jesli zmiany zachodza tylko okazjonalnie, okres pomiedzy transferami moze byc dluzszy. Jesli jednak transfery zachodza tylko co 24 godziny, to po przeniesieniu serwera Mercury pod inny adres IP moga uplynac nawet 24 godziny, zanim zmiana dotrze do serwera wtórnego. Ponadto cala strefa moze wymagac przesyłu, co dodatkowo zwolni proces.

Jak Czytelnik zapewne sie zorientował, transfer strefy DNS nastapi prawdopodobnie wczesniej niz za 24 godziny: wzielismy pod uwage najgorszy scenariusz zalozony przy tworzeniu uslugi DNS. Wielosc uzywanych obecnie uslug DNS pozwala na przyrostowe transfery stref, dzieki którym przesyłanie calego pliku strefy nie jest konieczne. Mozemy tez ustawić bardzo dlugi okres pomiedzy transferami, lecz skonfigurowac powiadomianie, aby zmiany byly propagowane natychmiast. Widac z tego, jak uzywane przez nas protokoly dostosowuja sie do rosnacych rozmiarow sieci. Wielosc sieciowych uslug uwierzytelniania oraz WINS stosuje taka sama logike. Oznacza to, iz te uslugi usiluja zredukowac ruch sieciowy tla.

Jednakze uslugi typu DHCP nie sa równie dobrze skonfigurowane. Ich konstrukcja nie zaklada nawet obecnosci innych, zapasowych serwerów w sieci, wobec czego trudniej jest dodac do sieci kolejny serwer DHCP. Prosze pamietac, ze DHCP sluzy do dzierzawienia adresów ograniczonych do określonej podsieci z puli (zakresu) adresów IP z tej podsieci. Gdybysmy chcieli zapewnic nadmiarowosc, inne serwery DHCP również musialyby posiadac zakres adresów i informacje o tym, które adresy zostaly wydzierzawione lub zaoferowane do wydzierzawienia, a które nie.

Nawet gdyby taki protokol aktualizacji istnial, nadal mozliwe byloby zaoferowanie przez dwa serwery DHCP tego samego adresu w mniej wiecej tym samym czasie. Wymagaloby to protokolu rozwiazujacego konflikty, aby jeden klient mógł wydzierzawic adres, a drugi odmówic. Wyobraźmy sobie skale zamieszania spowodowanego w wielosci biur w chwili, gdy uzytkownicy zalaczaja komputery i wszyscy potwierdzaja adresy!

Wobec tego serwery DHCP po prostu nie „rozmawiaja” ze soba. Aby wiec dodac zapasowy system DHCP, musielibysmy utworzyc dwa zakresy (lub wiecej w przypadku wiekszej liczby serwerów) i przydzielic po jednym zakresie do kazdego serwera. Gdyby zakresy te nakladaly sie, wówczas pojawiłyby sie konflikty adresów IP; wobec tego planowanie zakresów i zapewnienie synchronizacji zakresów i opcji serwera jest bardzo wzazne.

Dla uslug WINS, DNS i uwierzytelniania sieciowego dodawanie kolejnych serwerów powinno byc proste, lecz dla DHCP nie jest. Dla wielosci innych typów serwerów dostepna sa różne metody, których skonfigurowanie pozwoli na koordynacje danych. Czytelnik musi wiedziec, w jaki sposob funkcjonuja serwery w sieci, aby skutecznie planowac nadmiarowosc.

Wracajac do rysunku 20.4, mo zemy zobaczyć, ze dodanie jednego serwera jeszcze nie zapewni nadmiarowosci. W tym przypadku musimy dodac jeszcze przynajmniej dwa serwery — po jednym w kazdej sieci szkieletowej — aby otrzymac przynajmniej podstawowy poziom nadmiarowosci. Dopiero to zapobiegnie zapasci sieci w przypadku awarii pojedynczego serwera lub chocby jego skladnika.

Jednakże nawet wtedy, gdy serwery są nadmiarowe, awaria rutera zawsze wpłynie na jakąś część sieci — albo na trzy podsieci klienckie, albo na połączenie pomiędzy dwiema lokalizacjami. Proszę pamiętać, że ruter też jest usługa sieciowa, która trzeba wziąć pod uwagę podczas planowania nadmiarowości. Oznacza to, że musimy zapewnić zaporowy ruter dla każdego połączenia jednej podsieci z inną, jeśli chcemy uodpornić sieć na wszelkie pojedyncze punkty awarii. W tym przypadku musimy podwoić liczbę routerów lub zastosować routery z wbudowaną nadmiarowością.

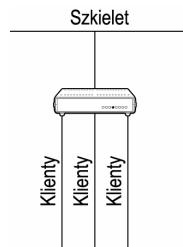
Systemy wieloadresowe

Połączas planowania równowagienia obciążenia jednym z rozwiązań, które możemy wziąć pod uwagę, jest wieloadresowość (*multihoming*), która wymaga użycia więcej niż jednej karty sieciowej w serwerze, aby użytkownicy mogli korzystać z systemu lokalnie, a nie przez ruter. Wieloadresowość jest prostym rozwiązaniem równowagowym obciążenia — i zwiększającą wydajność. Wprowadź ilość pamięci i szybkość procesora oraz dysków ma znaczenie, lecz nadal ważkim argumentem jest karta sieciowa. Wszystkie zadania klientów i wszystkie informacje dostarczane im przez serwer muszą przejść przez ten jeden interfejs.

Problemem może być nie karta sieciowa, lecz wysokie obciążenie danego segmentu. W takim przypadku połączenie dwóch kart sieciowych w jednym systemie do tego samego segmentu może nie sprawić zbytniej różnicy, zwłaszcza w przypadku dostępnego obecnego sprzętu sieciowego. Lecz spójrzmy na rysunek 20.5.

Rysunek 20.5.

Przykład systemu wieloadresowego



Czytelnik może pomyśleć, że na rysunku jest przedstawiony ruter. W rzeczywistości jest to system wieloadresowy. W rozdziale 19. mówiliśmy, że ruter jest wieloadresowym systemem, posiadającym przynajmniej jeden protokół IP. Nic nie przeszkodzi nam w zamontowaniu kilku kart sieciowych w dowolnym systemie. Gdy dodamy więcej kart sieciowych do serwera, będziemy mogli połączyć go bezpośrednio do podsieci zawierającej użytkowników.

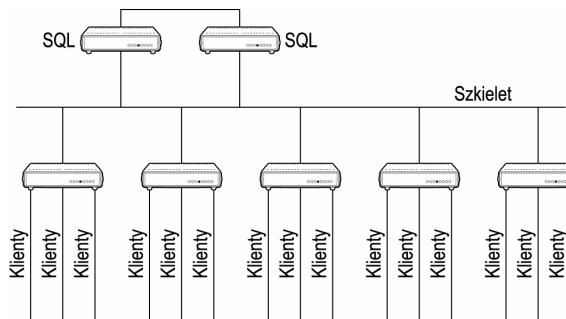
W istocie nic nie przeszkadza w wykorzystaniu do roli routera komputera działającego pod systemem Windows lub Unix. Wiele organizacji używa wewnętrznie komputerów wieloadresowych jako routerów, ponieważ rozwiązanie takie jest często taniej od kupowania wyspecjalizowanych routerów sprzętowych. W najgorszym przypadku wieloadresowy komputer może nam posłużyć jako ruter zapasowy.

Prawdę mówiąc, wieloadresowość możemy zastosować w większości serwerów, w tym w serwerach aplikacji; w ich przypadku technika ta pozwoli znaczco zwiększyć liczbę klientów, która serwer będzie w stanie równoczesnie obsłużyć. Na przykład, rysu-

nek 20.6 przedstawia dwa serwery SQL replikujące pomiędzy sobą dane przez sieć prywatną, jednocześnie obsługujące klienty poprzez sieć główną. Takie rozwiązanie przenosi ruch replikacji z głównej sieci szkieletowej do prywatnego szkieletu SQL.

Rysunek 20.6.

Serwery mogą do komunikacji pomiędzy sobą używać prywatnej sieci szkieletowej



Poniewaz serwery SQL moga aktualizowac wzajemnie swoje dane poprzez czeste replikacje, zmiany moga byc niemal natychmiastowe (jesli nie natychmiastowe), dzieki takim narzedziom SQL, jak replikacje i potwierdzanie dwuetapowe. Wobec tego klienty moga uzywac dowolnego z serwerow SQL, poniewaz sa bliznacze. Polowa klientow moze byc skonfigurowana do korzystania z jednego serwera SQL, a reszta klientow — z drugiego. W razie niedostepnosci jednego serwera, drugi moze przejac jego zadania (poniewaz ma w pelni aktualne dane). Taka konfiguracja zapewnia rownowazenie obciążenia oraz pewna nadmiarowosc.

Odrebny szkielet, jak pomiedzy serwerami z rysunku 20.6, moze byc wydajnie wykorzystany z dowolna usluga, ktora replikuje swoje dane do innego serwera. Wada tego rozwiazania jest wieksza liczba sieci, ktore trzeba nadzorowac, jednak ze sieci zawierajace tylko kilka systemow sa zwykle stabilniejsze od sieci z wieloma klientami.

Na rysunku 20.7 podstawowa struktura z rysunku 20.6 została rozbudowana o wiele dodatkowych klientów i proporcjonalna liczba dodatkowych serwerów. Ponieważ liczba systemów w drugiej sieci szkieletowej jest ograniczona, ruch sieciowy, o który serwery muszą w niej rywalizować jest mniejszy, nawet przy radykalnym wzroście liczby klientów. Kolejna korzyść ze stosowania odrebnego szkieletu jest możliwość używania w nim własnego protokołu serwerów. Na przykład, gdyby serwery SQL z przykładu były produktami Microsoftu, wówczas ich prywatna sieć szkieletowa mogłaby obsługiwać NetBEUI — protokół szybszy w przypadku sieci jednosegmentowej.

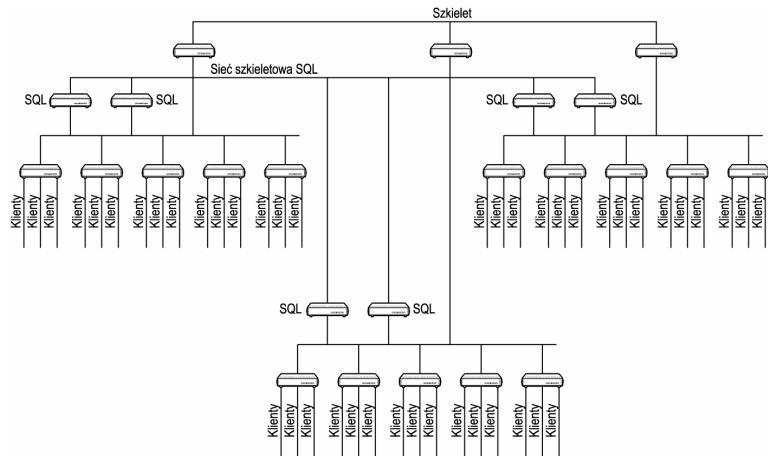
Wieladresosc i wykorzystanie odrenej sieci szkieletowej jest rozwiazaniem dobrze skalowalnym w przypadku uslug typu serwer SQL, ktore moga dokonywac replikacji i rozkladac zapytania na serwery. Nie nadaje sie to jednak dla takich uslug, jak np. DNS lub serwer proxy. W tych przypadkach zmiany dokonywane sa przez uzytkownikow w serwerze centralnym, a nastepnie rozprowadzane do innych serwerow. Nadal jednak moza uzywac hierarchicznych serwerow do redukcji ogólnego ruchu w sieci.

Serwery hierarchiczne

W przypadku serwerów SQL, na rysunkach 20.6 i 20.7, używana była odrewnia sieć szkieletowa, aby polepszyć komunikacje pomiędzy serwerami równorzędnymi. Jednakże w przypadku usługi DNS serwery nie są równorzędne — istnieje jeden serwer pod-

Rysunek 20.7.

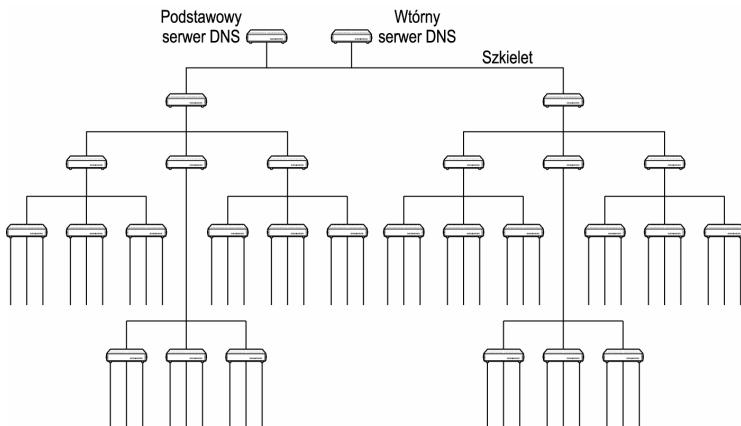
Wraz ze wzrostem rozmiarów sieci, liczba hostów w prywatnej sieci szkieletowej rośnie znacznie wolniej niż w klienckiej sieci szkieletowej



stawowy DNS. Musimy więc znaleźć inne rozwiązanie, aby zredukować ogólny ruch w sieci. Weźmy pod uwagę rysunek 20.8, który przedstawia dwa serwery DNS dla całej sieci. Taka konfiguracja bardzo ułatwia transfery stref; jednakże wszystkie zapytania klientów muszą przejść przez dwa routery, aby dotrzeć do serwera DNS. Nie tylko zwiększa to opóźnienia w rozwiązywaniu nazw, lecz również objętość ruchu sieciowego, który wychodzi z lokalnego segmentu.

Rysunek 20.8.

Podstawowa konfiguracja DNS-u z serwerem podstawowym i wtórnym

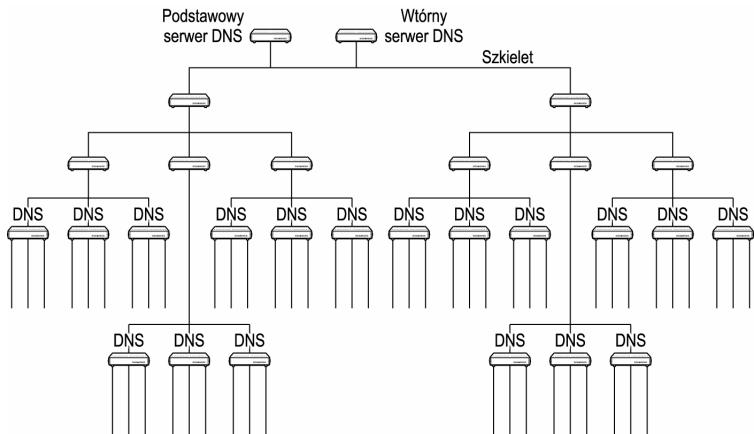


Rysunek 20.9 pokazuje, jak możemy dodać kolejne serwery wtórne w sieci, aby zmniejszyć obciążenie dwóch głównych serwerów i zredukować liczbę segmentów sieci, przez które musi przejść zapytanie. Możemy dodać usługę DNS w pierwszym poziomie routerów, które będą wtórnymi serwerami DNS względem głównego. Zredukuje to do minimum pochodzący od klientów ruch sieciowy, związany z rozwiązywaniem nazw.

Wada rozwiązania z rysunku 20.9 jest konieczność replikacji wszystkich zmian, dokonanych w DNS-ie, do dużej liczby serwerów DNS. W tej sieci będzie przypuszczalnie spora liczba serwerów, co oznacza, że aktualizacje będą krytyczne. Połączenie dużej liczby serwerów wymagających odbioru aktualizacji z ważnością aktualizacji oznacza, że okresy pomiędzy transferami stref muszą być krótkie, co może spowodować niemożliwy do zaakceptowania poziom ruchu sieciowego.

Rysunek 20.9.

Dodanie serwerów wtórnego blizej klientów zmniejszy ruch sieciowy związany z rozwiązywaniem nazw



W tym przypadku mozemy tez skonfigurowac serwery DNS nizszego poziomu jako tylko buforujace, czyli nie przechowujace kopii pliku strefy. W celu zapewnienia mo zliwosci rozwiązywania nazw powinny one byc tak skonfigurowane, by przesyaly zadania dotyczące nieznanych im nazw do serwerów podstawowego i wtórnego na najwyzszy poziomie. Serwery buforujace odpytuja główny serwer DNS o rozwiązywanie nazw, a nastepnie zapisuja lokalnie wyniki w pamieci podrecznej na określony czas, bez koniecznosci transferu całej strefy. Zmniejsza to liczbe serwerów, którymi musimy zarzadzac. Taka konfiguracja sprawdza sie bardzo dobrze przy zalozeniu, iz użytkownicy korzystajacy z określonego serwera DNS wykonuja w pracy te same podstawowe zadania. W takim przypadku te same nazwy serwerów bylyby rozwiązywane raz za razem, zazwyczaj z lokalnej pamieci podrecznej. Zastosowanie w tym przypadku serwerów buforujacych jest dobrym rozwiązaniem, zwlaszcza jesli czas zycia (TTL) rekordów DNS jest wystarczajaco dlugi.

Uwaga

Jesli na potrzeby powyzszego rozwiązania ustawimy wysokie wartosci TTL, musimy pamietac, by zmniejszyc TTL przed modyfikacją wpisów w DNS-ie. Na przykład, jesli zamierzamy przeniesc serwer pocztowy, a obecny TTL dla rekordów wynosi 8 godzin, mozemy na okolo 8 godzin przed przenosinami zmniejszyc TTL do 15 minut. W niektórych systemach mozna ustawiac TTL indywidualnie dla rekordów, co umozliwia modyfikacje tego parametru dla jednego serwera.

Jesli zamierzamy zastosowac lokalny serwer buforujacy skonfigurowany tak, by przesyjal zadania „w góre” do głównego serwera DNS, mozemy tez rozważyć wykorzystanie serwera podporządkowanego (*slave*). Zapobiegnie to próbom przekazywania przez serwer DNS zapytan o nazwy do Internetu lub skonfigurowanych dla danego serwera serwerów poziomu głównego. Mozemy również skonfigurowac dla lokalnych serwerów DNS ich serwer podstawowy jako strefę główną.

Niezależnie od decyzji, jak skonfigurowac lokalne serwery (jako buforujace lub wtóre), mozemy uzyskac nadmiarowosc, dodajac po prostu w klientach (za pomoca DHCP) wpis drugiego serwera DNS, wskazujacy na serwer główny. Konfiguracja taka dobrze sprawdza sie w przypadku serwerów DNS, WINS i proxy. W przypadku serwera WINS wymagany jest dodatkowy krok, w którym lokalny serwer WINS wypycha zmiany do jednego z głównych serwerów WINS, natomiast główny serwer WINS sciaga zmiany z serwera lokalnego. To pozwoli zaimplementowac jednokierunkowa replikacje pomiedzy lokalnymi serwerami WINS i serwerem głównym.

Jako metode równowazenia obciazenia wywolanego zadaniami uzytkownikow i zapewnienia nadmiarowosci mozemy stosowac dodatkowe serwery, wieloadresowosc i serwery hierarchiczne — albo polaczenie wszystkich trzech metod. Kazda z nich sie nadaje, lecz wszystkie opieraja sie na zasadzie rozkladu obciazenia na wiecej serwerow. Dobra wiadomosc: wiekszosc uslug, ktore musimy udostepnic w sieci, pozwala na takie rozwiazania. W przypadkach, gdy to niemozliwe (na przyklad, dla niektórych baz danych i systemow poczty elektroniczej), musimy znalezc inne rozwiazanie — na przyklad klastry.

Stosowanie grupowania

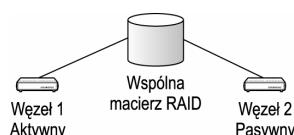
Jesli nasza firma ma pieniadze do wydania i dane, których nie moze utracic w serwerze mogacym ulec awarii, rozwiazaniem jest grupowanie. Wezmy pod uwage duzy, ogólnokrajowy dom handlowy, który pozwala klientom skladac zamówienia telefonicznie. Posiada on centrum telefoniczne, które odbiera telefony z calego kraju i przyjmuje co godzine tysiące zamówien. Kazde zamówienie obejmuje setki drobnych porcji informacji. W takim przypadku awaria serwera bedzie kosztowac firme dosłownie setki tysiecy złotych. Znalezienie, zbudowanie, zapelnienie i odtworzenie innego systemu po awarii moze oznaczac utratę godzin lub dni.

Czasami nieobecnosć serwera po prostu nie wchodzi w rachube, zas kilka serwerów, które musza replikowac medzy soba dane, moze nie nadazac za przeplywem danych. W takim przypadku jedynym rozwiazaniem jest grupowanie (inaczej klastrowanie — *clustering*).

Grupowanie w najprostszej postaci obejmuje dwa systemy ze wspólnym systemem dyskowym. Kazdy z systemów regularnie sprawdza stan drugiego, a jesli podstawowy system zawiedzie, drugi uzywa informacji na wspólnych dyskach, by podjac jego zadania. Rysunek 20.10 przedstawia taka prostą formę grupowania.

Rysunek 20.10.

Podstawowy przykład grupowania aktywny-pasywny



Rysunek 20.10 jest przykładem grupowania aktywny-pasywny. Drugi system nie robi nic, dopóki podstawowy wezel nie zawiedzie. W takim przypadku wezel zapasowy przejmuje funkcje podstawowego. Taki proces moze zadzialac tylko wtedy, gdy wszystkie dane dla przejmowanych uslug znajdują sie w zestawie wspoluzytkowanych dysków. Ponadto zestaw dysków musi byc nadmiarowy, gdyz w przeciwnym razie awaria napędu spowodowałaby zalamanie calego klastra.

Macierz dyskowa jest podzielona na różne obszary. W jednym z nich kazdy system utrzymuje informacje o bieżacych polaczeniach. Drugi system, sledzac informacje o polaczeniach na dysku i przechowujac je w pamieci, w przypadku przejecia funkcji systemu uszkodzonego, moze uzyskac od niego te informacje. Kazda usluga posiada również własny obszar, tak by mogla funkcjonowac w dowolnym systemie.

Kazdy wezel w klastrze musi miec zaladowane wszystkie uslugi, ktore ma przejac, zas konfiguracja uslug w obu wezlach musi byc synchronizowana. Sama usluga potrzebuje odrebnego adresu IP, niezaleznego od systemu, by uzytkownicy mogli laczyc sie z nia niezalezsnie od tego, ktory system aktualnie udostepnia usluge.

Wprawdzie takie grupowanie zapewnia nadmiarowosc, lecz nie udostepnia rownowazenia obciazenia w zadnej postaci. Aby zapewnic rownowazenie obciazenia, musimy skonfigurowac system aktywny-aktywny, w którym obo systemy beda w stanie równoczesnie korzystac z zasobów dyskowych. Aby taka konfiguracja mogla dzialac, klaszter musi obslugiwac jakis mechanizm blokowania, aby dane modyfikowane w jednym systemie nie mogly zostac równoczesnie zmodyfikowane przez inny system. Blokowanie pozwala jednemu systemowi zablokowac czesc lub wszystkie zasoby, aby nie zaszla rywalizacja o zasob — próba równoczesnego dostepu do tego samego zasobu przez dwa procesy. Oznacza to, ze aplikacja i usluga grupowania musza wspolpracowac ze soba i byc specjalnie do tego zaprojektowane. W sytuacjach, gdy usluga grupowana w systemie aktywny-aktywny jest cos w stylu prostego serwera WWW, zadanie to jest o wiele łatwiejsze, poniewaz klienty nie aktualizuja zadnych danych.

Do dystrybucji klientow pomiedzy aktywne wezly klastra potrzebna jest dodatkowa usluga. Moze do tego posluzyc karuzela rekordow w DNS-ie lub inna usluga, na przyklad Network Load Balancing Microsoftu. W srodowisku TCP/IP karuzela (*round robin*) w DNS-ie jest rozwiazaniem prostym i latwym do zaimplementowania. Metoda ta polega na wprowadzeniu do pliku strefy dwóch lub wiecej rekordow hosta o tej samej nazwie, lecz z różnymi adresami IP. Serwer DNS wydaje te adresy na przemian, jeden po drugim, do nastepujacych po sobie odebranych zadan. W niektórych implementacjach serwer DNS również bierze pod uwage adres IP klienta — zwracajac adres IP lokalnego interfejsu, jesli takim dysponuje.

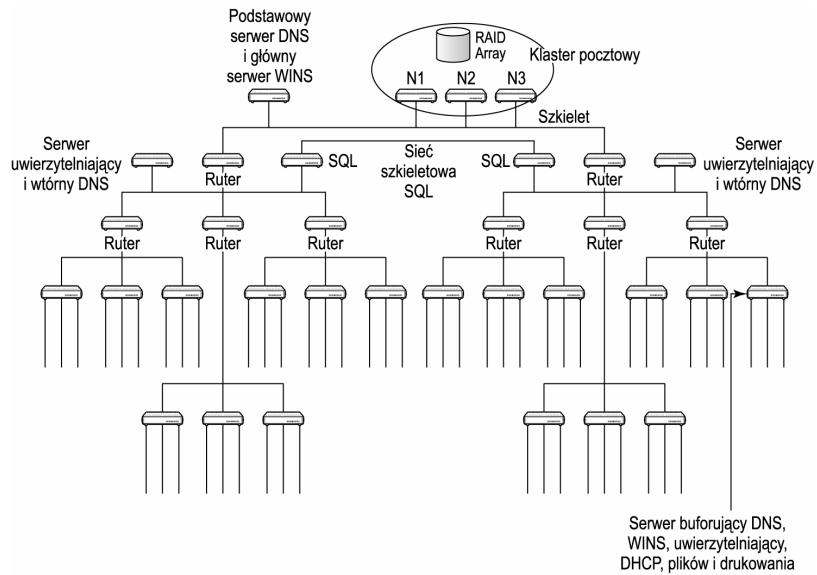
Grupowanie moze wykraczac poza dwa systemy i rozrosnac sie do znacznie wiekszej liczby wezlow. Wraz ze wzrostem tej liczby rośnie złożoność klastra i procesu planowania. Grupowanie jest rozwiazaniem kosztownym, które zwykle stosuje się w przypadku serwerów niezbednych do funkcjonowania przedsiębiorstwa, lub w sytuacjach, gdy nie mamy innego wyboru, poniewaz usluga nie moze pracowac w trybie rozproszonym.

Jest kilka sposobów na osiągniecie rownowazenia obciazenia i nadmiarowosci. Koszty i złożoność tych metod mogą różnić się w bardzo szerokim zakresie, lecz dla najważniejszych usług sieciowych musimy wziąć pod uwagę jakiś mechanizm nadmiarowości i rownowazenia obciazenia. Rysunek 20.11 przedstawia kompletna sieć z tymi mechanizmami.

W tym przypadku usługa pocztowa uruchomiona jest w klastrze złożonym z trzech serwerów, zas do rozkładu obciążenia pomiędzy te serwery służą karuzela DNS. Podstawowy serwer DNS jest połączony do sieci szkieletowej i skonfigurowany do rozsypania zmian do serwerów wtórnego. Z kolei routery polozone najbliżej klientów, będące dodatkowo serwerami buforującymi DNS, korzystają z dwóch serwerów wtórnego jako forwarderów. Serwery SQL replikują informacje pomiędzy sobą, zas klienci są tak skonfigurowane, byłączyć się z najbliższym serwerem SQL. Kazdy ruter lokalny zapewnia dodatkowo uwierzytelnienie sieciowe i replikuje informacje do głównego serwera uwierzytelniającego dla danej części sieci. Ten z kolei replikuje wszystkie swoje informacje do innych serwerów uwierzytelniających. Na potrzeby używanego w sieci systemu uwierzytelniania mogą być skonfigurowane dla klientów również serwery alternatywne.

Rysunek 20.11.

*Kompletna siec
z zapewnionym
równowagiem
obciążenia
i nadmiarowością*



Klienci, które muszą tego dokonać, rejestrują się w lokalnym serwerze WINS, uruchomionym w lokalnym ruterze, zas ten serwer replikuje rejestracje klientów do głównego serwera WINS. Klienci posiadają lokalne serwery DNS i WINS skonfigurowane jako pierwsze na liście oraz główne serwery nazw jako drugie. W lokalnych sieciach konfiguracja ta jest rozprowadzana za pomocą DHCP. Serwery DHCP otrzymują małe porcje adresów dla dwóch pozostałych najbliższych sieci, zas routery lokalne są skonfigurowane do przekazywania pakietów BOOTP.

Jedyna czescia sieci nie oferujaca nadmiarowosci sa rutery i serwery plików i drukowania. Nadmiarowosc ruterow mozemy osiągnac za pomocą klastrów, lecz z uwagi na koszty zwykle takie rozwiązanie nie jest stosowane. Ponieważ jednak koszty klastrów maleja, niektóre organizacje mogą rozważać nadmiarowość na tym poziomie.

Rozdzial 21.

Wprowadzenie do lacznosci

W tym rozdziale:

- ◆ Laczenie lokalizacji
- ◆ Budowanie sieci WAN
- ◆ Wybór strategii dostepu telefonicznego
- ◆ Praca na odleglosc

Transmisja danych oznacza elektroniczne przesyłanie danych przez nosnik fizyczny. Mozemy rozpoczac transmisje po przeprowadzeniu tak prostej operacji, jak np. przeprowadzenie swiatlowodu pomiedzy dwoma pietrami budynku, moze tez wymagac utworzenia duzej infrastruktury o ogólnoswiatowym zasiegu.

Jesli lacznosc, z ktorej korzystamy, ma zasieg ogólnoswiatowy, przypuszczalnie nie bedziemy przeciągac własnych swiatlowodów po dnie oceanu. Zamiast tego skorzystamy z uslug operatora, który już dysponuje takim swiatlowodem (lub innym sposobem na przesyłanie danych na duza odleglosc). W tym przypadku tak naprawde nie musimy znac szczegolów funkcjonowania calej sieci.

Czytelnik zna już zapewne terminy LAN i WAN, lecz przedstawimy tu ich definicje:

- ◆ *LAN (local area network — siec lokalna)* — wydzielona, szybka siec, która moze skladac sie z wielu segmentow; jednakże wszystkie te segmenty znajdują sie w jednej lokalizacji.
- ◆ *CAN (campus area network — siec osrodka)* — siec lacząca ze sobą sieci LAN (na przykład rozrzucone po kilku budynkach). Lacze komunikacyjne sluzą do kierowania danych z jednego budynku do innego. Taki typ sieci zwykle obejmuje tylko lacza o krótkim zasięgu — na przykład uczelnia moze posiadać w jednym miejscu kilka budynków i laczyć je swiatlowodami.
- ◆ *MAN (metropolitan area network — siec miejska)* — siec o zasięgu większym od CAN i zazwyczaj korzystająca z usług jakiegos dostawcy. Ogólnie mówiąc, sieci MAN sa ograniczone w zasięgu do jednego miasta, czyli sa mniejsze od sieci WAN.
- ◆ *WAN (wide area network — siec rozległa)* — siec o dużym zasięgu geograficznym, na przykład pomiędzy miastami lub kontynentami. Siec WAN moze na przykład laczyć USA, Kanadę i Wielką Brytanię. Ogólnie mówiąc, każda siec o zasięgu wykraczającym poza jedno miasto jest uznawana za siec rozległą.

Wprawdzie od czasu do czasu Czytelnik spotka terminy CAN lub MAN, lecz w praktyce termin LAN odnosi sie do sieci w granicach budynku, zas WAN do wszelkich sieci posiadajacych polaczenia na zewnatrz budynku. Pozostala czesc niniejszego rozdzialu bedzie dotyczyla lacznosci, rozwoju technologii, ktore umozliwiaja te lacznosc, oraz decyzji, ktore trzeba podjac implementujac te technologie.

Podstawy lacznosci

Cala minniejsza ksiazka jest poswiecona lacznosci. Omowilismy, jak system uzytkownika opakowuje dane w segment TCP i przesypla dane do warstwy IP. Rzonalismy, jak IP wykorzystuje protokol ARP do znalezienia nastepnego hosta, a nastepnie przekazuje dane do kabla, ktrzym zostana przeslane do zdalnego hosta lub karty sieciowej rutera.

Lecz co sie dzieje, gdy wychodzimy z wlasnej sieci? Dokad dane udaja sie po opuszczeniu zwyklej sieci Ethernet lub Token Ring? Czy caly Internet jest tak naprawde jedynie ogromna siecia Ethernet?

Aby zrozumiec lacznosc, musimy cofnac sie o kilka lat — do roku 1878, w którym zostaly zainstalowane pierwsze sieci. Nie, to nie jest blad w druku. System telefonii byl pierwsza siecia uzywajaca komutacji — dokladniej mówiac, komutacji obwodów. Za pomoca przelacznic recznych lub automatycznych firma telefoniczna udostepnila polaczenie dwupunktowe pomiedzy telefonami rozmówców.

Laczenie lokalizacji

System telefonii byl pierwszym, uzywajacym istniejacej techniki do zapewnienia lacznosci pomiedzy odleglymi lokalizacjami. Poczatkowo lacznosc byla uzyskiwana za pomoca mechanicznych przelacznikow, tworzacych obwod elektryczny pomiedzy dwiema lokalizacjami. Dzis nadal uzywamy podobnych kabli do laczenia uzytkownikow z siecia LAN.

Oczywiscie potrzebny byl jakis rodzaj przelacznic; w przeciwnym razie niezbedne byloby polaczenie kablami wszystkich telefonow na zasadzie kazdy z kazdym (podniesienie sluchawki laczyloby uzytkownika z wszystkimi pozostalymi telefonami na swiecie. Wyobrazmy sobie ten zgielk w sluchawce!)

Technologia laczeniowa

Pierwsze lacznice byly prostymi urzadzeniami obslugiwanyimi przez operatora, który wkladal wtyczke do gniazda w lacznicy, aby polaczyc dwa telefony. Zasadniczo telefony na okres rozmowy byly laczone fizycznie, a polaczenie to bylo ciagle tylko pomiedzy dwoma telefonami — inaczej mówiac, bylo tymczasowe, ciagle i wylaczne.

Komutacja obwodów sprawdza sie bardzo dobrze w komunikacji glosowej, poniewaz przesywanie glosu odbywa sie w czasie rzeczywistym. Na dlugich dystansach oznacza to jednak koniecznosc posiadania wielu kabli, aby kazdy telefon posiadal dwa przewody wymagane do polaczenia. Oznacza to olbrzymie zuzycie kabla i do dzis zwiększa koszty rozmów miedzyniestowowych (poniewaz koszty central i okablowania musza sie

zwrócić). Do transmisji danych wymagającej rzadko połączeń obwody musiały pozostawać otwarte, co było bardzo kosztowne. Aby komunikacja mogła prosperować, trzeba było opracować inne metody.

Komutacja pakietów

Przyglądając się danym, które komputer generuje i wysyła przez sieć, zauważymy dwie różnice w stosunku do transmisji głosu. Pierwsza różnica jest fakt, iż w transmisji danych dopuszczalny jest *pewienny* poziom opóźnień, ponieważ odbiorca może zaczekać na dane — w transmisji głosu powodowałoby to rwanie lub niezrozumiałosc rozmowy.

Druga różnica jest postać danych — transmisja głosu używa ciąglej fali sinusoidalnej, natomiast dane komputerowe składają się z bitów zgrupowanych w bajty (o długości pięciu, siedmiu lub osmu bitów). Oznacza to, że grupa bajtów może zostać dalej zapakowana w zaadresowany pakiet. W rzeczywistości transmisje danych (również potoki bajtów) są często dzielone na osobne jednostki.

W latach 60. inżynierowie zdecydowali, by traktować dane jako ciąg pakietów, a nie potok informacji. W ten sposób zlikwidowali konieczność stosowania *ciągłych* połączeń, otwierając drogi dla transmisji danych. Obecnie, gdy dane są wysłane, mogą dojść do urządzenia na granicy sieci, które podzieli je na porcje o odpowiednich rozmiarach, zaadresuje i wysła w sieć. Ten proces można przyporównać do działania referatu pocztowego firmy.

TCP/IP jest przykładem sieci z komutacją pakietów. Przelacznik na logicznym skraju każdego segmentu sieci dane przeznaczone dla innych segmentów sieciowych dzieli na pakiety (datagramy) o rozmiarach odpowiednich dla topologii następnej sieci oraz wysyła je do kolejnego przełącznika. Ten sprawdza datagram i przesyła do kolejnego przełącznika, aż pakiet dotrze do ostatecznego miejsca przeznaczenia. Jak możemy się domyślić, omawianymi przełącznikami są tak naprawdę routery. W rzeczywistości routery zostały jako pierwsze urządzenia wykorzystane do utworzenia sieci szkieletowej ARPANet — pierwszego wcielenia Internetu.

Problem z komutacją pakietów polega na tym, że wraz ze wzrostem ruchu w sieci ruter musi pracować coraz częściej, aby nadążyć. W pewnym momencie ruter „zatka” się i zacznie odsyłać lub tracić datagramy. W końcu ulegna temu wszystkie routery, powodując zatrzymanie sieci. Razem z nowymi danymi zaczyna pojawiać się retransmisje, dalej pogarszające problem.

Taka sytuacja może prawie uniemożliwić przesyłanie danych. Większa część problemu powodowana jest przez naturę komutacji pakietów. Nie istnieją stałe trasy, opisujące trasy danych przez sieci. Pakiety mogą podróżować od A do B różnymi drogami, w miarze zmian warunków w sieci, co prowadzi do odbierania pakietów w niewłaściwej kolejności lub ich odrzucenia po upływie limitu czasu. Do rozwiązania problemu sieci z komutacją pakietów potrzebne były podstawowe połączenia komutacji pakietów, lecz bez konieczności tworzenia ciągłych połączeń. Tu w grę wchodzi technika komutacji (przelaczania) ramek — *frame relay*.

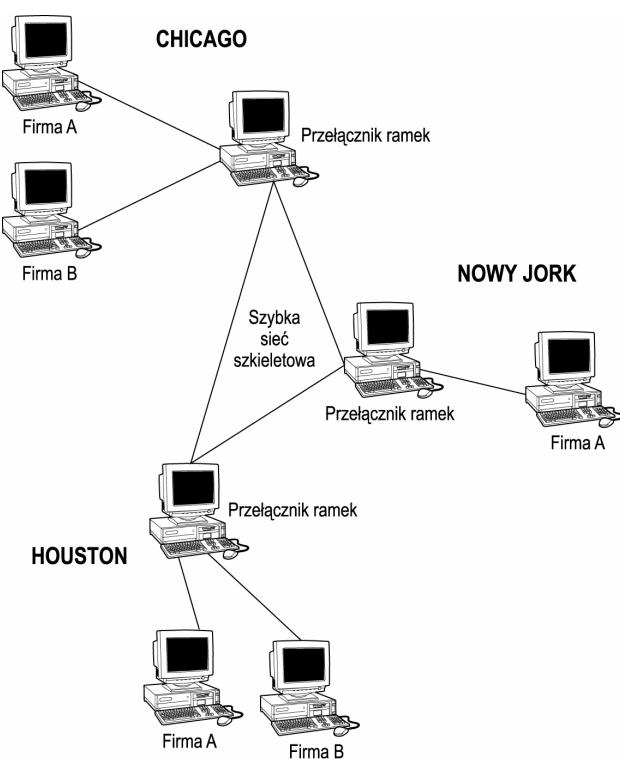
Przelaczanie ramek

W technice *frame relay* dla każdej transmisji — jak dla rozmowy telefonicznej — dworzony jest pełny obwód. Wirtualny obwód prywatny może zostać zaprogramowany

w przelaczniku, zas gdy potrzebna bedzie laczosc, utworzony zostanie w locie komutowany obwod wirtualny (*Switched Virtual Circuit*). Jednakze w przeciwienstwie do przelacznikow obwodów, polaczenia pomiedzy przelacznikami moga byc wspoluzytowane przez dane kilku róznych klientów. Przelaczanie ramek pozwala również tworzyc stale obwody wirtualne (*Permanent Virtual Circuit*).

Aby lepiej zrozumiec przelaczanie ramek, spójrzmy na rysunek 21.1. Przedstawia on trzy lokalizacje — Houston, Chicago i Nowy Jork — posiadajace przelaczniki pakietów, podlaczone do szybkiej sieci szkieletowej. W tym przypadku dostawca uslug posiada fizyczne polaczenie z Houston do Chicago, z Houston do Nowego Jorku, jak również pomiedzy Chicago i Nowym Jorkiem.

Rysunek 21.1.
Przyklad pozwalajacy
zrozumiec technike
frame relay



Zalozmy teraz, ze firma B placi za przenoszenie danych pomiedzy Chicago i Houston. W tym przypadku dostawca uslug tworzy SVC pomiedzy tymi dwoma miastami. Obwód mówi przelacznikowi, iz wszelkie dane pochodzące ze sprzetu klienta w jednym mieście powinny przechodzić do przelacznika w drugim mieście, aby mogły zostać dostarczone do sprzetu w lokalizacji klienta w tym mieście. Firma A posiada oddziały w każdym z miast, wiec dla niej przelacznik w każdym mieście posiada dwa komutowane obwody wirtualne do urządzeń w pozostałych miastach.

Dane firm A i B, przesypane pomiedzy Chicago i Houston, będą przechodzić po tym samym fizycznym nosziku laczacym oba miasta. Przelacznik zapewni dotarcie właściwych danych do właściwych klientów. Oczywiście w razie przeciazenia linii wszelki ruch pomiedzy tymi dwiema lokalizacjami zostanie spowolniony. To nadal jest proble-

mem w przelaczaniu ramek. Problemy z opóźnieniami nie mają zbyt dużego wpływu na transmisje danych, lecz praktycznie uniemożliwiają przesyłanie nie skompresowanych danych audio i wideo. Na dodatek nie ma sposobu, by zidentyfikować dane krytyczne.

Przelaczanie komórek

Przelaczanie komórek stanowi bardziej radykalne podejście do problemu przenoszenia danych próbując utworzyć jedną technologię, mogącą przesyłać głos, dane, obraz i tak dalej. Ponieważ firmy telefoniczne używają swoich łączów fizycznych do przesyłania zarówno głosu, jak i danych, zdolność do efektywnego współuzyskiwania posiadanych łączów stanowi dla nich olbrzymi problem.

W komutacji pakietów lub ramek przesyłane dane miały zmienne długość. Jednakże większość systemów komutacji pakietów i ramek jest obecnie skonfigurowana do przesyłania pakietów lub ramek, których rozmiar opiera się na standardzie Ethernet, stanowiącym aktualnie dominującą topografię sieciową. Jedna z podstawowych zmian w komunikacji komórek jest przesyłanie małych pakietów o stałej długości — *komórek*. Rozmiary ramek mogą być duże (do 4096 oktetów), natomiast komórki zawsze posiadają 53 oktetów (48 oktetów danych i 5 oktetów informacji towarzyszących).

Stale rozmiary komórek oznaczają, że przełączniki nie muszą już ustalać rozmiarów pakietów, zanim zaczyna dokonywać innych operacji na danych. Nie jest stosowana fragmentacja i ponowne składanie, bufore mogą pomieścić dużą liczbę komórek, zasada logiki potrzebna do obsługi małych komórek można zaimplementować sprzętowo na poziomie układów scalonych.

Umieścienie logiki w układzie scalonym zwiększa jednak koszt kart. Jest to jeden z głównych powodów, poza rozwojem gigabitowego Ethernetu, dla których implementacje typu ATM (*Asynchronous Transfer Mode*) nie torują sobie drogi na biurka. A ponieważ każdym 48 oktetom danych towarzyszy 5 oktetów dodatkowych, stosowanie komunikacji komórek niesie za sobą bardzo wysoki ruch sieciowy tła.

Podobnie jak w komunikacji pakietów, w komunikacji komórek tworzony jest wirtualny obwód w użytkowym wspólnie połączeniu. Ponadto komunikacja pakietów może zapewnić gwarantowaną jakość usług (*Quality of Service*), co jest bardzo przydatne dla przesyłania danych czasu rzeczywistego, na przykład dźwięku i wideo, tymi samymi zespołami współpracującymi łączami, które służą do przesyłania danych.

W wielu przypadkach organizacja nie jest w stanie położyć fizycznego kablowania, używanego do szybkiej komunikacji na skalę światowej. Wobec tego musimy podłączyć się do sieci szkieletowej lub szkieletu publicznego (czyli Internetu), co oznacza, że potrzebne jest połączenie od naszej lokalizacji do większych sieci. W przypadku Internetu możemy wykorzystać do tego celu lokalnego ISP, zasługując na pomocne łączę dzierżawione — większego dostawcę lub lokalną firmę telefoniczną.

Połączenie przez lokalnego ISP

Połączenie się z lokalnym dostawcą usług jest prostą metodą uzyskania dostępu do Internetu; jednakże lokalny ISP zazwyczaj nie posiada połączeń dwupunktowych lub łączów dzierżawionych. Lokalny ISP jest z kolei połączony z większym operatorem, który za-

pewnia dostep ISP do Internetu — ISP w zasadzie odsprzedaje pasmo laczna, wobec czego mozemy oczekiwac od niego dodatkowych uslug.

Przy podejmowaniu decyzji o wyborze lokalnego ISP, musimy rozważyć wiele czynników. Przede wszystkim zapytajmy siebie samych, czy określony ISP utrzyma się na rynku przez następne szesc miesięcy. Wielu małych dostawców usług internetowych pojawiło się i znikło w przeciagu kilku ostatnich lat. W niektórych przypadkach znikli szybko, zostawiając klientów na łodzie. Proszę nie zapominac, że dostajemy to, za co placimy. Jesli umowa jest zbyt piekna, by mogła być prawdziwa, zazwyczaj nie jest. Jesli połaczenie ma jedynie służyć użytkownikom do surfowania w Internecie, jest to nieważne. Jesli jednak połaczenie to ma być łączem do świata zewnętrznego dla poczty elektronicznej i służyć do łączenia biur za pomocą tunelowania, to lepiej nie skapic na usłudze.

Warto jeszcze upewnić się, czy ISP stosuje rozsądne proporcje w wielkościach sprzedanego pasma do rzeczywistego — podobnie jak towarzystwa lotnicze, dostawcy usług internetowych sprzedają więcej usług niż posiadają. Zazwyczaj uchodzi to na sucho, ponieważ nie wszyscy użytkownicy są równoczesnie online. Jesli jednak ISP sprzedal o wiele za dużo usług, dostępne pasmo może na tym ucierpiec. Ponadto możemy przyznać się ogłoszeniom ISP o zatrudnieniu, aby zorientować się, jak dużo placów swoim pracownikom, a więc na jakim poziomie jest jego personel. Zwłaszcza lokalni dostawcy usług internetowych mogą znacznie się różnić pod względem dosiadczania i profesjonalizmu obsługi.

Na koniec zidentyfikujmy rodzaj połaczenia ISP z jego dostawca. Jesli korzysta z łączna T1, może nie być w stanie obsłużyć zbyt wielu użytkowników; jesli jednak posiada kilka łącz T3 lub OC12 do więcej niż jednego dostawcy, to możemy być spokojni, że ISP będzie w stanie świadczyć nam usługi dobrej jakości.

Glowni dostawcy uslug internetowych i operatorzy telefonii

Jesli chcemy mieć zagwarantowaną przepustowość lub naprawdę potrzebujemy łączności dwupunktowej bez stosowania protokołów typu L2TP lub PPTP, powinnismy poszukać dużego dostawcy usług internetowych lub operatora telefonii. W większości przypadków operatorzy telefonii są dużymi ISP, ponieważ firmy telefoniczne miały czas na rozciągnięcie okablowania w całym kraju. Operatorzy telewizji kablowej wkrzyli do walki ze swoimi modemami kablowymi, co pomogło utrzymać względna uczciwość firm telefonicznych.

Korzystanie z usług dużego przedsiębiorstwa telefonicznego ma kilka zalet. Po pierwsze, firmy takie zwykle zainwestowały spore sumy w technologie, pracowników i zakłady, więc najczęściej potrafią świadczyć najlepsze usługi. Po drugie, posiadają możliwości techniczne udostępniania połączeń dwupunktowych za pomocą łącz dedykowanych, komutacji ramek i komórek. Maja też zwykle dość funduszy, by zainwestować w całkowicie nadmiarowe systemy, więc zwykle brak zasilania lub zerwanie łączna nie powinno mieć wpływu na połaczenie. Słowa *zwykle* uzylem nie bez powodu — w niektórych regionach świata lokalni ISP posiadają lepsze połączenia i lepszą infrastrukturę od kompanii telefonicznych. Oplaca się więc przeanalizować oferty.

Po stronie minusów trzeba zanotować, że firmy telefoniczne zwykle są droższe od lokalnych ISP. Musimy zdecydować, czy zdolność do komunikacji dwupunktowej jest naprawdę warta dodatkowych kosztów.

Dopasowanie elementów składanki

Czytelnicy mogą zastanawiać się, jak te wszystkie elementy — ISP, operatorzy telefonii, komutacja komórek, komutacja obwodów itp. pasują do siebie, i jak naprawdę wygląda współpraca pomiędzy lokalnym ISP, firmami telefonicznymi i firmami korzystającymi z usług. Wszystkie elementy omówione w tym rozdziale tworzą hierarchię, na dole której znajduje się użytkownik. Możemy łączyć się z lokalnym ISP, który zwykle łączy się z lokalną siecią szkieletową kompanii telefonicznej za pomocą routera brzegowego. Odtąd pakiety są przesyłane do lokalnego szkieletu operatora telefonii, a następnie trasowane do docelowego miasta, w którym schodzą do routera brzegowego, a następnie do docelowego serwera.

Na rysunku 21.2 została przedstawiona ta hierarchia. Próbujemy połączyć się z komputerem biurowego (w Vancouver w Kolumbii Brytyjskiej) z serwerem WWW (St. John's w Nowej Fundlandii). Jak widać na rysunku, zadanie wysłane z systemu użytkownika do lokalnego routera przechodzi do lokalnego ISP. Łączy się z nimi nasza sieć a ISP jest dwupunktowe. U ISP znajduje się wiele routerów brzegowych, a dane naszej firmy przechodzą przez jeden z nich. ISP przesyła następnie dane do swojej sieci szkieletowej i dalej do kompanii telefonicznej. Ponownie ruch przechodzi dalej do routera brzegowego, a następnie do centralnego obszaru drugiej lokalizacji.

Z lokalnego biura Telco dane przesyłane są przez sieć szkieletową ATM do odległego osiedla Telco, gdzie odbywa się odwrotny proces. Dane przechodzą przez router brzegowy do głównego połączenia lokalnego ISP. Stąd połączenie przechodzi do routera brzegowego ISP i głównego routera w sieci docelowej. Wreszcie jesteśmy w odległej sieci i łączymy się ze zdalnym systemem.

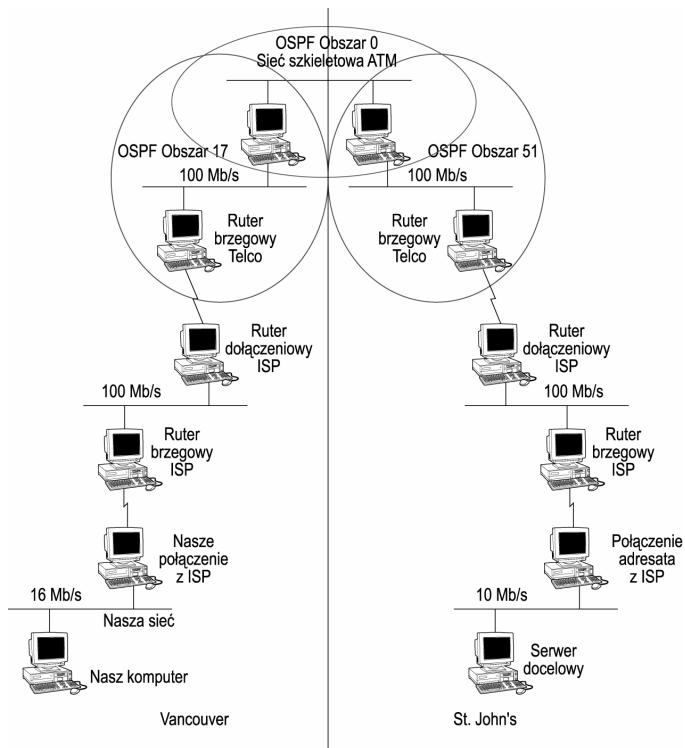
W przykładzie z rysunku trasa jest raczej krótka. Weźmy teraz pod uwagę, jak sytuacja wyglądałaby, gdyby nie było bezpośredniego połączenia pomiędzy Vancouver i St. John's (spoglądając na mapę zdajemy sobie sprawę, że takiego nie istnieje). Pakiet musiałby powieść się np. do Toronto i ewentualnie Halifaxu, przedłużając ścieżkę o kolejne hopy. I oczywiście zakładamy, że dwa lokalni ISP korzystają z usług tego samego dostawcy w sieci szkieletowej — chociaż taka sytuacja jest raczej wyjątkiem, a nie normą. Poniższy listing przedstawia trasę z systemu w Montrealu do systemu w Winnipeg. W tym przypadku obie lokalizacje przyłączone są do sieci Belli.

```

1  10 ms  <10 ms  <10 ms  207.236.145.33
2  *       <10 ms    10 ms  10.30.235.1
3  120 ms   91 ms    90 ms  mtlcorr01-fe0-0-0.in.bellnexxia.net
[206.108.105.129]
4  <10 ms    10 ms    10 ms  core1-montreal02-pos11-0.in.bellnexxia.net
[206.108.97.145]
```

Rysunek 21.2.

*Pelna sciezka
polaczenia pomiedzy
komputerem
uzytkownika
i docelowym serwerem*



```

5   10 ms   20 ms   20 ms   core2-toronto63-pos2-0.in.bellnexxia.net
[206.108.107.185]
6   10 ms   20 ms   20 ms   core2-toronto63-pos7-0.in.bellnexxia.net
[206.108.107.153]
7   40 ms   40 ms   40 ms   core1-winnipeg32-pos6-1.in.bellnexxia.net
[206.108.98.86]
8   40 ms   40 ms   50 ms   dis4-winnipeg32-pos10-0.in.bellnexxia.net
[206.108.102.94]
9   40 ms   40 ms   50 ms   mts-gw.dis1-winnipeg32-atm6-1-0-
1.in.bellnexxia.net [206.108.110.6]
10  40 ms   50 ms   40 ms   wmpgbr27-v102.mts.net [205.200.28.89]
11  41 ms   50 ms   50 ms   WEBHOUSE [205.200.252.82]

```

W tym przypadku pierwsze kilka hopów przenosi nas szybko do routera brzegowego `mtlcorr01-fe0-0-0`. Ponieważ Bell jest dostawcą usługi, połączenie nie przechodzi przez żadnego lokalnego ISP. Z lokalnego biura Bella dane podróżują do routera wejściowego w głównej sieci Toronto i wychodzą znów, przechodząc do routera w Winnipeg. Tam dane przechodzą do routera dystrybucyjnego, przez `mts-gw.dis1-winnipeg32-atm6-1-0-1` do lokalnego ISP (w tym przypadku lokalnej firmy telefonicznej). Z niej istnieje połączenie do systemu lokalnego o nazwie *WEBHOUSE*.

Następna trasa (pokazana za pomocą narzędzia `tracert` w systemie Windows lub `traceroute` w systemach uniksowych) pokazuje sytuację, w której zaangażowanych jest wiele firm telefonicznych.

```

1  <10 ms   10 ms   <10 ms   207.236.145.33
2  <10 ms   10 ms   10 ms   10.30.235.1

```

```
3 <10 ms 10 ms 10 ms mtlcorr01-fe0-0-0.in.bellnexxia.net  
[206.108.105.129]  
4 <10 ms 10 ms 10 ms core1-montreal02-pos11-0.in.bellnexxia.net  
[206.108.97.145]  
5 10 ms 20 ms 10 ms Ncore1-newyork83-pos4-0.in.bellnexxia.net  
[206.108.99.190]  
6 10 ms 20 ms 10 ms bx1-newyork83-pos3-0.in.bellnexxia.net  
[206.108.103.186]  
7 10 ms 20 ms 10 ms s1-gw-nyc-7-3.sprintlink.net [160.81.43.13]  
8 10 ms 20 ms 10 ms 144.232.7.93  
9 10 ms 20 ms 10 ms s1-bb22-nyc-14-0-2480M.sprintlink.net  
[144.232.7.102]  
10 30 ms 40 ms 30 ms s1-bb20-rly-15-0.sprintlink.net  
[144.232.18.26]  
11 30 ms 30 ms 20 ms 144.232.9.90  
12 30 ms 30 ms 30 ms gbr3-p50.wswdc.ip.att.net [12.123.9.50]  
13 50 ms 50 ms 50 ms gbr3-p80.sl9mo.ip.att.net [12.122.2.145]  
14 91 ms 150 ms 90 ms gbr3-p20.sffca.ip.att.net [12.122.2.74]  
15 110 ms 181 ms 120 ms 12.122.255.222  
16 90 ms 90 ms 90 ms 216.148.209.66  
17 90 ms 90 ms 90 ms www.redhat.com [216.148.218.195]
```

W tym przykladzie pakiet musial przejsc przez siec Bella do Nowego Jorku, gdzie wszedl do sieci Sprint. Ta siec prowadzi do Waszyngtonu D.C., gdzie AT&T przejmuje pakiet i przesyla gdzies do Kalifornii. Na koniec pakiet przechodzi do sieci docelowej. Jak widac, przenoszenie danych z miejsca na miejsce wymaga wiele pracy i współpracy.

Budowanie własnej sieci WAN

Omówilismy juz typy komutacji i sposob, w jaki współpracuja ze sobą wszyscy gracze: firma uzytkownika, lokalny ISP i operatorzy telefonii, wiec Czytelnik zapewne nie moze sie doczekac, aby zaczac budowac własny Internet II. Jednakze w wiekszosci przypadkow musimy dobrze przemyslec budzet i spróbować zrównoważyc koszty i uzytecznosc.

Bardzo ważne jest zrozumienie ruchu sieciowego, który odbywa sie pomiędzy różnymi lokalizacjami. Trzeba tu wziac pod uwage dwa czynniki: ile danych musimy przeslac pomiędzy dwiema lokalizacjami lub pomiędzy jedna lokalizacja i Internetem oraz jakie opóźnienia dopuszczane sa w transmisji danych. Podobnie jak różne protokoly wyboru trasy, omówione w rozdziale 19., te dwa czynniki beda bardzo ważne przy podjeciu decyzji, z jakiego typu polaczenia skorzystac. Wybrany typ polaczenia z kolei wpłynie na wybór sieciowego systemu operacyjnego i sposób jego rozmieszczenia.

Po ustaleniu, ile danych musimy przeslac pomiędzy dwiema lokalizacjami lub pomiędzy lokalizacją i Internetem oraz jaki poziom opóźnien bedzie dopuszczalny w transmisji danych, mozemy zaczac planowac polaczenia. W niektórych przypadkach warto tez zaplanowac nadmiarowosc lub alternatywne polaczenie pomiędzy lokalizacjami.

Polaczenie dwupunktowe

Pierwszy etap projektowania infrastruktury lacznosci jest stosunkowo latwy: musimy ustalic, które punkty w sieci trzeba ze sobą polaczyc — inaczej mówiac, obmyślic trasy pomiędzy dwoma punktami, którymi przepływać beda dane. W wiekszosci przypadków obejmuje to polaczenie z ISP lub firma telefoniczna, udostepniająca trase w naszej sieci od punktu A do punktu B — rozwiązanie proste, aczkolwiek prawdopodobnie kosztowne.

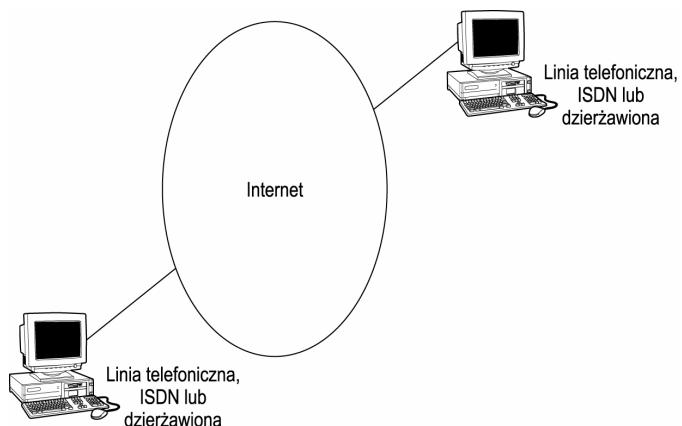
Polaczenie z ISP lub operatorem telefonicznym zwykle bedzie sie odbywac za pomoca lacza ISDN lub linii dzierzawionej. W obu przypadkach ISP lub firma telefoniczna moze wykorzystac przekazywanie ramek lub przekazywanie komórek do przesyłania danych do biura blizej miejsca przeznaczenia, a nastepnie kolejne polaczenie dwupunktowe do celu. Zazwyczaj najwolniejszym ogniwem w tej konfiguracji jest lacze z biura do operatora telefonii.

Tunelowanie

Zamiast prostego polaczenia dwupunktowego mozemy wybrac opcje zwana tunelowaniem. W jej przypadku uzyskujemy polaczenie z Internetem od jednego lub kilku dostawcow, co daje nam podstawowy poziom lacznosci pomiędzy dwiema lokalizacjami korzystajacymi z Internetu. Po uzyskaniu tej lacznosci na podstawowym poziomie, mozemy utworzyc wirtualne polaczenie za pomoca protokolu tunelowania. Rysunek 21.3 przedstawia dwa systemy polaczony z Internetem.

Rysunek 21.3.

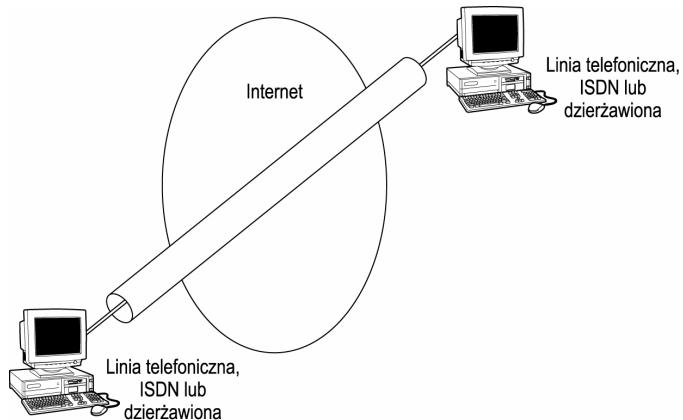
Dwa systemy polaczone
z Internetem



Teraz dwie stacje moga sie juz komunikowac. Problemem jest fakt, iz siec pomiedzy stacjami nie jest bezpieczna, wobec tego wszelkie dane przesyłane tym polaczeniem moga zostac odczytane w dowolnej z posredniczacych sieci. Nie jest to oczywiscie problem w przypadku zwyklych stron WWW, jednak ze w wielu przypadkach chcielibysmy zabezpieczyc dane. Wlasnie na tym etapie wiekszosc organizacji wybiera polaczenie tunelowe, pokazane na rysunku 21.4.

Rysunek 21.4.

Dwa systemy polaczone z Internetem mogą utworzyć wirtualny tunel, wykorzystując Internet jedynie jako połaczenie fizyczne



Jesli potraktujemy Internet (bazowa siec) jak połaczenie na poziomie fizycznym, a następnie opakujemy informacje do transmisji, bedziemy mogli wykorzystac Internet w roli połaczenia. Zasadniczo musimy uruchomic dwie sesje sieciowe — jedna zapewniajaca podstawowa łączność oraz druga grajaca role tunelu, przez który bedziemy przesyłac dane. Oznacza to, ze przez czesc warstw stosu sieciowego dane beda przechodzic wiecej niz jednokrotnie; jednakże otrzymamy proste rozwiazanie dla łącznosci.

W tunelowaniu wszystkie dane przesyłane pomiędzy dwoma systemami mogą być szyfrowane. Mozemy stosowac różne metody szyfrowania, w zależności od protokołu tunelowania użytego do stworzenia wirtualnej sieci prywatnej (VPN — *Virtual Private Network*). Do dostepnych protokołów naleza L2TP i PPTP.

Protokół PPTP

Wprawdzie PPTP (*point-to-point tunneling protocol* — protokół tunelowania dwupunktowego) jest przeznaczony przede wszystkim dla użytkowników korzystających z połaczeń telefonicznych, lecz może też posłużyć do połaczenia dwóch biur poprzez siec TCP/IP, jak np. Internet. Ponieważ protokół PPTP został opracowany przez Microsoft, sam tunel nie jest uwierzytelniany; zamiast tego dane logowania przesyłane tunelem są szyfrowane. Uwierzytelnianie zdalnego użytkownika zachodzi w tradycyjny sposób.

Dla użytkowników połaczeń telefonicznych PPTP jest protokołem łatwym do skonfigurowania i stosunkowo prostym. System kliencki inicjuje połaczenie z serwerem przez istniejące już połaczenie TCP/IP, szyfrowane tylko pomiędzy tymi dwoma punktami. Oznacza to, że jeśli protokół służy do połaczenia dwóch biur, to jedynie komunikacja dwupunktowa — to znaczy, na końcach tunelu pomiędzy dwoma systemami Windows grającymi role ruterów — będzie zaszyfrowana.

Na przykład, na rysunku 21.5 dane w sieciach A i B nie są szyfrowane, a jedynie dane przesyłane tunelem pomiędzy dwoma ruterami. Jeśli klient w sieci A wysyła dane do klienta w sieci B, dane nie są szyfrowane, ponieważ musialyby najpierw przejść do stoku protokołu PPTP, a następnie zostać zapakowane w pakiet TCP/IP. Ponieważ dane klientów w sieci A przechodzą do stoku IP, nie będą w pierwszej kolejności przesłane przez stok PPTP, a więc nie zostaną zaszyfrowane.

Na rysunku 21.6. dane z aplikacji przechodzą w dół stoso protokolu PPTP, a nastepnie przez stos TCP/IP. Pierwszy stos zapewnia szyfrowanie, zas drugi faktyczny transport. Poniewaz pakiety z sieci nie beda podazac ta sama trasa, PPTP tak naprawde nie nadaje sie do laczenia sieci.

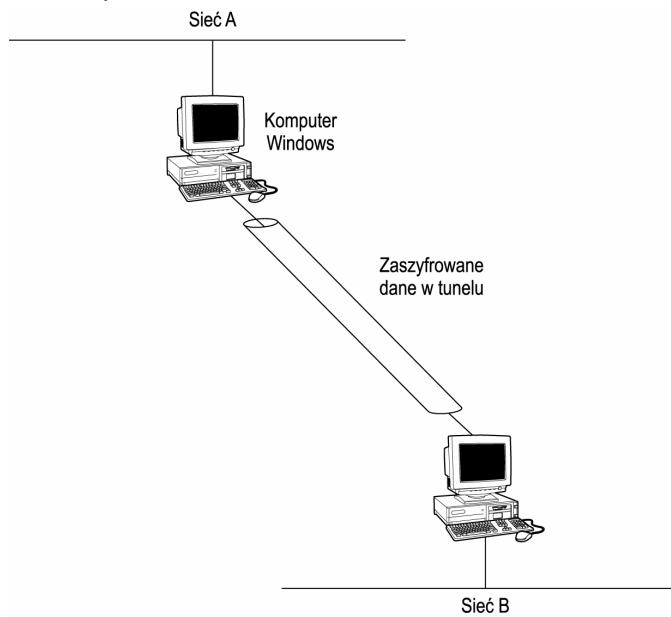
Protokól L2TP

L2TP (*Layer 2 tunneling protocol* — protokól tunelowania w warstwie 2.) jest bardziej popularnym protokolem na różnych platformach. Poniewaz L2TP jest obsługiwany przez innych producentów, stanowi wspólny protokól dla sieci, w których systemy Microsoftu nie sa uzywane w roli ruterów. Firmy Microsoft (dla Windows 2000) i Cisco opracowaly wspólnie L2TP jako metode zabezpieczania transmisji danych we wszystkich typach nosników.

Kolejna zaleta L2TP jest zdolnosc do pracy w dwóch trybach — tunelowania i pracy dwupunktowej. Tryb dwupunktowy dziala podobnie jak w PPTP — jedynie dane z lokalnego systemu sa szyfrowane, a jesli system ten gra role rutera, nie szyfruje przekazywanych danych. Poniewaz jednak protokól L2TP zostal wbudowany w stos protokolów TCP/IP, mozemy wybrac tryb tunelowania, który pozwala L2TP szyfrowac również dane przechodzace przez rutery.

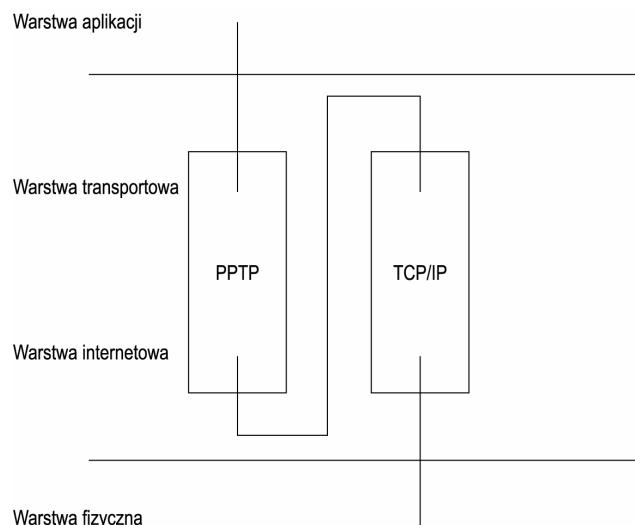
Rysunek 21.5.

Polaczenie PPTP
pomiedzy dwiema
sieciami



Rysunek 21.6.

Architektoniczne spojrzenie na sposób działania protokołu PPTP



Jesli planujemy uzyc Internetu jako sieci szkieletowej, warto zastosowac L2TP do przesyłania danych pomiedzy koncami polaczenia. Lecz jak bedzie w przypadku linii dzierzawionych? Jak juz powiedzieliśmy, trwale polaczenia dwupunktowe nie sa stosowane — przekazywanie ramek i przekazywanie komórek jest uzywane przez kompanie telefoniczne nawet dla polaczen dwupunktowych, poniewaz pozwalaja lepiej wykorzystac przepustowosc. Wobec tego warto szyfrowac również dane przesylane takimi liniami.

Szyfrowanie danych jest zwykla funkcja ruterów. Juz od jakiegos czasu w wiekszosci ruterów wbudowuje sie jakieś formy szyfrowania. L2TP staje sie standardowym protokołem uzywanym w ruterach.



Pelniejsze omówienie bezpieczenstwa zawiera nastepny rozdział.

Planowanie dostępu zdalnego

Rosnaca liczba użytkowników pracujacych z domu, a takze użytkowników laptopów, uczynily w ciagu ostatnich kilku lat z dostępu zdalnego zagadnienie kluczowe. Szczerze laptopy stanowią wyzwanie, poniewaz czasami polaczone sa z siecią za pomocą karty sieciowej Ethernet, czasami za pomocą modemu, a czasem nie sa podlaczone wcale. Wykorzystanie protokołów typu DHCP uproscilo łączność, poniewaz klienty nie musza nieustajaco rekonfigurowac informacji o adresach IP.

Do niedawna zapewnienie dostępu zdalnego było kwestią przyniesienia do biura kilku modemów, podciagniecia linii telefonicznych i rozdania numerów telefonów. Ta technika działała, lecz często była kosztowna z uwagi na ceny modemów, opłaty za połączenia telefoniczne oraz wsparcie użytkowników i opłaty za połączenia międzymiastowe. Wraz z rozwojem techniki pojawiło się jednakże kilka opcji dostępnych przy przylaczaniu użytkowników zdalnych.

Wybór strategii polaczen telefonicznych

Dokonujac oceny dostepnych opcji polaczen telefonicznych, powinnismy pamietac o kilku czynnikach. Pierwszym jest bezpieczenstwo. Jesli nasza organizacja uwaza, iz szyfrowanie za pomoca L2TP lub PPTP nie jest wystarczajace, bedziemy musieli zastosowac własne serwery dostepowe. Jesli L2TP z potrójnym szyfrowaniem DES wystarczy, to mozemy je wykorzystac. Uzytkownicy moga wówczas laczyc sie z ISP przez laze telefoniczne i tworzyc polaczenia poprzez Internet.

Wlasne konta dostepu telefonicznego

Jesli naprawde musimy stosowac własne konta dostepu telefonicznego z uwagi na bezpieczenstwo, przygotujmy sie na koszty. Uruchomienie wlasnej puli modemów wymaga wyspecjalizowanego sprzetu i linii telefonicznych. Na dodatek musimy zastanowic sie, jak obsluzyc uzytkowników podrózujacych. Albo zmusimy ich do wybierania numeru centralnego, albo rozmiescimy modemy w kazdym oddziale, z którym uzytkownik bedzie sie laczyc. Jesli uzytkownicy lacza sie z numerem centralnym, mozemy rozwazyc opcje numeru bezplatnego (0-800), aby obnizyc koszty polaczen. Jesli ta opcja jest niesiagalna, mozemy pomyslec nad polaczeniami *callback*. Do tego moga posluzyc polaczenia typu wychodzacego, aby utrzymac niskie koszty. Jedyna sytuacja, która spowoduje klopoty w tej strategii to ta, gdy uzytkownicy dzwonia z hotelu — numer telefonu w pokoju moze nie byc dostepny z zewnatrz.

Umieszczenie modemu w kazdym biurze moze zminimalizowac koszty, lecz nadal bedziemy mieli do czynienia z rachunkami za polaczenia zamiejscowe, gdy podrózujacy uzytkownicy beda daleko od biura. Potrzebny bedzie jeszcze jakis sposob kontroli uwierzytelniania, poniewaz uzytkownik moze byc w Houston, a jego konto w Albany. W takim przypadku informacje o koncie uzytkownika musza byc dostepne dla serwera w Houston. Kolejny problem pojawia sie, gdy uzytkownik nie moze polaczyc sie, nie znajac numeru dostepowego w Houston. Takie informacje musza byc przechowywane w laptopie; w przeciwnym razie uzytkownicy beda zmuszeni dzwonic do obslugi technicznej w drugim mieście, aby otrzymac numer.

Wykorzystanie tunelowania

Alternatywa dla posiadania wlasnych modemów dostepowych jest przydzielenie uzytkownikom kont u lokalnego, krajowego lub medzynarodowego dostawcy uslug internetywych. Numery dostepowe lokalnego ISP zapewnia podstawowe polaczenie z Internetem, zas numery obslugi technicznej ISP (zwykle 0-800 opłacane przez ISP) zapewnia podstawowe wsparcie w razie klopotów z polaczeniem. W ten sposob zwolnimy budżet eksploracyjny z kosztów modemów, linii telefonicznych i polaczen zamiejscowych — nawet liczba telefonów do pomocy technicznej z zapytaniami o numery telefoniczne moze sie zmniejszyc.

Uzytkownicy po podlaczeniu do Internetu moga „dodzwonic sie” do naszego serwera PPTP lub L2TP. Sprawdza sie to również dla uzytkowników domowych, którzy posiadaja juz modem kablowy lub laze DSL. Nawet po wliczeniu kosztów lacza internetywego i kont telefonicznych ta metoda jest tansza od wlasnych uslug dostepu telefonicznego, zwlaszcza gdy pojawia sie nowe pokolenie modemów.

Praca zdalna

Omówiliśmy jak dotad czesc zagadnien zwiazanych z lacznoscia, wiec pora przyjrzec sie problemom pojawiajacym sie podczas pracy zdalnej. Wielu uzytkowników narzeka, iż praca przez modem jest wyjatkowo wolna. Maja racje. 56 kb/s to predkosc mniejsza niz w przypadku zwyklych sieci.

Niestety, niewiele mozna zrobic, by przyspieszyc proste czynnosci opierajace sie na plikach (na przyklad otwieranie plików z sieci lub zapisywanie z powrotem). Gdy uzytkownik usiluje przeniesc plik z jednego miejsca w drugie, wówczas operacja bedzie powolna, poniewaz plik jest przenoszony przez zdalny system. Inaczej mówiac, caly plik jest przenoszony do pamieci lokalnej w systemie uzytkownika, a nastepnie do nowej lokalizacji. Sa jednak dostepne pewne opcje, które mozemy wykorzystac próbujac pracowac z aplikacija w odleglej sieci, na przyklad: Telnet, narzedzia oparte na WWW i serwer terminali.

Telnet

Telnet (emulacja terminala) jest narzedziem, które przyspiesza uslugi dla zdalnego uzytkownika. Telnet tworzy sesje w serwerze i w rzeczywistosci wykonuje polecenia w tym komputerze. Usluga ta jest zasadniczo tekstowa, lecz do przenoszenia i kopowania plików oraz uruchamiania aplikacji tekstowych nadaje sie dobrze. Komputer klienta wprowadza dane i polecenia oraz sluzy jako urzadzenie wyjsciowe, to znaczy, ze jedynie nacisniecia klawiszy i odpowiedzi tekstowe sa przesypane przez polaczenie.

Przegladarki WWW

Poniewaz Telnet jest narzedziem tekstowym, uzytkownicy musza byc w stanie pisac i pracowac z prostymi poleceniami tekstowymi. Oczywiscie nie nadaje sie to dla wszystkich uzytkowników, wobec czego wiele organizacji obecnie uzywa aplikacji opartych na WWW, zamiast starszych aplikacji tekstowych lub tradycyjnych typu klient-serwer. Pojawienie sie takich technologii jak Java umoziwiło nawet wysyłanie do przeglądarki skryptów i programów uruchamianych po stronie klienta. W ten sposób tworzono jest srodowisko robocze klient-serwer nie wymagajace szybkich laczy.

W idealnych warunkach, gdy calosc kodu jest przechowywana w serwerze, a twórcy oprogramowania zachowuja prostote projektów, mozemy uruchamiac aplikacje oparte na WWW za pomoca przeglądarki nawet przy najwolniejszych laczach. Zdolosc do korzystania z wolnych laczy i malych klientow zaczyna byc wzarna z uwagi na wzrost popularnosci palmtopów z dostepem do Internetu — w wielu przypadkach urzadzenia te pozwalaja jedynie na transfer z predkoscia 19,2 kb/s i (zazwyczaj) nie posiadaja wbudowanych klientow jazyka Java.

Serwer terminali

Jesli musimy cala swoja prace wykonywac zdalnie, mozemy zdecydowac sie na serwer terminali. Serwery te sa rozszerzeniem idei Telnetu; pozwalaja na zdalne stosowanie grafiki oprócz tekstu. Citrix dostarcza serwery terminali dla wiekszosci platform, zas

Microsoft wbudował serwer terminali we wszystkie odmiany systemu operacyjnego Windows 2000 Server.

Rozdział 22.

Planowanie bezpieczeństwa sieci

W tym rozdziale:

- ◆ Szacowanie ryzyka
- ◆ Równowazenie bezpieczeństwa i uzytecznosci
- ◆ Zabezpieczanie sieci

Bezpieczeństwo jest zagadnieniem pierwszoplanowym dla kazdego, kto pracuje z komputerami. Czytelnik zna zapewne opowiesci grozy o systemach, które padły ofiara włamania i użytkownikach, którzy stracili wszystkie swoje dane — i szczerze mówiąc, wiekszosc z tych historii jest prawdziwa. To, czego nam trzeba, to proaktywna strategia ochrony danych przed wszelkimi mozliwymi zdarzeniami. Planowanie z góry jest niezbedne, poniewaz gdy pojawia sie problemy, bedziemy zbyt zaabsorbowani próbami odzyskania informacji i przywrócenia dostępu użytkownikom, zeby zastanawiac sie, skad wziac nowy komputer lub gdzie schowalismy kopie zapasowa z ubieglego miesiaca. Jednakze niniejsza ksiazka poswiecona jest TCP/IP, a nie odzyskiwaniu danych. Wobec tego w tym rozdziale zajmiemy sie zagrozeniami bezpieczeństwa związanymi z TCP/IP.

Szacowanie ryzyka

Podczas planowania strategii bezpieczeństwa pierwszym krokiem jest oszacowanie ryzyka. Inaczej mówiąc, musimy przeanalizowac posiadane dane i stosowane zwyczaje, a nastepnie ustalic wrażliwe punkty. Napastnicy moga wziac na cel dwa główne obszary systemu: dane lub usługi. Skutki ataku moga byc różne, w zaleznosci od zaatakowanych danych lub uslug. Na przykład, jesli ktos zdobedzie liste telefonów przedsiębiorstwa, moze to byc irytujace, lecz nie doprowadzi do bankructwa. Natomiast jesli ktos skradnie dokumentacje naszego cennego produktu, zmodyfikuje ja i sprzedzie produkt po nizszej cenie — no cóz, mówiąc wprost, bedzie zle.

Kolejnym ważnym krokiem planowania zasad bezpieczeństwa jest identyfikacja różnych poziomów bezpieczeństwa dla różnych typów danych. Aby ustawić różne poziomy zabezpieczeń, musimy zidentyfikowac kryteria poufnosci — czyli, jakie dane firmy mozna udostepnic publicznie, a jakie trzeba chronic.

Przykładem danych, które można, a nawet powinno się, udostępniać są materiały marketingowe. Jednym z zadań tych materiałów (do których należą również witryny WWW) jest stworzenie określonego wizerunku przedsiębiorstwa. Inne firmy decydują, czy prowadzić z przedsiębiorstwem interesy, również na podstawie jego wizerunku. Znaczy to, że czytanie tych informacji przez ludzi jest pozadane, natomiast wprowadzanie zmian przez osoby nieupoważnione jest bardzo niepożądane.

Dochodzimy tu do kolejnej kwestii — w przypadku pewnych danych, na przykład arkuszy kont w systemie finansowym, wiele osób potrzebuje prawa do ich czytania, lecz jedynie kilku pracowników powinno mieć uprawnienia do modyfikacji. Tabela 22.1 ilustruje te kwestie — wymienia kilka typów informacji istniejących w przedsiębiorstwie, kto może je tworzyć i korzystać z nich oraz skutki utraty tych danych dla przedsiębiorstwa.

Tabela 22.1. Typy informacji i związane z nimi zagadnienia bezpieczeństwa

Typ informacji	Używane przez	Tworzone przez	Wpływ na
Witryny WWW i materiały marketingowe	Wszystkich	Marketing	Wizerunek firmy, a co za tym idzie, sprzedaż
Dane wewnętrzne, np. listy telefonów i karty urlopowe	Pracowników firmy	Dział personalny i kierownictwo	Codzienna funkcjonowanie firmy i ataki socjotechniczne
Dane prywatne, np. prognozy finansowe	Kierownictwo	Finanse i kierownictwo	Wizerunek firmy i codzienne funkcjonowanie
Tajne informacje, np. specyfikacje i plany projektowe	Produkcja	Eksperci w danej dziedzinie	Ogólna pomysłowość firmy

Przedstawione tu typy informacji są przykładowe. Faktyczne informacje, z którymi Czytelnik będzie miał do czynienia, będą inne, lecz podczas ustalania wymaganych poziomów bezpieczeństwa należy zwracać uwagę na te same cechy informacji w przedsiębiorstwie.

Napastnicy mogą kierować się różnymi motywami: chcą zdobycia informacji, które posłużą do innych ataków lub do konkurenckiego z inną firmą, chcą zmodyfikowania danych tak, by stały się bezużyteczne lub chcą wpłyńcia na procesy, używające tych informacji.

Atak na dane zwykle oznacza, że ktoś z powodzeniem zaatakował serwer, zdobył do niego dostęp, a następnie usunął lub zmodyfikował dane w serwerze. Jednym z najprostszych sposobów, by uzyskać dostęp do serwera, jest zdobycie nazwy użytkownika i hasła. Sa na to różne sposoby. Na przykład, za każdym razem, gdy pobieramy lub wysyłamy pliki za pomocą FTP, hasło i dane przesypane są przez sieć otwartym tekstem; poczta elektroniczna również wysyłana jest przez Internet otwartym tekstem, a za każdym razem, gdy czytamy poczty z internetowego konta pocztowego, też przesyłana jest otwartym tekstem. Inaczej mówiąc, większość protokołów używanych w Internecie — w istocie większość protokołów tradycyjnie używanych z TCP/IP — wysyła dane nieszyfrowanym tekstem. Wobec tego każdy użytkownik sieci posiadający analizator pakietów (może nim być chociażby program uruchomiony w laptopie) może przechwytać wszelkie informacje wedrujące w Internecie,łącznie z naszą nazwą użytkownika i hasłem.

Nawet jesli napastnik nie zdobedzie nazwy uzytkownika i hasla, serwery moga zostac zaatakowane. Na przyklad, ataki typu blokady uslug (*denial of service*) polegaja na wyslaniu takich objetosci danych do serwera, ze ten nigdy nie nadazy z obslugiwaniem zadan. W innych przypadkach napastnik moze wyslac odpowiednio znieksztalcony pakiet — na przyklad nastepujace po sobie zadania synchronizacji, ktore serwer musi obsluzyc. W poczatkach istnienia systemow Windows 95 i NT 4.0 jeden z popularnych atakow polegal na wyslaniu znieksztalconego pakietu na port 135 (port RPC Microsoft), powodujacego zawieszenie systemu, a w wielu przypadkach restart — ten atak byl znany pod nazwa WinNuke. W atakach tego typu dane nie sa atakowane, lecz przestaja byc dostepne.

Po takim wprowadzeniu do atakow na systemy Czytelnik moze poczuc ochote wyrwac ze sciany kabel laczacy komputer z Internetem. Jesli jednak uzywamy dobrej zapory firewall i stosujemy sie do zasad bezpieczenstwa, Internet nie jest najwiekszym zagrozeniem. W rzeczywistosci wiekszosc udanych atakow na systemy pochodzi ze srodka organizacji, nie z zewnatrz. Na przyklad, moglismy wpuscic do biura konsultanta, rewidenta lub inna osobe niezatrudniona z laptopem. Jesli Czytelnik podlaczyl taki komputer do sieci, aby umozliwic goisciowi drukowanie, zapewne nie zdawał sobie sprawy z podejmowanego ryzyka. Gosc mogl korzystac z „weszyciela pakietow” (*packet sniffer*) lub programu do lamania hasel, np. *10phcrack*.

Oznacza to, ze musimy dokladnie przeanalizowac system pod katem mozliwych zagrozen wewnętrznych. Jaki typ dostepu uzytkownicy posiadaja do poufnich danych przedsiebiorstwa? Ogólnie mówiac, atak z wewnatrz moze pochodzić ze strony aktualnego pracownika, bylego pracownika, konsultanta, pracownika tymczasowego lub nawet osoby obcej. We wszystkich tych przypadkach mozemy zrobic bardzo niewiele dla ochrony informacji, poza rozważnym zarządzaniem kontami i opracowaniem dobrych zasad bezpieczenstwa. Niektórzy sa po prostu ciekawscy i nie mają zlych zamiarów; w innych przypadkach atak jest niebezpieczna próba dostepu do poufnich informacji.

Ataki wewnętrzne moga również opierac sie na taktykach socjotechnicznych, wykorzystujacych zyczliwosc osób trzecich. Czytelnik zapewne widzial niejeden film, w którym bohater po prostu wchodzi do budynku udajac pracownika, dostawce pizzy lub sanitariusza. W podobny sposób napastnik moze udawac serwisanta sprzetu komputerowego i niewinnie zapytac uzytkownika o haslo, aby móc naprawic komputer. Napastnik moze tez zadzwonic do pomocy technicznej z telefonu wewnętrznego, udajac takiego to a takiego asystenta i poprosic o zmiane swojego hasla. Walka z tego typu atakami polega na zaznajamianiu kazdego pracownika z zasadami bezpieczenstwa firmy i zadaniu zgłoszenia wszelkich podejrzanych czynnosci w otoczeniu.

Mozemy skuteczniej bronic sie przed atakami wiedzac, jakie dane sa dostepne dla potencjalnych hakerów. Na przyklad, w wiekszosci sieci opartych na systemach Windows uzytkownik moze przechwycic przesylane w sieci pakietы z danymi uwierzytelniajacymi, a nastepnie wyniesc dane i złamac hasla poza biurem. Plik z dosłownie tysiącami hasel mozna wyniesc na jednej dyskietce. Programy typu LC3 (najnowsza wersja 10phcrack) oraz PWDump ułatwiają przechwytywanie pakietów z sieci lub nawet skopiowanie bazy danych SAM (*Security Accounts Manager*) z nazwami uzytkowników i haslami.

Mamy wiec porzucic siec i wróć do systemu papierkowego? Nie, sa kroki zaradcze, które mozemy podjac, na przyklad zastosowanie IPSec do zabezpieczenia sieci. Ponadto, dobra polityka bezpieczenstwa przyniesie efekty, jesli bedziemy ja stosowac konsekwentnie w całej organizacji.

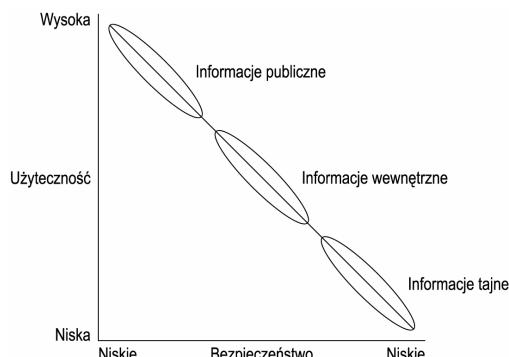
Równowazenie bezpieczenstwa i uzytecznosci

Zanim chocby pomyslimy nad szczegółowymi potrzebami w dziedzinie bezpieczeństwa, musimy rozważyc, jak zrównoważyc uzytecznosć sieci z bezpieczeństwem, jakie chcemy osiągnac. Jest to związane z poprzednimi rozważaniami na temat różnych poziomów bezpieczeństwa, poniewaz wprawdzie czesc danych moze byc dostepna dla uzytkowników, lecz inne musza byc scisłe kontrolowane.

Rysunek 22.1 przedstawia zaleznosc pomiędzy uzytecznoscia i bezpieczeństwem. Na tym wykresie przedstawione sa trzy różne poziomy danych — publiczne, wewnętrzne i tajne. Prosze pamietac, iż nazewnictwo poszczególnych klas danych moze byc różne w różnych przedsiębiorstwach.

Rysunek 22.1.

*Istnieje silna zaleznosc
pomiędzy
uzytecznoscią
i bezpieczeństwem*



Gdy zaczniemy planowac bezpieczeństwo w sieci, musimy zdecydowac, jak nazwac poszczególne poziomy bezpieczeństwa, a nastepnie ustalic, wedlug jakich kryteriów umieszczac dane w kazdej z kategorii. Podczas opracowywania tych kryteriów prosze pamietac: im prosciej, tym lepiej. Jesli nie chcemy sklasyfikowac wszystkich dokumentow w całej firmie, wytyczne musza byc wystarczajaco proste, by inni uzytkownicy mogli je zrozumiec i zastosowac. Po drugie, warto objasnic zagrozenia w ramach polityki bezpieczeństwa lub podczas przyjmuwania nowych pracowników. Inaczej mówiac, musimy wyjasnic, dlaczego dany dokument musi byc chroniony. Dla Czytelnika moze to byc az zanadto oczywiste, lecz ktos z mniejszym doswiadczeniem moze zaszufladowac zasady bezpieczeństwa do „kolejnych głupich przepisów” — o ile nie wyjaśnimy mu tych zasad.

Przeanalizowanie typów ataków, jakie moga nastapic, typów posiadanych danych i wymaganych dla nich poziomów bezpieczeństwa pomoze nam przejsc do zabezpieczania sieci.

Zabezpieczanie sieci

Proces zabezpieczania sieci obejmuje dwa aspekty — zabezpieczenie dostepu do danych i zabezpieczenie transmisji danych. Wiele organizacji zabezpiecza dostep do danych, lecz bardzo niewiele z nich wykonuje kolejny krok — zabezpieczenia transmisji danych. Mowiac inaczej, wiekszosc organizacji wymaga uwierzytelnienia uzytkownika przed dostepem do danych, lecz nie szyfruje wszystkich transmisji. Niestety siec nie jest tak naprawde chroniona, o ile dostep do danych i transmisja danych nie sa zabezpieczone.

Szyfrowanie transmisji danych

Szyfrowanie danych przesyłanych przez siec przedsiebiorstwa zwiększa bezpieczeństwo uwierzytelniania i zapobiega przechwytywaniu przesyłanych danych przez osoby niepowolane. Dane moga byc szyfrowane za pomocą dowolnego z kilku dostepnych standardów, stosujacych różne algorytmy szyfrowania wiadomosci tak, by jedynie prawowity odbiorca wiedzial, jak je rozszyfrowac. Wiekszosc metod szyfrowania polega na wykorzystaniu wartosci wspólnego klucza do szyfrowania danych, a nastepnie deszyfracji po stronie odbiorcy. Ten typ szyfrowania jest ogólnie znany pod nazwą *infrastruktury klucza publicznego* (PKI — Public Key Infrastructure).

W skrócie: PKI wykonuje dwie funkcje, zapewne juz znane Czytelnikowi. Mozemy umiescic w wiadomosci podpis cyfrowy lub zapiecztowac (zaszyfrowac) wiadomosc. Dla kazdej z tych funkcji proces przebiega nieco odmiennie i dla kazdego uzywana jest inna para kluczowa. Para kluczowa uzywana jest razem — jeden sluzy do szyfrowania danych, zas drugi do deszyfracji. Pary kluczowe sa tworzone w taki sposob, by tylko drugi z nich móg³ posluzyc do odszyfrowania danych, zaszyfrowanych pierwszym kluczem. Sa to tzw. *klucze asymetryczne*, poniewaz kazdy jest inny.

Podpisywanie wiadomosci

Proces podpisania wiadomosci nie oznacza zaszyfrowania jej, lecz umieszczenie elektronicznego podpisu na koncu wiadomosci. Podpis zostaje zweryfikowany po drugiej stronie, w celu sprawdzenia, czy wiadomosc nie zostala zmodyfikowana w trakcie transmisiJI z miejsca na miejsce.

Do podpisania wiadomosci tworzony jest jej skrót (*digest*) za pomocą funkcji mieszajacej. Prostym przykladem mieszania moze byc wziecie pierwszych 1024 bitów wiadomosci i wykonanie funkcji XOR (patrz uwaga ponizej) z kolejnymi 1024 bitami. Na wyniku znnowu zostanie wykonana operacja XOR z nastepnymi 1024 bitami i tak dalej, az do konca wiadomosci. Na koniec otrzymamy lancuch 1024 bitów, unikatowy dla tej wiadomosci — a przynajmniej na tyle unikatowy, ze wartosc zmieni sie, jesli ktos choc troche zmodyfikuje wiadomosc.



Nazwa funkcji XOR pochodzi od Exclusive OR. Jest to logiczne porównanie dajace 0, jesli dwa bity sa takie same oraz 1, jesli sie różnia.

Wprawdzie proces mieszania niekoniecznie bedzie uzywac funkcji XOR, lecz procedura jest podobna. Liczba bitów wyniku zalezy od uzytego konkretnego algorytmu, lecz ogólnie wynosi przynajmniej 512. Ten skrót wiadomosci zostaje nastepnie zaszyfrowany kluczem prywatnym autora i wiadomosc zostaje wyslana.

Po odebraniu wiadomosci skrót zostaje z niej usuniety i za pomoca tego samego algorytmu, co u nadawcy, liczony jest u odbiorcy skrót wiadomosci. Nastepnie oryginalny odebrany skrót wiadomosci zostaje odszyfrowany za pomoca publicznego klucza nadawcy, sluzacego do podpisywania, a oba skróty zostaja porównane. Jesli sa takie same, dane podczas transportu nie zostaly zmodyfikowane. Jesli sie różnia, ktos mógł manipulowac wiadomoscia. Oczywiscie oznacza to, ze klucz publiczny sluzacy do podpisywania musi byc znany systemowi odbiorcy; w przeciwnym razie nie mozna by bylo sprawdzic wiadomosci i jej podpiswanie nie mialoby zbyt duzego sensu.

Pieczetowanie wiadomosci

Aby zapieczetowac wiadomosc, dzieli sie ja na kawalki, które zostaja zaszyfrowane za pomoca klucza publicznego odbiorcy. Wiadomosc moze nastepnie zostac wyslana przez siec, a po stronie odbiorcy odszyfrowana za pomoca jego klucza prywatnego. Podobnie jak podpiswanie, pieczetowanie opiera sie na mozliwosci udostepniania kluczy publicznych, co prowadzi do problemu, gdy wiadomosc ma zostac wyslana do wielu uzytkowników.

Plik klucza uzyty do zaszyfrowania wiadomosci jest generowany dla wielu adresatów. Klucz ten zostaje nastepnie zaszyfrowany indywidualnie dla kazdego odbiorcy za pomoca jego klucza publicznego. Po wyslaniu wiadomosci kazdy odbiorca moze ja odszyfrowac, uzywajac wlasnego klucza prywatnego. Nadal pozostaje jednak problem dystrybucji kluczy.

Szyfrowanie kluczem symetrycznym

Rozprowadzenie kluczy nawet w malej sieci i zapewnienie, aby wszyscy uzytkownicy dysponowali wszystkimi kluczami publicznymi, moze byc oczywiscie trudne. Na dodatek, gdyby do szyfrowania danych byly zawsze uzywane te same klucze, latwiej bylo by je złamac. Wobec tego infrastruktura PKI zwykle jest stosowana podczas nawiazywania sesji, tak by mozna było przeslac pomiedzy dwoma systemami wspólny klucz tajny. Inaczej mówiąc, PKI sluzi do szyfrowania klucza sesji, który zostanie wykorzystany przez dwa systemy.

Stosujac PKI jedynie do nawiazania sesji, a nastepnie korzystajac z klucza sesji, możemy stosunkowo łatwo poradzic sobie z szyfrowaniem i deszyfracją danych oraz z częsta wymiana kluczy.

Standardowe algorytmy szyfrowania

Omówilismy już klucze symetryczne i asymetryczne, wiec warto przyjrzec się przez chwilę niektórym standardowym algorytmom stosowanym do podpisywania i szyfrowania danych. Na poczatek wymienimy kilka protokołów uwierzytelniania wiadomosci:

- ◆ *DSA (Digital Signature Algorithm — algorytm podpisu cyfrowego)* — ten standard jest używany do generowania i weryfikacji podpisów. Pozwala na stosowanie kluczy o długości do 1024 bitów.
- ◆ *SHA-1 (Secure Hash Algorithm — algorytm bezpiecznego mieszania)* — jeden z algorytmów mieszających. Jest wolniejszy od MD5, lecz pozwala na dłuższe skróty wiadomości. SHA-1 może z wiadomości o długości do 2^{64} bitów (2EB) — eksponentem 160-bitowy skrót.
- ◆ *MD5 (Message Digest 5)* — kolejny popularny algorytm mieszający. Jest szybszy od SHA-1, lecz daje jedynie 128-bitowe skróty wiadomości.
- ◆ *HMAC (Hash Message Authentication Code)* — na podstawie wartości klucza jest tworzony kod uwierzytelniający wiadomości (MAC — *Message Authentication Code*), zawarty następnie w skrócie w taki sposób, że oryginalne dane i MAC zostają zmieszczone w jeden skrót. HMAC jest jedynie dodatkiem do SHA lub MD5.

Jak Czytelnik zapewne pamięta, uwierzytelnianie jest jedynie połowa sukcesu w zabezpieczeniu sieci. Musimy dodatkowo zapewnić szyfrowanie danych. Do standardów szyfrowania należą:

- ◆ *DES (Digital Encryption Standard — standard szyfrowania cyfrowego)* — przypuszczalnie najbardziej popularny standard, udostępniający szybkie, 56-bitowe szyfrowanie.
- ◆ *DES CBC (DES Cipher Block Chaining — wiązanie lancuchowe bloków zaszyfrowanych DES)* — w tym mechanizmie pierwszy blok otwartego tekstu jest szyfrowany za pomocą standardowego DES. Wszystkie następne bloki przechodzą przed zaszyfrowaniem operacje XOR z zaszyfrowaną wersją poprzedniego bloku tekstu.
- ◆ *3DES (Triple DES — potrójny DES)* — ogólnie mówiąc, w tym standardzie pojedynczy blok jest szyfrowany trzykrotnie trzema różnymi kluczami 56-bitowymi.
- ◆ *DESX (DES XOR)* — kolejna odmiana standardu DES, w której nie zaszyfrowane dane przechodzą przed zaszyfrowaniem operacje XOR z częścią klucza. Zaszyfrowany blok danych przechodzi następnie operacje XOR z inną częścią klucza.
- ◆ *RC2 (Rivest's (Ron's) Cipher 2 — szyfr Rona Rivesta 2)* — RC2, opracowany przez Rona Rivesta z RSA (www.rsa.com), jest algorymem szyfrowania blokowego (szyfrowania grup danych), używającym bloków o stałej długości 64 bitów. Długość klucza jest zmieniona.
- ◆ *RC4* — podobny do RC2, z wyjątkiem użycia szyfrowania potokowego. Inaczej mówiąc, długość bloku szyfrowanych danych nie jest stała.
- ◆ *RC5* — najnowsza wersja protokołu RC, używająca bloków 32-, 64- lub 128-bitowych, szyfrowanych kluczem o długości do 2048 bitów.

Uwierzytelnianie użytkowników

Proces uwierzytelniania zwykle polega na podaniu nazwy i hasła przez użytkownika. Ten standard jest stosowany już od dawna, a jego ograniczenia są dobrze znane. Jednakże korzystanie z haseł ma swoje wady: użytkownicy zapominają hasła, zapominają gdzie je zapisali lub stosują bardzo proste hasła, które mogą zostać łatwo odgadnięte.

Wiekszosc sieciowych systemów operacyjnych udostepnia obecnie jakies metody zmuszenia uzytkowników do korzystania z mocnych hasel o przynajmniej minimalnej dlugosci. Wymuszenie minimalnej dlugosci hasla moze byc przydatne, lecz prowadzi do wiekszej liczby prosb do operatorów o zmiane hasla oraz do czestszego pojawiania sie karteczek z zanotowanym haslem. Istotnie, mozemy czasem stwierdzic, jak powaznie dany uzytkownik traktuje bezpieczenstwo, wedlug polozenia tej karteczki: przyklejona do monitora — niskie bezpieczenstwo; pod klawiatura — srednie bezpieczenstwo; pod pudelkiem na olwki w szufladzie — wysokie bezpieczenstwo.

Tak czy tak, hasla nadal beda uzywane przynajmniej jako czesc skladowa uwierzytelniania uzytkownika. Wazne jest również, jak haslo jest przekazywane od uzytkownika do serwera.



Mocne haslo zawiera znaki z trzech lub czterech różnych grup — duzych liter, malych liter, liczb i symboli specjalnych.

Hasla nie szyfrowane

Prosta metoda zabezpieczenia jest wysyłanie hasla z systemu klienta do serwera otwartym tekstem. Metoda ta byla powszechnie stosowana w poczatkach rozwoju technik komputerowych i nadal uzywana jest dosc często. Wszystkie protokoly bazujace na Uniksie (czyli na TCP/IP) na poczatku uzywaly tej metody. Na przyklad, SMTP, POP, FTP, Telnet, NNTP i inne protokoly uzywaja uwierzytelniania otwartym tekstem.

Wysyłanie hasel otwartym tekstem mozemy uznam za metode zabezpieczania „na apatie”, poniewaz liczymy na apatie innych uzytkowników, którym nie bedzie sie chcialo przechwycic hasla. Z drugiej strony, gdy uzywamy dzis hasel otwartych, zazwyczaj samo polaczenie jest szyfrowane, co pozwala uwierzytelniac sie otwartym tekstem bez obaw o przechwycenie hasla po drodze przez osobe trzecia. Dwiema popularnymi metodami szyfrowania hcznosci sieciowej sa IPSec i SSL, omowane w dalszej czesci rozdzialu. Ogólnie mówiac, nie nalezy stosowac hasel nie szyfrowanych bez szyfrowania polaczenia.

Uwierzytelnianie NT/LAN Manager

W srodowisku Windows uwierzytelnianie technika NT/LAN Manager (NTLM) jest uzywane do uwierzytelniania polaczen. Serwer poprzez zaszyfrowane polaczenie wysyla 8-bajtowe wezwanie (*challenge*) do komputera klienta. Ten wykorzystuje haslo uzytkownika do utworzenia 21-bajtowego klucza sesji, który posluzy do zaszyfrowania wezwania. Zaszyfrowane wezwanie zostaje przeslane do serwera zabezpieczonym kanalem.

Serwer deszyfruje nastepnie odpowiedz klienta i wydobywa klucz sesji. Ten zostaje porownany z kluczem sesji, który serwer utworzył za pomoca hasla uzytkownika, i jesli oba sa identyczne, uzytkownik uzyskuje prawo zalogowania sie do serwera. Ta procedura posiada oczywista slaba strone, poniewaz klucz sesji jest taki sam i mozna go odzyskowac przez zlamanie oryginalnego 8-bajtowego wezwania. Problem jest wiekszy w sytuacji, gdy serwer musi udostepniac uslugi logowania dla Windows 95 lub starszych klientow, które uzywaly sekwencji tworzenia klucza LAN Manager.

Tworzenie klucza LAN Manager

W technice LAN Manager hasla maja dlugosc 8 bajtów i nie sa rozróżniane male i duze litery. Do wygenerowania klucza sesji system uzywa hasla skonwertowanego do postaci duzych liter. Haslo zostaje nastepnie wypełnione spacjami do 14 bitów i kolejnosc bitów w kazdym bajcie zostaje odwrócona. Uzyskany czternastobajtowy lancuch przechodzi standardowy algorytm szyfrowania (DES) aby otrzymac 14-bajtowe zaszyfrowane haslo.

Na tym etapie 8-bajtowe wezwanie jest szyfrowane za pomoca pierwszych 7 bajtów zaszyfrowanego, odwróconego hasla, a nastepnie to samo wezwanie jest szyfrowane za pomoca pozostalych 7 bajtów. W wyniku otrzymujemy 16-bajtowy klucz sesji, wypełniany 5 bajtami zer, by utworzyc 21-bajtowa odpowiedz.

Wprawdzie dawniej taki typ klucza sie sprawdzal, lecz dzisiejsze systemy posiadaja moc obliczeniowa wystarczajaca, by szybko odwrócić caly proces i ustalic oryginalne haslo. Programy typu l0phtcrack moga uczynic to z latwoscia.

Tworzenie klucza NT

Powyzsza slabosc klucza LAN Managera i nieustajace wysilki w podkopywaniu bezpieczeństwa systemów operacyjnych Microsoftu doprowadzily do wydluzenia klucza i rozróżniania wielkosci liter w Windows NT. W tym systemie wersja Unicode hasla, mogaca zawierac male i duze litery, przechodzi szyfrowanie MD4, aby utworzyc 16-bajtowy klucz wypełniany nastepnie zerami do 21 bajtów. Oznacza to, ze 8-bajtowe wezwanie moze zostac teraz zaszyfrowane 16-bajtowym kluczem, co zwiększa bezpieczeństwo logowania w Windows NT — lecz pod warunkiem, iz klient bedzie móg³ wygenerowac taki klucz. Windows NT domyslnie zada obu kluczy i oba zostaja wygenerowane i wyslane. Systemy Windows NT i 2000 mozna tak skonfigurowac (w Rejestrze), by nie zdaly klucza LAN Managera, lecz opcja ta musi zostac skonfigurowana zarówno u klienta, jak i w serwerze.

Prosze zwrócić uwagę, iz po uwierzytelnieniu uzytkownika i zalogowaniu do stacji roboczej ten sam proces uwierzytelnienia musi zostac powtórzony dla kazdego serwera, z którym uzytkownik sie laczy. Kazdy serwer musi w kontrolerze domeny zatwierdzic logowanie uzytkownika do serwera za pomoca podobnego procesu.

Certyfikaty X.509

Uwierzytelnienia mozna tez dokonywac za pomoca certyfikatu. Certyfikat jest cyfrowym odpowiednikiem dowodu osobistego lub paszportu. Podobnie jak te dokumenty, sluzy do udowadniania tozsamosci uzytkownika. Obecnie stosowanym standardem certyfikatów cyfrowych jest X.509.

Certyfikaty wykorzystuja infrastrukturę klucza publicznego (PKI) jako metode uwierzytelniania i przesyłania kluczy publicznych. Gdy uzywane sa hasla — zarówno nie zaszyfrowane, jak i zaszyfrowane technika NTLM — uzytkownicy sa odpowiedzialni za bezpieczeństwo hasel. Dla niektórych uzytkowników moze to stanowic problem, a ponadto w wiekszosci przypadków oznacza, ze uzytkownik musi zalogowac sie lub przechowywac haslo w systemie — byc moze nie zaszyfrowane.

Ogólnie mówiąc, certyfikat zawiera trzy główne grupy informacji:

- ♦ klucz publiczny podmiotu,
- ♦ informacje o podmiocie,
- ♦ informacje o wydawcy klucza i jego podpis.

Dopóki ufamy wydawcy certyfikatu, wystarczy sprawdzić, czy tożsamość serwera lub użytkownika jest taka, jak w certyfikacie oraz czy certyfikat nie został odwołany. Certyfikaty wydawane są przez urzędy certyfikacji, na przykład VeriSign. Urząd certyfikacji powinien zweryfikować dane osoby, której wydaje certyfikat, i wydać go jedynie gdy wszystkie dane zostaną sprawdzone.

W rzeczywistości bardzo łatwo jest otrzymać certyfikat dla takich usług, jak poczta elektroniczna, na podstawie bardzo skromnych dowodów swojej tożsamości lub w ogóle żadnych. Jednakże w przeciwnieństwie do np. certyfikatów PGP, certyfikaty X.509 zwykle są platne. Jest to konieczne tylko wtedy, gdy używamy certyfikatu w Internecie do ogólnych zastosowań. Jeśli nie zamierzamy korzystać z certyfikatu w Internecie, możemy utworzyć własny urząd certyfikacji, instalując i konfigurując serwer certyfikatów.

Serwer certyfikatów przechowuje certyfikaty i może jedynie je składować lub należać do implementacji PKI. PKI obejmuje nie tylko magazynowanie certyfikatów, lecz również narzędzia administracyjne niezbędne do wydawania, odwoływanego, składowania i pobierania certyfikatów oraz weryfikacji certyfikatów innych organizacji. Budowanie infrastruktury PKI zaczyna się od utworzenia głównego serwera certyfikatów. Podczas tego procesu certyfikat urzędu certyfikacji zostanie utworzony dla osoby zarządzającej bezpieczeństwem.

W niektórych sieciach może znajdować się tylko jeden serwer certyfikatów; inne mogą wymagać większej liczby tych serwerów. Kolejne serwery certyfikatów są podległe wzgledem głównego i muszą zostać uwierzytelnione przez urząd certyfikacji. Serwery te będą mogły następnie wydawać certyfikaty dla serwerów i osób.

X.509 jest przypuszczalnie najpowszechniej zaakceptowanym standardem certyfikacji (możemy również stosować certyfikaty PGP). X.509 jest standardem międzynarodowego zjednoczenia telekomunikacji ITU-T (*International Telecommunications Union — Telecommunication Standardization Sector*) i w istocie stanowi część składową specyfikacji X.500 zajmującej się usługami katalogowymi. Kazdy certyfikat X.509 zawiera poniższe dane:

- ♦ *Numer wersji X.509* — identyfikuje wersję standardu X.509 dla danego certyfikatu, która decyduje, jakie informacje mogą być zawarte w certyfikacie.
- ♦ *Klucz publiczny* — klucz publiczny właściciela certyfikatu oraz identyfikator algorytmu, określający algorytm szyfrowania używany przez klucz.
- ♦ *Numer seryjny* — każdy certyfikat wydany przez dany urząd posiada unikatowy numer wersji.
- ♦ *Unikatowy identyfikator* — nazwa wyróżniająca (DN — *distinguished name*) X.500 podmiotu certyfikatu. Nazwa ta powinna być unikatowa w obrębie sieci wewnętrznej lub całego Internetu.

- ◆ *Okres waznosci* — data poczatkowa i koncowa okresu waznosci certyfikatu.
- ◆ *Nazwa wydawcy* — unikatowa nazwa urzedu certyfikacji (serwera), który podpisal dany certyfikat. Serwer musi byc zaufany; w przeciwnym razie certyfikat bylby bezuzyteczny.
- ◆ *Podpis cyfrowy* — podpis uzywa prywatnego klucza serwera certyfikatow i gwarantuje, iz certyfikat od chwili wydania nie zostal zmodyfikowany.
- ◆ *Identyfikator algorytmu podpisu* — identyfikuje algorytm uzyty do podpisania certyfikatu.

Certyfikaty X.509 staja sie obecnie bardzo popularne, nie tylko w Internecie, lecz również w sieciach wewnętrznych. Kolejnym systemem zyskujacym na popularnosci jest Kerberos.

Kerberos

Kerberos został opracowany w MIT jako otwarty protokół uwierzytelniania sieciowego. Okazał się protokołem solidnym, a co za tym idzie, powszechnie akceptowanym. Nawet system Windows 2000 Microsoftu zastosował Kerberos zamiast schematu uwierzytelniania Windows NT/LAN Manager.

Kerberos stosuje mocna kryptografia, pozwalajaca klientom udowodnic swoja tozsamosc wobec serwera (i vice versa) poprzez niezaufane polaczenie sieciowe. Po wykorzystaniu Kerberosa przez klienta i serwer do udowodnienia swojej tozsamosci, mogą one również szyfrować cała swoją komunikację, by zapewnić prywatność i integralność danych w codziennych działańach.

Jak działa uwierzytelnianie Kerberos

Kerberos opiera się na wymianie komunikatów pomiędzy klientem i serwerem. Komunikaty te są szyfrowane różnymi metodami, obejmującymi DES i 3DES (w zależności od implementacji), wykorzystanie certyfikatów X.509, a nawet wspólne hasło tajne.

Komunikacja uwierzytelniająca użytkownika zawsze odbywa się pomiędzy klientem i serwerem uwierzytelniającym. Komunikacja odbywa się też pomiędzy różnymi serwery w sieci i serwerem uwierzytelniającym. Hasło użytkownika zostaje użyte do odblokowania lokalnej kopii klucza szyfrującego lub do wygenerowania klucza szyfrującego. W obu przypadkach poziom bezpieczeństwa w sieci zależy od tego, czy użytkownicy wybierają dobre hasła i jak je chronią.

Jesli zarówno klienci, jak i serwery „rozmawiają” z serwerem uwierzytelniającym, może wydawać się, że klienci i serwery nie będą nigdy w stanie zaszyfrować danych. Jednakże po uwierzytelnieniu przez serwer uwierzytelniający klient może odwoływać się do niego za każdym razem, gdy będzie chciał skomunikować się z innym serwerem. Serwer uwierzytelniający generuje klucz sesji dla lacznosci pomiędzy klientem i serwerem oraz używa biletu, by wydać klucz serwerowi.

Serwer uwierzytelniający dokonuje tego, wydając bilet klientowi, który może następnie przedstawić bilet serwerowi. Ten może zaakceptować bilet podpisany przez serwer

uwierzytelniajacy podobnie jak certyfikat, po zweryfikowaniu podpisu. Moze to stano-wic problem, jesli bilet zostal po drodze przechwycony. I jeszcze jedno: bilet musi zo-stac wykorzystany w okreslonym czasie — w przeciwnym razie klucz sesji stanie sie niewazny. Oznacza to, iz ustawnienie poprawnego czasu w serwerach jest niezbedne do zaimplementowania Kerberosa. Bilet Kerberosa zawiera:

- ◆ klucz sesji,
- ◆ nazwe uzytkownika,
- ◆ czas wygasnienia,
- ◆ podpis cyfrowy.

Podpis serwera uwierzytelniajacego jest tworzony za pomoca zestawu kluczy znanego jedynie serwerom, a nie klientom. W ten sposob serwer uzyskuje gwarancje, iz bilet po-chodzi od serwera uwierzytelniajacego. Klient wysyla do serwera uwierzytelnienie razem z biletem, ktory otrzymal od serwera uwierzytelniajacego. Ten bilet zawiera:

- ◆ biezacy czas,
- ◆ sume kontrolna,
- ◆ opcjonalny klucz szyfrujacy.

Bilet i uwierzytelnienie sa szyfrowane za pomoca klucza sesji z biletu otrzymanego od serwera uwierzytelniajacego. Gdy dane docieraja do serwera, ten weryfikuje bilet i wy-dobywa klucz sesji, a nastepnie deszyfruje uwierzytelnienie i weryfikuje informacje. Je-sli znajdująca sie w nim suma kontrolna jest poprawna, mozemy zalozyc, ze prawdziwy klient zaszyfrowal uwierzytelnienie, poniewaz tylko on posiada klucz sesji. Nastepnie sprawdzany jest znacznik czasu w celu sprawdzenia, czy uwierzytelnienie jest „swieze”. Zazwyczaj znacznik czasu nie moze różnic sie od czasu w serwerze o wiecej niz piec minut. Jesli znacznik czasu jest swiezy, bilet zostaje przyjety, a tozsamosc klienta zosta-jе potwierdzona.

W tym momencie serwer generuje odpowiedz, zawierajaca znacznik czasu z uwierzy-telnienia i dodatkowe informacje, na przyklad nazwe serwera. Szyfruje ja i wysyla do klienta. Znacznik czasu potwierdza, iz serwer jest tym, z którym klient usilował sie po-laczyc, zas inne informacje sluzą do weryfikacji nazwy serwera. Ta procedura pozwala na wzajemne uwierzytelnienie klienta i serwera.

Caly ten proces pochłania mnóstwo pracy. Wymaga od uzytkownika wprowadzenia hasla za kazdym razem, gdy chce sie polaczyc z innym serwerem lub buforowania hasla w systemie lokalnym.

Aby uniknac wprowadzania przez uzytkownika hasla dla kazdego polaczenia lub loka-lnego zapisywania hasla, Kerberos uzywa biletu przyznajacego bilety (*ticket-granting ticket*). Gdy uzytkownik po raz pierwszy w danej sesji loguje sie i zostaje uwierzytel-niony przez serwer uwierzytelniajacy, ten zwraca bilet i klucz sesji z uslugi przyznaja-cego bilety. Bilet ten sluzy do przyznawania biletów i jest buforowany w stacji roboczej przez krótki okres, typowo przez 8 godzin. Poniewaz bilet przyznajacy bilety potwier-dza tozsamosc uzytkownika i ma krótki czas zycia, haslo mozna usunac z pamieci.

Od tego momentu proces jest podobny do opisanego powyżej z tym wyjątkiem, iż użytkownik zamiast zwracając się do serwera uwierzytelniającego, może użyć serwera przyznającego bilety. Nawet jeśli serwera mi uwierzytelniającym i przyznającym bilety jest jeden i ten sam serwer, oddzielenie procesu uwierzytelniania od procesu przyznawania biletów jest podejściem lepszym, ponieważ hasło nie jest przesyłane wielokrotnie przez sieć i nie jest zapisywane w stacji roboczej klienta.

Jak działa szyfrowanie w protokole Kerberos

Jak dotąd omówiliśmy proces uwierzytelniania w Kerberosie; zajmiemy się teraz szyfrowaniem. Uważny Czytelnik prawdopodobnie spostrzegł, że do danych zawartych w każdym bielecie należy klucz sesji. Aplikacje umiejscowione skorzystają z tego klucza, na przykład IPSec, mogą go wykorzystać do zabezpieczenia łączności pomiędzy systemami.

Koncowa uwaga o protokole Kerberos: wszystkie serwery i klienci używające pojedynczego serwera uwierzytelniającego w implementacji Kerberosa są uznawane za *obszar (realm)* Kerberosa. Protokoł ten pozwala na uwierzytelnianie pomiędzy obszarami za pomocą kluczy międzyobszarowych (Kerberos 4) lub kluczy wspólnych w strukturze hierarchicznej (Kerberos 5).

Jednoczesne stosowanie szyfrowania i uwierzytelniania

Jeśli chcemy w pełni zabezpieczyć swoją sieć, musimy stosować uwierzytelnianie i szyfrowanie jednocześnie. Ostatni punkt tego rozdziału omówią niektóre technologie, wykorzystujące równoczesnie metody szyfrowania i uwierzytelniania.

PGP

Pretty Good Privacy (PGP) jest implementacją PKI, która zdobyła powszechną aprobatę, gdyż działa — a poza tym jest darmowa. Kolejna duża zaleta PGP jest wieloplatformowość. PGP zasadniczo działa podobnie jak PKI, z kilkoma drobnymi różnicami.

PGP zawsze tworzy klucz sesji, który służy do zaszyfrowania pliku. Klucz sesji jest następnie szyfrowany kluczem publicznym odbiorcy, który może zostać wysłany otwartym tekstem lub otrzymany z publicznego centrum dystrybucji kluczy PGP, jak np. dostępne w MIT (web.mit.edu/network/pgp.html). Plik standardowo jest dodatkowo kompresowany. Wiele metod łamania szyfrów polega na znajdowaniu wzorców w zaszyfrowanej wiadomości. Kompresja redukuje te wzorce, utrudniając hakerom dostęp do informacji.

W PGP klucze są tworzone przez użytkownika oraz, jak już wspomniano, rozpowszechniane otwartym tekstem lub za pomocą publicznej usługi dystrybucyjnej. Klucz prywatny jest chroniony w komputerze użytkownika za pomocą hasła, które trzeba podawać za każdym razem, gdy użytkownik chce zaszyfrować lub odszyfrować dane. Może to prowadzić do ataku metoda posrednika (*man-in-the-middle*). W tej metodzie osoba trzecia generuje klucz wyglądający tak, jakby był prawdziwym kluczem innej osoby. Posiadacz fałszywego klucza może przechwytać transmisje za pomocą różnych narzędzi, a następnie odszyfrować dane w lokalnym systemie za pomocą klucza prywatnego skojarzonego z fałszywym kluczem publicznym. Problem ten można obejść za pomocą certyfikatów, takich jak omawiane wcześniej certyfikaty X.509.

Mnóstwo informacji o PGP i wymaganym oprogramowaniu mozna znalezc pod adresem www.pgpi.org (*The International PGP Home Page*).

SSL

Protokół *Secure Sockets Layer* (SSL) stal sie juz czescia zycia codziennego, poniewaz sluzy do zabezpieczania HTTP i innych typów serwerów internetowych. Mówiac w skrócie, SSL udostepnia metode szyfrowania danych na poziomie gniazd — to znaczy pomiedzy warstwa aplikacji i warstwa transportowa. SSL wykorzystuje PKI do bezpiecznej wymiany klucza sesji pomiedzy klientem i serwerem. W tym przypadku certyfikat sluzy zarówno do przeslania kluczy, jak i do uwierzytelnienia systemu serwera.

Proces SSL zaczyna sie od polaczenia klienta z bezpiecznym gniazdem w serwerze — na przyklad z portem 443 dla HTTPS. Serwer wysyla do klienta swój certyfikat X.509, zawierajacy nazwe serwera, dla którego certyfikat zostal wystawiony oraz publiczny klucz szyfrujacy.

Klient moze teraz zweryfikowac certyfikat. Wydawane certyfikaty sa podpisywane przez urzad certyfikacji. Klient moze albo usilowac znalezc ten urzad i uzyskac jego publiczny klucz podpisujacy, albo juz posiada dany certyfikat (wszystkie przegladarki WWW zawieraja klucze podpisujace urzedow certyfikacji dla wiekszosci popularnych urzедow, takich jak VeriSign).

Jak juz powiedzielismy, wiadomosc jest mieszana w celu uzyskania skrótu wiadomosci — w tym przypadku wiadomoscia jest certyfikat. Podpis (oryginalny skrót certyfikatu) zostaje odszyfrowany i porównany z podpisem utworzonym przed chwilą przez serwer w procesie mieszanina. Jesli oba sa takie same, certyfikat mozemy uznam za zaufany (jesli zaufany jest urzad, który go wystawil), zas nazwa serwera z certyfikatu zostaje porównana z nazwa serwera, z którym wlasnie polaczylismy sie za pomoca przegladarki.

Jesli nazwy sa takie same, klucz sesji dla tego polaczenia zostaje utworzony i zaszyfrowany za pomoca publicznego klucza piecetujacego (szyfrujacego) z certyfikatu. Klucz sesji zostaje wyslany do serwera, który deszyfuje go wlasnym prywatnym kluczem piecetujacym. Nastepnie do bezpiecznego przesypania danych pomiedzy klientem i serwerem uzywane jest szyfrowanie symetryczne.

W tym przypadku dlugosc klucza sesji bedzie podyktowana informacjami w certyfikacie, które wskazuja na mozliwosci serwera. W niektórych przypadkach serwer i klient nie beda w stanie komunikowac sie, jesli jeden z nich wymaga poziomu szyfrowania nie udostepnianego przez strone przeciwna.

IPSec

Tworzenie bezpiecznej lacznosci jest zalezne od zdolnosci programow w warstwie aplikacji do obslugi szyfrowania i uwierzytelniania. Oznacza to, iz aby uczynic z FTP lub Telnetu protokoly bezpieczne, nalezaloby napisac na nowo standardowe klienty i serwery tych uslug. W istocie, z uslug dzialajacych przez TCP/IP wiekszosc trzeba by napisac na nowo, by zawrzec w nich jakies zabezpieczenia. Prowadzi to do kolejnego problemu — standardow. Poniewaz niektóre z tych aplikacji musza ze soba współpracowac, wazne jest zastosowanie dla wszystkich takich samych, bezpiecznych mechanizmow.

Tutaj może się przydać *IP Security* (IPSec). Jeśli przypomnimy sobie stos TCP/IP, wszystkie aplikacje — po stronie klienta i serwera — mieścią się w warstwie aplikacji. Używa je różnych portów z warstwy gniazd do komunikacji z TCP (dla komunikacji połączeniowej) lub UDP (dla komunikacji bezpołącznościowej). TCP i UDP z kolei wykorzystują IP do pakowania danych i kierowania ich w różne strony przez właściwy interfejs sieciowy.

Zasadniczo IPSec odpowiedzialność za szyfrowanie i uwierzytelnianie przenosi z programów warstwy aplikacji do warstwy internetowej. Dzięki temu wszystkie dane przesypane w dół stosu TCP/IP mogą być szyfrowane bez angażowania programów warstwy aplikacji.

Jak Czytelnik zapewne się domyslił, IPSec wiąże się bardzo wygodnie z Kerberosem, ponieważ ten dokonuje uwierzytelniania użytkowników i przesyłania kluczy sesji tam i z powrotem. Połączenie IPSec i Kerberosa jest pełnym systemem zabezpieczenia — Kerberos zarządza uwierzytelnianiem użytkowników, zaś IPSec zabezpiecza dane przesypane w sieci.

IPSec składa się z dwóch różnych protokołów: AH (*Authentication Header* — nagłówek uwierzytelniający) oraz ESP (*Ecapulating Security Payload* — zabezpieczenie ładunku). Protokoły te zapewniają uwierzytelnianie pakietów — nie w taki sposób, jak Kerberos lub certyfikat X.509, lecz przez podpisywane wysyłanych pakietów i weryfikacje przy odbiorze. Protokoły te mogą również szyfrować dane przesypane w pakiecie, tak że jest on po drodze niemal niemożliwy do odczytania. W kilku następnych punktach omówimy oba protokoły oraz *Security Associations* (SA) — odpowiednik sejji w IPSec.

Authentication Header

Protokół AH zapewnia integralność danych, stosując algorytm mieszający i numery kolejne. Podobnie jak podpis cyfrowy, algorytm bezpiecznego mieszanego służy do utworzenia wartości kontrolnej integralności (ICV — *Integrity Check Value*). Wartość ta może zostać utworzona za pomocą algorytmów HMAC (*Hash Message Authentication Code*), MD5 (*Message Digest 5*) lub HMAC SHA (*Secure Hash Algorithm*).

Algorytm mieszający mieszany części nagłówka IP i porcje danych IP w pakiecie IP. Jedynie część nagłówka IP jest używana, ponieważ wartość mieszana zawarta w nagłówku nie jest znana przed obliczeniem, wobec czego nie może zostać objęta algorytmem. Do innych wykluczonych pól należą suma kontrolna nagłówka, flagi, przesunięcie fragmentu, TTL i typ usługi, ponieważ wartość każdego z nich może ulec po drodze zmianie.



Jeśli datagram musi zostać w źródle podzielony z uwagi na topografię wykorzystywanej sieci, przetwarzanie przez AH musi nastąpić przed fragmentacją datagramu. Wobec tego fragmenty muszą też zostać złożone w hostie docelowym przed przetworzeniem przez AH.

Pozostałe informacje są mieszane, a wynikowa wartość zostaje umieszczona w nagłówku IP. Oznacza to, że teraz nagłówek zawiera dodatkowe informacje:

- ◆ *Dlugosc (Length)* — dlugosc nagłówka uwierzytelniajacego.
- ◆ *Indeks parametrów zabezpieczenia (SPI — Security Parameters Index)* — identyfikuje *Security Association* dla lacznosci.
- ◆ *Numer kolejny (Sequence Number)* — prosty licznik, zwiększały o 1 dla każdego kolejnego pakietu wysłanego przez SA. Początkowa wartość wynosi 0 i nie może nigdy przejść przez 0 ponownie; wskazuje numer pakietu przesłanego przez SA podczas lacznosci. Odbiorca sprawdza to pole, by potwierdzić, czy pakiet o danym numerze dla skojarzonego SA nie został już raz odebrany. W takim przypadku pakiet jest odrzucony.
- ◆ *Dane uwierzytelniające (Authentication Data)* — dane te zawierają wartość kontrolną integralności (ICV — Integrity Check Value). Odbiorca oblicza wartość skrótu i porównuje z tą wartością, aby zweryfikować integralność przesłanych danych. W taki sposób podpis cyfrowy zapewnia integralność danych.

Numer kolejny udostępnia prostą metodę ochrony przed atakami przez odtworzenie (*replay attack*). Ten typ ataku polega na przechwyceniu sesji przez sniffera pakietów. Część danych zostaje następnie zmodyfikowana, a transmisja zostaje odtworzona ponownie — udaje transmisje z wiarygodnej stacji roboczej, by oszukać odbiorcę.

Poczas negocjacji sesji (która nosiła nazwę SA — *Security Association*) numer kolejny jest ustalany na zero. Pierwszy wysłany pakiet będzie miał numer kolejny 1, następny 2 i tak dalej. Dodatkowo numery kolejne nie mogą się powtarzać w czasie trwania SA, wobec tego po osiągnięciu maksymalnej wartości numeru kolejnego (2^{32}), musi zostać nawiązana nowa SA, dla której numery kolejne znów zaczynają się od zera.

Nagłówki uwierzytelniające mogą być używane w trybie tunelowania lub transportowym. Oba tryby działają w ten sam sposób, lecz podpisywane dane są inne. W trybie transportowym — to znaczy, w bezpośrednim połączeniu między komputerami — po nagłówku AH następuje nagłówek IP, który obejmuje statyczne pola nagłówków IP i AH oraz dane IP. W trybie tunelowania IPSec tworzy tunel pomiędzy dwoma punktami końcowymi, na przykład między routерami. W tym przypadku nagłówek AH następuje po nowym tunelowanym nagłówku IP i skrót jest licząny z pół nowego nagłówka IP, nagłówka AH i danych IP.

Ecapsulating Security Payload

Protokół ESP może być użyty do zapewnienia integralności danych w sposób podobny do AH, lecz dodatkowo zapewnia szyfrowanie danych. ESP jednakże nie rusza informacji nagłówka. Gdy protokoły ESP i AH są używane razem, wówczas ESP najpierw szyfruje dane, a następnie AH podpisuje pakiet — co zapewnia bardzo mocne zabezpieczenie. W tym przypadku będziemy mieli nagłówek IP, nagłówek AH i nagłówek ESP. ESP do podpisywania może wykorzystać te same dwa protokoły podpisujące, co AH. Do szyfrowania ESP może być użyty protokołów DES-CBC, DES 40-bitowy i 3DES.

Pola zawarte w nagłówku ESP są podobne do pól nagłówka AH:

- ◆ *Indeks parametrów zabezpieczenia (SPI — Security Parameters Index)* — identyfikuje *Security Association*.
- ◆ *Numer kolejny (Sequence Number)* — używany w ten sam sposób, jak w AH.

- ◆ *Ladunek danych* — ladunek moze ulegac zmianom w zaleznosci od trybu dzialania. W trybie transportowym obejmuje naglowek TCP lub UDP i dane; w trybie tunelowania zawiera rózniez oryginalny naglowek IP.

Poza naglowkiem ESP dodaje na koniec danych jedna lub dwie stopki. Pierwsza zawiera nastepujace pola:

- ◆ *Wypelnienie (Padding)* — sluzy do zaokraglania danych do 32 bitów lub pelnego bloku, zaleznie od uzytej metody szyfrowania.
- ◆ *Dlugosc wypelnienia (Padding Length)* — wskazuje dlugosc pola Wypelnienie, aby odpowiednia objetosc wypełniajacych danych mogla zostac odrzucona przez odbiorce.
- ◆ *Nastepny naglowek (Next Header)* — mówi odbiorcy, jaki typ danych zostal zaszyfrowany. Uzywany jest tu identyfikator protokolu IP (IP = 0, TCP = 6, UDP = 17).

Druga stopka jest uzywana jedynie wtedy, gdy ESP zosta³ skonfigurowany rózniez do podpisywania przesypanych danych. Stopka ta zawiera tylko jedno pole — dane uwierzytelniajace (*Authentication Data*). Podobnie jak w protokole AH, jest to wartosc ICV obliczona z naglowka ESP, ladunku i stopki ESP.

ESP, podobnie jak AH, moze dzialac w dwóch trybach, lecz różnica lezy w tym, co jest podpisywane — lub w tym przypadku szyfrowane. W trybie transportowym, w którym dwa hosty komunikuja sie bezpo¶rednio, podpisywane sa naglowek ESP, naglowek i dane transportowe oraz stopka ESP. Dane sa dodatkowo szyfrowane. W trybie transportowym szyfrowanie danych przebiega nastepujaco:

1. Naglowek i dane transportowe (UDP lub TCP) sa pakowane w ladunek danych.
2. W razie potrzeby dodawane jest odpowiednie wypelnienie. Wymogi wypelnienia zaleza od algorytmu szyfrujacego.
3. Ladunek danych i pola stopki (wypelnienie, dlugosc wypelnienia i nastepny naglowek) sa szyfrowane.

Jesli wybrane zostały równoczesnie szyfrowanie i uwierzytelnianie, szyfrowanie odbywa sie najpierw.

W trybie tunelowania ESP podpisuje naglowek ESP, oryginalny naglowek IP, naglowek i dane transportowe oraz stopke ESP. Wedlug tych samych trzech kroków odbywa sie szyfrowanie; różnica polega na zaszyfrowaniu calego oryginalnego datagramu IP.

Security Association

W wiekszosci przypadkow dwie stacje musza utworzyc sesje, zanim beda mogly komunikowac sie ze soba. W IPSec sesja nosi nazwe kojarzenia zabezpieczen (*Security Association*). SA definiuje wspólne ustawienia zabezpieczen i klucze uzyte do ochrony lacnosci pomiedzy punktami koncowymi. Oczywiscie moze istniec wiele SA dla komputera komunikujacego sie równoczesnie z wieloma innymi i uzywajacego IPSec. W taki⁹ sytuacji odbiorca uzywa pola SPI (*Security Parameters Index*), aby skojarzyc pakiet z właściwym SA, a co za tym idzie, z właściwymi kluczami szyfrujacymi.

Do negocjowania ustawien zabezpieczeń, które stana sie SA, sluzy protokół ISAKMP (*Internet Security Association and Key Management Protocol*). W komputerze nadajacym IPSec SA jest składowany w bazie danych, która kojarzy SA, ID użytkownika i adres docelowy. Po stronie odbiorcy te same informacje sa zapisywane w podobnej bazie danych. Dodatkowo odbiorca tworzy wzajemny SA. SPI sluzy do kojarzenia nadchodzących pakietów z właściwym SA. Podczas negocjacji system docelowy tworzy SPI i przesyła do nadawcy. System dolacz SPI do kazdego nagłówka protokołów AH i ESP. Odbiorca uzywa skrótu z SPI i adresu docelowego, aby szybko znalezć odpowiedni SA w bazie danych SA.

Do tworzenia SA pomiędzy dwoma komputerami organizacja IETF opracowała standardowa metode kojarzenia zabezpieczeń i wymian kluczy, która laczy ISAKMP i generowanie klucza Oakley. ISAKMP definiuje ogólne procedury i formaty komunikatów, których nalezy uzyc podczas tworzenia, utrzymania, modyfikacji i usuwania SA.

Konkretnym protokołem, którego ISAKMP uzywa do tego celu, jest *Oakley Key Determination Protocol* (protokół ustalenia klucza Oakley). Zanim IPSec zacznie przetwarzac pakiety, musza odbyc sie dwie negocjacje. Oakley generuje i zarządza uwierzytelnicymi kluczami uzywanymi do szyfrowania i deszyfrowania informacji w obu negocjacjach. Oakley stosuje protokół wymiany kluczy Diffiego-Hellmana.

Prosze zwrócić uwagę, iz Oakley dziala w dwóch trybach. Tryb główny (*Oakley Main Mode*) zapewnia material do generowania nowych kluczy i nowy klucz szyfrujacy. Tryb szybki (*Oakley Quick Mode*) jest uzywany wtedy, gdy obie strony posiadaja juz material do generowania kluczy, lecz musi zostac wygenerowany nowy klucz szyfrujacy. Tryb szybki moze byc uzyty tylko po uzyciu trybu głównego.

Aby zapewnic udana, bezpieczna lacznosc, ISAKMP/Oakley wykonuje dwufazowa operacje. W kazdej z faz poufnosc i uwierzytelnienie sa wynikiem wykorzystania wynegocjowanych algorytmów szyfrowania i uwierzytelniania, uzgodnionych przez dwa równorzędne hosty ISAKMP. A lgorytmy te mozna konfigurowac.

Pierwsza negocjacja obejmuje uwierzytelnienie tozsamosci obu hostów chęciowych komunikowac sie ze sobą oraz wymiane kluczy sesji do zabezpieczenia danych. Pierwsza negocjacja zarządza ISAKMP.

Druga negocjacja nastepuje po wymianie kluczy. Oba hosty musza uzgodnic ustawienia zabezpieczeń, których beda uzywac do zabezpieczenia lacznosci IP. Zasady definiujące reguły tej negocjacji (na przykład, jakie algorytmy sa dopuszczalne) noszą nazwę *zasad IPSec (IPSec policy)*.

Pierwsza negocjacja tworzy bezpieczny kanal komunikacyjny pomiędzy dwoma komputerami, tzw. ISAKMP SA. Aby utworzyc bezpieczny kanal, ISAKMP uwierzytelnia tozsamosci komputerów i wymienia informacje, aby uzyskac wspólny klucz tajny. Tryb główny Oakley zapewnia niezbedna ochronę tozsamosci podczas tej wymiany. Zapewnienia to całkowita prywatność, ponieważ pomiędzy komunikującymi się hostami nie są przesyłane bez szyfrowania żadne informacje o tozsamosci.

W drugiej negocjacji zostaje nawiazane skojarzenie zabezpieczeń (SA) pomiędzy dwoma komputerami. Informacje o SA są przekazywane do sterownika IPSec (łącznie z kluczem wspólnym) w obu komputerach, nadawcy i odbiorcy. Podczas tej negocjacji w razie potrzeby materiał kluczy jest odswieżany lub tworzony są nowe klucze.

Rozdział 23.

Rozwiązywanie problemów z siecią i łącznością

W tym rozdziale:

- ◆ Wprowadzenie
- ◆ Sprawdzanie konfiguracji IP
- ◆ Sprawdzanie łączności
- ◆ Sprawdzanie rozwiązywania nazw
- ◆ Sprawdzanie klienta i serwera

W świecie idealnym wszystko działa przez cały czas. Zyjemy jednak w świecie rzeczywistym, w którym urządzenia się psują, a błędy ludzkie są przyczyną wypadków. W obliczu tych niedoskonałości naszym jedynym wyjściem jest lokalizacja źródła problemów i znalezienie środków zaradczych.

Mówiąc prosto, rozwiązywanie problemów (*troubleshooting*) to proces ustalania, dla którego określona procedura lub produkt nie działa. Istnieją dwie metody rozwiązywania problemów: pierwsza polega na znalezieniu problemu z nowym produktem lub procesem, którego nie mamy uruchomic; druga na ustaleniu, dlaczego cos, co działało, przestało działać. Typ wykonywanej pracy — instalatora, administratora sieci lub programisty — decyduje o metodach rozwiązywania problemów. W większości przypadków, aby zmusić cos do działania po raz pierwszy, wystarczy przeczytać podręcznik. Może zaskakiwać fakt, iż tylko około 4 procent ludzi pracujących w dziedzinach technicznych czyta podręcznik, zanim spróbuje zainstalować nowy produkt.

Ogólnie mówiąc, pomyslnie rozwiązywanie problemu wymaga dokładnego zrozumienia systemu, w którym rozwiązywamy problemy; nieważne, czy jest to tak prosty system jak np. brak połączenia w komputerze, czy skomplikowany — na przykład nowa siedmiopoziomowa aplikacja, która zawodzi w określonej funkcji. Jeśli wiemy, jakie kroki aplikacja wykonuje i jak są one powiązane ze sobą, znalezienie problemu polega po prostu na przesledzeniu po kolejnych kroków oraz identyfikacji (zwykle przez obserwację), gdzie procedura ulega przerwaniu.

W przypadku TCP/IP proces rozwiązywania problemów jest taki sam. TCP/IP jest systemem składników programowych i sprzętowych, a każdy z nich musi funkcjonować.

poprawnie, aby cały proces zadziałał. Na początek zajmiemy się podstawowym procesem rozwiązywania problemów.

Proces rozwiązywania problemów

Połączenie dwóch komputerów może wydawać się prosta sprawa, lecz to tylko pozory — łączność wymaga bezawaryjnego działania wielu elementów, a także wykonania przez administratora szeregu konkretnych operacji. Na przykład, jeśli próbujemy połączyć się z witryna WWW innej firmy, nasz komputer musi posiadać adres IP, co oznacza, iż nasz serwer DHCP musi działać poprawnie. Aby pakiety DHCP dostarczyć do naszego komputera, routery muszą je przekazywać, agent przekazujący DHCP musi działać lub komputer musi być podłączony do tej samej podsieci co serwer DHCP. Dysponując już adresem, musimy dostarczyć się z lokalnej podsieci do bramy zewnętrznej, co oznacza, iż nasz lokalny ruter i wszystkie routery po drodze muszą działać, oraz iż protokół wyboru tras używany w naszej sieci musi funkcjonować. Dochodząc do bramy zewnętrznej, możemy przechodzić przez serwer proxy, który musi działać, oraz być może przez zapory firewall. Zapora może być powiązana z systemami uwierzytelniającymi, które również muszą działać.

Gdy zadanie opuszcza ją nasza sieć przez zapory, jesteśmy na lasce i niewidoczni w Internecie. Ta część podróży obejmuje ruter, który łączy naszą firmę z ISP, sieć wewnętrzna ISP oraz wewnętrzna sieć i routery docelowego ISP, które wszystkie muszą działać. Laczne pomiędzy dostawcami usług internetowych istnieje dzięki temu, iż dostawca usług lub operator telefonii na wyższym poziomie posiada oprócz ruterów i wewnętrznych sieci działające linie lub satelity. Wymaga to również funkcjonujących protokołów wyboru tras we wszystkich trzech systemach, zapewniających znalezienie ruterów od punktu A do punktu B. W sieci mieszczącej serwer WWW, z którym chcemy się połączyć, muszą działać: zapora, wewnętrzne trasowanie, koncentrator lub przełącznik, zdalny komputer serwera WWW oraz oprogramowanie serwera WWW. Ponieważ wiele witryn obecnie jest sterowanych danymi, serwer bazy danych na odległym końcu również musi działać.

Idąc trochę dalej w scenariuszu z dwóch poprzednich akapitów, wszystkie systemy muszą komunikować się na podstawowym poziomie, co oznacza, iż wszystkie warstwy stosu TCP/IP każdego systemu muszą funkcjonować poprawnie. Warstwa aplikacji musi bezpośrednio „rozmawiać” z portami, które powinny być w stanie przesłać dane do TCP lub UDP bez zniekształceń. Protokole te wykorzystują IP, który polega na ARP i musi współpracować z ICMP i IGMP. I, oczywiście, karty sieciowe muszą być sprawne.

Ostatnie trzy akapity zawierają przegląd procesów, które zachodzą za każdym razem, gdy odwiedzamy stronę WWW. Jak widać, komunikacja przez TCP/IP nie jest ani prosta, ani wyjątkowo wydajna, aczkolwiek wydaje się działać całkiem nieźle. „Tysiącmilowa podróż zaczyna się od pierwszego kroku”, tak też odbywa się proces rozwiązywania problemów. W naszym przypadku pierwszym krokiem jest system lokalny. Zaczniemy od niego.

Sprawdzenie konfiguracji IP

Jednym z najbardziej oczywistych miejsc, od których można zacząć rozwiązywanie problemu, jest konfiguracja TCP/IP. Obejmuje ona adres IP i maskę podsieci, określające tożsamość systemu. Konfiguracja zawiera również adres bramy, potrzebny do przekazywania pakietów do innych sieci, wskazniki do serwerów DNS i WINS, niezbędnych do rozwiązywania nazw, oraz adresy portów, potrzebne przyłączeniu się z systemem zdalnym. Proszę pamiętać, iż działający system zwykle działa dopóty, dopóki nie „zadziała” na niego użytkownik; zas użytkownicy czasem zmieniają konfiguracje.

Kontrola konfiguracji IP dla Microsoft Windows

Sposób sprawdzania konfiguracji IP zależy od określonego systemu operacyjnego (Microsoftu lub Unix i jego pochodne), którego używamy. Ponieważ wiele komputerów biurowych zawiera systemy operacyjne Microsoftu, zaczniemy od nich.

Narzędzie ipconfig

Systemy Windows 95 i Windows 98 zawierają narzędzie graficzne o nazwie WINIPCFG, które pozwala skontrolować konfigurację IP. Narzędzie to można uruchomić z menu *Start* polecienniem *Uruchom*. W zestawie narzędzi dla Windows 95 i 98 znajdziemy również kopię ipconfig — narzędzia uruchamianego z wiersza poleceń, które daje takie same informacje, jak WINOPCFG. Ponieważ narzędzie ipconfig jest dostępne na wszystkich platformach Windows, przyjrzymy się bardziej szczegółowo.

Program ipconfig pozwala przejrzeć konfigurację systemu. Wyjście tego programu wygląda następująco:

```
Windows 98 - konfiguracja IP

0 Ethernet karta :

    Adres IP. . . . . : 192.168.0.1
    Maska podsieci . . . . . : 255.255.255.0
    Domyslna brama . . . . . :

1 Ethernet karta :

    Adres IP. . . . . : 48.53.66.7
    Maska podsieci . . . . . : 255.255.192.0
    Domyslna brama . . . . . : 48.53.66.1
```

Z tego możemy szybko sprawdzić adres IP, maskę podsieci i bramę domyślną. W powyższym przykładzie adres IP i maska podsieci są poprawne, zas domyślna brama jest dla systemu zdefiniowana.

W nowszych wersjach Windows jest kilka określonych adresów, na które należy zwrócić uwagę. W pierwszej kolejności spójrzmy na dowolny system z adresem z zakresu 169.145.x.x. Ten zakres jest używany do automatycznej konfiguracji IP klientów DHCP. Gdy klient próbuje bez powodzenia otrzymać adres IP od serwera DHCP, wówczas systemy Windows 2000 i Windows Me przydzielają adres z tego zakresu. Mogliby to oczywiście prowadzić do powielania adresów IP, więc Microsoft wprowadził kontrolę

Jeżeli adresu przed przydzieleniem przez pingowanie, aby uniknąć podwójnych adresów IP. Gdy system przechodzi na automatyczne adresowanie IP, nie informuje o tym użytkownika. Zazwyczaj użytkownik dowiaduje się o problemie, gdy nie może się zalogować, a kończy się to telefonem do pomocy technicznej.

Powinnismy tez zwracac uwage na adres 192.168.0.1. Jedna z nowych mozliwosci Windows 2000 jest udostepnianie polaczenia, ktore pozwala na posiadanie jednego polaczenia poprzez dostawce uslug (na przyklad z Internetem) oraz drugiego z siecia bialkalna; w ten sposob przez klikniecie przycisku nasz komputer staje sie prostym ruterem pomiedzy laczem do dostawcy uslug i wewnętrzna siecia 192.168.0.0. Uzytkownik eksperymentujacy w systemie moze zignorowac pojawiajace sie duze ostrzezenie i spróbowac udostepniania polaczen. I nagle jedna z kart sieciowych otrzymuje niewlasciwy adres, a system przestaje sie komunikowac.

Zarówno automatyczne adresowanie IP, jak i udostępnianie połączeń to część planu Microsoftu, aby ułatwić tworzenie i konfigurację sieci użytkownikom domowym i niewielkim biurom przy minimum wiedzy technicznej. Wyłącznik tych funkcji jest jednakże trudny do znalezienia i trudno nim operować.

Rozwiązywanie problemów z DHCP za pomocą ipconfig

Microsoft promuje protokół DHCP od wielu lat. Jednakże DHCP nie jest nieomylny i kilka problemów, które stwarza możemy naprawić z pomocą polecenia ipconfig.

Automatyczne adresowanie IP, omówione przed chwilą, jest przyczyną podstawowego problemu — host albo otrzymał niewłaściwe informacje, albo nie otrzymał żadnych. Aby sprawdzić status DHCP w adapterze, możemy użyć polecenia `ipconfig /all`, którego wynik przedstawiлиśmy poniżej.

Konfiguracja IP systemu Windows NT

```
Nazwa hosta . . . . . : hydra.scrimtech.com
Serwery DNS . . . . . : 207.236.145.41
                           207.236.145.40
Typ wezla . . . . . : Hybryda
ID zakresu NetBIOS. . . . . :
Routing IP wlaczony . . . . . : Tak
WINS Proxy wlaczzone . . . . . : Nie
NetBIOS Resolution uzywa DNS. : Tak
```

Karta Ethernet 0

```
Opis. . . . . : NDIS 5.0 driver
Adres fizyczny. . . . . : 00-E0-18-C4-1A-56
DHCP wlaczzone . . . . . : Nie
Adres IP. . . . . : 192.168.0.1
Maska podsieci. . . . . : 255.255.255.0
Domyslna brama. . . . . :
Podstawowy serwer WINS. . . . . :
Zapasowy serwer WINS. . . . . :
Dzierzawa otrzymana . . . . . :
Dzierzawa wygasza . . . . . :
```

dated January 1.

```

Adres fizyczny. . . . . : 00-40-05-5B-C6-A5
DHCP włączone . . . . . : Nie
Adres IP. . . . . . . . . : 24.112.82.45
Maska podsieci. . . . . : 255.255.252.0
Domyslna brama. . . . . : 24.112.92.1
Podstawowy serwer WINS. . . . . :
Zapasowy serwer WINS. . . . . :
Dzierzawa otrzymana . . . . . :
Dzierzawa wygasła. . . . . :

```

W tym przypadku musimy sprawdzić wpis *DHCP włączone* i upewnić się, czy jest poprawny. Jeśli host ma mieć DHCP włączone, a nie ma, przypuszczalnie konfiguracja IP jest błędna. Nawet jeśli klient jest poprawnie skonfigurowany, system może nie funkcjonować właściwie, więc być może trzeba będzie usunąć istniejące informacje i spróbować ponownie.

Mozemy wydać polecenie `ipconfig /release`, aby wyczyścić informacje, a następnie ponowić próbę uzyskania przez system nowego adresu polecienniem `ipconfig /renew`. Jeśli czynności te zostaną zakończone pomyślnie, komputer będzie gotów do komunikacji. Jeśli nie, należy ręcznie sprawdzić konfigurację IP, ponieważ wszelkie wartości skonfigurowane lokalnie będą blokować wartości z serwera DHCP. Jeżeli nie jesteśmy w stanie uzyskać adresu DHCP przy próbie odnowienia, problem nie leży w konfiguracji, lecz w podstawowej łączności.

Kontrola konfiguracji IP w systemach uniksowych

Podobnie jak środowisko Windows, środowisko uniksowe zawiera interfejsy graficzne. Niektóre z tych narzędzi pozwalały sprawdzić konfiguracje IP, a niektóre nie. Niniejszy punkt zajmuje się polecienniem `ifconfig`, które jest podobne do `ipconfig`, lecz ma większe możliwości.

Podstawowe polecenie `ifconfig` zwraca wiele danych konfiguracyjnych. Na przykład:

```

[root@www3 /etc]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:90:27:DC:89:5D
          inet addr: 48.53.66.9 Bcast:24.255.255.255
          Mask:255.255.192.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1185636154 errors:38 dropped:0 overruns:0 frame:0
          TX packets:1379083270 errors:7 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:19 Base address:0xb000

```

Ponownie chcemy sprawdzić podstawowy adres IP i maskę podsieci. Chcemy też upewnić się, czy interfejs jest złączony (UP). Jak widać, w tym miejscu nie można sprawdzić bramy domyślnej. Musimy użyć polecenia `route`:

```

[root@www3 /etc]# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use
Iface

```

```

48.53.64.0      *
eth0              *
127.0.0.0        *
lo               default
default          48.53.64.1    0.0.0.0
eth0

```

Musimy znalezc wpis dla trasy domyslnej (prosze zwróćic uwage na litere G w kolumnie *Flags*, która wskazuje na brame — *Gateway*). W tym przypadku domyslny wpis znajduje sie w ostatnim wierszu i wskazuje na wlasciwy adres. W niektórych wersjach i pochodnych trasa domyslna moze byc pokazana jako 0.0.0.0 z maska podsieci 0.0.0.0 lub bedziemy musieli wydac polecenie `route print` zamiast `route`. Jesli powyzsze wartosci nie beda poprawne, trzeba bedzie je skorygowac.

Zakladajac, iz konfiguracja jest w porzadku, przechodzimy do nastepnego kroku — sprawdzenia lacznosci. Nalezy zaczac od upewnienia sie, czy kabel sieciowy jest podlaczony zarowno do gniazda w scianie, jak i do komputera.

Testowanie lacznosci

Sprawdzajac lacznosc probujemy po prostu ustalic, czy komputer jest w stanie nadawac w sieci. Najpierw upewnijmy sie, czy swieca sie diody LED na karcie sieciowej (wielkszosc kart sieciowych posiada wbudowane diody swiecace). Nalezy również sprawdzic, czy sygnalizator kolizji nie swieci sie w sposob ciagly. Jesli tak, moze my miedz powazniejsze problemy, na przyklad z nasyceniem sieci.

Podzielmy na kilka punktow funkcje wymagane od systemu lokalnego, aby móg³ z powodzeniem komunikowac sie z innymi hostami:

- ♦ Maska podsieci, adres IP i skonfigurowana brama musza byc poprawne dla danego segmentu sieci.
- ♦ Aplikacja musi znac wlasciwy port i byc w stanie skomunikowac sie z warstwa gniazdz.
- ♦ Warstwa gniazd musi byc w stanie przekazywac dane do protokolow transportowych: TCP i UDP.
- ♦ TCP i UDP musza umiec opakowac informacje i komunikowac sie z IP.
- ♦ IP musi byc w stanie okreslic, czy pakiet przeznaczony jest dla sieci lokalnej, czy zdalnej; w drugim przypadku IP musi posiadac trasę do danej sieci.
- ♦ IP musi umiec wykorzystac protokól ARP, aby ustalic adres karty sieciowej dla nastepnego hopu.
- ♦ IP musi byc w stanie przekazac dane do warstwy interfejsu sieciowego; ta z kolei musi byc w stanie wykorzystac warstwe fizyczna do wyslania danych w sieci.
- ♦ Warstwa fizyczna musi byc polaczona z siecią.

Najlepszym i najszybszym sposobem przetestowania wiekszosci z powyzszych funkcji jest polecenie `ping`. Ping jest prostym narzędziem, wysylajacym pakiet z jednego sys-

temu do drugiego i czekającym na odpowiedź. W większości przypadków sprawdzamy łączność polecienniem ping, zanim nawet skontrolujemy adres IP. Tabela 23.1 wymienia miejsca docelowe, które możemy skontrolować tym polecienniem. Kazde z nich wykluca inny problem (zakładamy, że lokalny system posiada adres 48.53.66.7 z maską podsięci 255.255.192.0).

Tabela 23.1. Wykorzystanie polecienia ping do testowania TCP/IP

Adres docelowy	Co zostaje sprawdzone:
ping 127.0.0.1 (adres petli zwrotnej)	Czy warstwa gniazd działa i może komunikować się z warstwą transportową. Sprawdza, czy TCP i UDP mogą komunikować się z warstwą IP i czy warstwa IP jest w stanie odczytać tablice tras.
ping 48.53.66.7 (adres systemu lokalnego)	Czy warstwa IP jest w stanie odczytać tablice tras, oraz czy warstwa interfejsu sieciowego prawidłowo zarejestrowała lokalny adres IP w warstwie IP.
ping 48.53.66.9 (adres sąsiedniego systemu w tej samej podsieci)	Czy ARP jest w stanie rozwiązać adresy, i czy warstwa IP może wysyłać i odbierać dane przez warstwy interfejsu sieciowego i fizycznego. Upewnia się dodatkowo, czy adres IP jest poprawny dla danej podsieci.
ping 48.53.120.1 (adres dalszego systemu w tej samej podsieci)	Czy maska podsieci nie jest zbyt restrykcyjna, co powoduje rozpoznawanie przez IP hostów lokalnych jako zdalnych.
ping 48.53.180.22 (adres systemu zdalnego)	Czy maska podsieci nie jest zbyt ogólna, co powoduje rozpoznawanie przez IP hostów zdalnych jako lokalnych. Dodatkowo sprawdza, czy tabela tras jest w użyciu i czy jesteśmy w stanie połączyć się z lokalną bramą, a przy okazji —czy adres bramy jest poprawny.

Większość możliwych problemów, wymienionych wcześniej, można z łatwością sprawdzić za pomocą polecenia ping. Wyjątkiem jest to, czy usługi używają właściwych portów. Możemy skontrolować to dosyć łatwo, sprawdzając plik usług (w katalogu Windows dla Windows 9x i Me, w katalogu Winnt\system32\drivers\etc dla NT i 2000, lub w katalogu /etc dla większości implementacji Uniksa).

Jeśli mamy problem z określona usługą, z której użytkownicy chcą się połączyć, warto sprawdzić plik usług (oprócz innych podstawowych faktów, np. czy usługa jest uruchomiona). Poniższy listing przedstawia początek pliku usług.

```

echo          7/tcp
echo          7/udp
discard      9/tcp    sink null
discard      9/udp    sink null
systat       11/tcp
systat       11/tcp    users
daytime      13/tcp
daytime      13/udp
netstat      15/tcp
qotd         17/tcp    quote
qotd         17/udp    quote
chargen     19/tcp    ttypst source
chargen     19/udp    ttypst source

```

```

ftp-data          20/tcp
ftp              21/tcp
telnet           23/tcp
smtp             25/tcp      mail
time             37/tcp      timserver
time             37/udp      timserver
rlp              39/udp      resource    # resource location
name             42/tcp      nameserver
name             42/udp      nameserver
whois            43/tcp      nickname   # usually to sri-nic
domain           53/tcp      nameserver # name-domain server
domain           53/udp      nameserver
nameserver       53/tcp      domain     # name-domain server
nameserver       53/udp      domain     # name-domain server
ntp               57/tcp      # deprecated
bootp            67/udp      # boot program server
tftp              69/udp
rje              77/tcp      netrjs
finger            79/tcp
link              87/tcp      ttylink
supdup            95/tcp
hostnames        101/tcp     hostname   # usually from sri-nic
iso-tsap          102/tcp
dictionary        103/tcp     webster
x400              103/tcp      # ISO Mail
x400-snd          104/tcp
csnet-ns          105/tcp
pop               109/tcp     postoffice
pop2              109/tcp      # Post Office
pop3              110/tcp     postoffice

```

Format pliku uslug jest bardzo prosty — w kazdym wierszu po kolej protokol, port i protokol transportowy, a w niektórych przypadkach alias i ewentualny komentarz (po znaku # — na przyklad „# ISO Mail”). Pierwsze 1024 porty sa pod kontrola IANA (*Internet Assigned Numbers Authority*) i sa wspolne dla wszystkich platform. Po pierwszych 1024 portach nastepne numery sa określone dla aplikacji, która decyduje zmusić do pracy.

Powinnismy również zdawać sobie sprawę, iż porty powyżej numeru 1023 mogą być używane przez klienty łączące się z serwerem. Oprogramowanie klienta zwykle zajmuje dynamiczny port powyżej 1023. Mozemy sprawdzić aktywne połączenia polecienniem netstat. Wynik będzie wyglądać mniej więcej tak:

Aktywne połączenia

Protokół	Adres lokalny	Obcy adres	Stan
TCP	MEDUSA:1034	cyclops:ms-sql-s	ESTABLISHED
TCP	MEDUSA:ftp	nic-31-c26-199.mm.mediaone.net:3361	
CLOSE_WAIT			
TCP	MEDUSA:ftp	as2-5-7.dro.hs.bonet.se:1414	CLOSE_WAIT
TCP	MEDUSA:ftp	pD953868C.dip.t-dialin.net:3704	
CLOSE_WAIT			
TCP	MEDUSA:3389	cr32507-b.rchrd1.on.wave.home.com:2766	
ESTABLISHED			
TCP	MEDUSA:ftp	h24-70-96-200.cg.shawcable.net:61186	
ESTABLISHED			
TCP	MEDUSA:http	proxyl-external.hnsn1.on.home.com:3105	
ESTABLISHED			
TCP	MEDUSA:http	proxyl-external.hnsn1.on.home.com:11891	
ESTABLISHED			

```

TCP      MEDUSA:http          1Cust247.tnt4.krk1.da.uu.net:1789
ESTABLISHED
TCP      MEDUSA:http          1Cust247.tnt4.krk1.da.uu.net:1790
ESTABLISHED
TCP      MEDUSA:http          1Cust247.tnt4.krk1.da.uu.net:1791
ESTABLISHED
TCP      MEDUSA:http          1Cust247.tnt4.krk1.da.uu.net:1792
ESTABLISHED
TCP      MEDUSA:http          frasier.ford.com:25643  ESTABLISHED
TCP      MEDUSA:http          frasier.ford.com:26175  ESTABLISHED
TCP      MEDUSA:http          161.184.2.194:2284   ESTABLISHED
TCP      MEDUSA:http          161.184.2.194:2285   ESTABLISHED
TCP      MEDUSA:http          206.191.84.251:1843  ESTABLISHED
TCP      MEDUSA:http          206.191.84.251:2295  ESTABLISHED
TCP      MEDUSA:http          ppp-207-193-12-179.hstntx.swbell.net:1843
ESTABLISHED

```

Mozemy sprawdzic porty otwarte w systemie za pomocą polecenia netstat -a, które sluzy do wyswietlenia wszystkich portów, lacznie ze sluchajacymi. Wynik bedzie zbliżony do ponizszego:

Aktywne polaczenia

Protokół	Adres lokalny	Obcy adres	Stan
TCP	MEDUSA:ftp	MEDUSA:0	LISTENING
TCP	MEDUSA:smtp	MEDUSA:0	LISTENING
TCP	MEDUSA:domain	MEDUSA:0	LISTENING
TCP	MEDUSA:http	MEDUSA:0	LISTENING
TCP	MEDUSA:epmap	MEDUSA:0	LISTENING
TCP	MEDUSA:https	MEDUSA:0	LISTENING
TCP	MEDUSA:microsoft-ds	MEDUSA:0	LISTENING
TCP	MEDUSA:1025	MEDUSA:0	LISTENING
TCP	MEDUSA:1026	MEDUSA:0	LISTENING
TCP	MEDUSA:1029	MEDUSA:0	LISTENING
TCP	MEDUSA:1030	MEDUSA:0	LISTENING
TCP	MEDUSA:1032	MEDUSA:0	LISTENING
TCP	MEDUSA:1033	MEDUSA:0	LISTENING
TCP	MEDUSA:1034	MEDUSA:0	LISTENING
TCP	MEDUSA:3372	MEDUSA:0	LISTENING
TCP	MEDUSA:3389	MEDUSA:0	LISTENING
TCP	MEDUSA:1034	cyclops:ms-sql-s	ESTABLISHED
TCP	MEDUSA:ftp	nic-31-c26-199.mm.mediaone.net:3361	
CLOSE_WAIT			
TCP	MEDUSA:ftp	as2-5-7.dro.hs.bonet.se:1414	CLOSE_WAIT
TCP	MEDUSA:ftp	pD953868C.dip.t-dialin.net:3704	
CLOSE_WAIT			
TCP	MEDUSA:netbios-ssn	MEDUSA:0	LISTENING
TCP	MEDUSA:3389	cr32507-b.rchrd1.on.wave.home.com:2766	
ESTABLISHED			
TCP	MEDUSA:ftp	h24-70-96-200.cg.shawcable.net:61186	
ESTABLISHED			
TCP	MEDUSA:http	1Cust247.tnt4.krk1.da.uu.net:1789	
ESTABLISHED			
TCP	MEDUSA:http	1Cust247.tnt4.krk1.da.uu.net:1790	
ESTABLISHED			
TCP	MEDUSA:http	1Cust247.tnt4.krk1.da.uu.net:1791	
ESTABLISHED			
TCP	MEDUSA:http	1Cust247.tnt4.krk1.da.uu.net:1792	
ESTABLISHED			
TCP	MEDUSA:http	frasier.ford.com:25643	ESTABLISHED
TCP	MEDUSA:http	frasier.ford.com:26175	ESTABLISHED

```

TCP      MEDUSA:http          aigb35sqylue.ab.hisia.telus.net:2041
ESTABLISHED
TCP      MEDUSA:http          aigb35sqylue.ab.hisia.telus.net:2043
ESTABLISHED
TCP      MEDUSA:http          aigb35sqylue.ab.hisia.telus.net:2047
ESTABLISHED
TCP      MEDUSA:http          aigb35sqylue.ab.hisia.telus.net:2048
ESTABLISHED
TCP      MEDUSA:http          aigb35sqylue.ab.hisia.telus.net:2049
ESTABLISHED
TCP      MEDUSA:http          aigb35sqylue.ab.hisia.telus.net:knetd
ESTABLISHED
TCP      MEDUSA:http          161.184.2.194:2284      ESTABLISHED
TCP      MEDUSA:http          161.184.2.194:2285      ESTABLISHED
TCP      MEDUSA:http          ppp-207-193-12-179.hstntx.swbell.net:1843
ESTABLISHED
UDP      MEDUSA:epmap         *.*
UDP      MEDUSA:microsoft-ds  *.*
UDP      MEDUSA:1028           *.*
UDP      MEDUSA:1031           *.*
UDP      MEDUSA:3456           *.*
UDP      MEDUSA:domain        *.*
UDP      MEDUSA:1027           *.*
UDP      MEDUSA:domain        *.*
UDP      MEDUSA:netbios-ns    *.*
UDP      MEDUSA:netbios-dgm   *.*
UDP      MEDUSA:domain        *.*
UDP      MEDUSA:netbios-ns    *.*
UDP      MEDUSA:netbios-dgm   *.*
UDP      MEDUSA:domain        *.*
UDP      MEDUSA:domain        *.*
UDP      MEDUSA:domain        *.*

```

Jesli znalezlismy problem za pomoca polecenia ping, nastepnym krokiem jest ustalenie, co naprawde jest nie w porzadku. Najpierw zweryfikujmy konfiguracje hosta, przy którym pracujemy i informacje otrzymywane od DHCP (jesli host uzywa tej uslugi). Jesli konfiguracja systemu lokalnego jest prawidlowa, musimy przejsc do innych mozliwych rozwiazan.

Pierwszym posunieciem podczas rozwiazywania problemow z systemem, który uzywa interfejsu graficznego, na przyklad Windows, jest restart komputera. Graficzne systemy operacyjne zwykle wiaza podstawowe poziomy funkcjonalnosci systemu ze srodowiskiem graficznym. Maja one zla slawe z uwagi na problemy z pamiecia; czasami marnie napisana aplikacja, uruchomiona w jednym z takich systemow operacyjnych, wplynie na pamiec uzywana przez system do konfiguracji lub na inny wzorny fragment kodu. W duzym stopniu problemy takie zostaly naprawione w Windows NT, który jest wyjatkiem od powyzszej reguly.

Zakladajac, iz lokalny komputer jest w porzadku, musimy przetestowac polaczenie ze zdalnym hostem. Mozemy uzyc narzedzia traceroute do sprawdzenia trasy, która pakiety podazaja z lokalnego systemu przez rutery do hosta docelowego. Polecenie traceroute uzywa tego samego polecenia Echo Request protokolu ICMP, jak ping, lecz zaczyna od pakietu o czasie zycia (TTL) rownym 1. Zmusza to pakiet do przekroczenia limitu czasu juz w pierwszym ruterze, który zwróci do systemu komunikat ICMP o tym zdarzeniu. Wysylajac pakiet po pakiecie i zwiekszajac dla kazdego TTL o 1,

traceroute (tracert w systemie Windows) buduje liste ruterów, przez które pakiety przechodzą. traceroute w miarę postępów badania wyświetla trasę. Wynik tego polecenia wygląda mniej więcej tak:

```
Trasa sledzenia do hungryminds.com [168.215.86.100]
Σprzewyzsza maksymalna liczbe przeskoków 30
    1  <10 ms   10 ms  <10 ms  207.236.145.33
    2  *       10 ms   10 ms  10.30.235.1
    3  <10 ms   10 ms   10 ms  mtlcorr02-fe0-0-0.in.bellnexxia.net
[206.108.105.130]
    4  <10 ms   10 ms   10 ms  core1-montreal02-pos11-1.in.bellnexxia.net
[206.108.97.149]
    5  10 ms   20 ms   10 ms  Ncore2-newyork83-pos2-0.in.bellnexxia.net
[206.108.103.214]
    6  10 ms   20 ms   10 ms  bx2-newyork83-pos4-0.in.bellnexxia.net
[206.108.103.198]
    7  10 ms   20 ms   10 ms  jfk1-core1-s3-1.atlas.icix.net
[165.117.50.253]
    8  10 ms   20 ms   10 ms  jfk3-core2-pos7-0.atlas.icix.net
[165.117.48.165]
    9  10 ms   20 ms   10 ms  ord2-core2-pos5-0.atlas.icix.net
[165.117.48.38]
    10 30 ms   40 ms   30 ms  ord2-core3-pos7-0.atlas.icix.net
[165.117.48.94]
    11 30 ms   30 ms   20 ms  ord2-core4-pos5-0.atlas.icix.net
[165.117.48.98]
    12 30 ms   30 ms   30 ms  dfw3-core2-pos6-0.atlas.icix.net
[165.117.48.69]
    13 50 ms   50 ms   50 ms  dfw3-core2-pos7-0.atlas.icix.net
[165.117.48.122]
    14 91 ms   150 ms   90 ms  dfw3-core3-pos6-3.atlas.icix.net
[165.117.48.126]
    15 110 ms  181 ms  120 ms  iah2-core2-s3-0-0.atlas.icix.net
[165.117.57.81]
    16 90 ms   90 ms   90 ms  216.148.209.66
    17 226 ms  247 ms  221 ms  core-01-ge-0-3-0.chcg.twtelecom.net
[168.215.54.29]
    18 239 ms  229 ms  229 ms  core-02-ge-2-3-0.chcg.twtelecom.net
[168.215.53.6]
    17 228 ms  228 ms  238 ms  dist-01-so-2-0-0.ipkt.twtelecom.net
[168.215.53.18]
    18 234 ms  231 ms  235 ms  atagg-01-pos-0-0-0.ipkt.twtelecom.net
[207.67.94.202]
    19 233 ms  229 ms  240 ms  207-67-94-186.gen.twtelecom.net
[207.67.94.186]
    20 235 ms  231 ms  241 ms  websrv.hungryminds.com [168.215.86.100]

Sledzenie zakonczone.
```

Ten slad z serwera w Montrealu w Kanadzie do serwera WWW wydawnictwa Hungry Minds w Indianie pokazuje, że pakiety niekoniecznie wybrały najkrótszą trasę. Asterisk (*) w drugim wierszu wskazuje na przekroczenie limitu czasu. W istocie, czasami spotkamy przeskoki, dla których wszystkie trzy próby przekroczyły limit czasu. Taka sytuacja wskazuje, iż ruter w ogóle nie odpowiedział w dopuszczalnym czasie. Jeśli będzie to zachodziło raz po raz, znaleźliśmy problem — ruter nie działa.

Kolejny problem występuje wtedy, gdy dwa systemy w sieci posiadają taki sam adres IP. Nie powinno to mieć miejsca, lecz gdy się zdarzy, nasz system może rozwiązać adres docelowy raz na MAC jednego systemu, a raz na MAC drugiego.

Jesli protokol rozwiazywania adresow nie funkcjonuje prawidlowo, nie bedziemy w stanie rozwiazac adresu IP na sprzetowy. Narzedzie Address Resolution Protocol — ARP.EXE — pozwala zweryfikowac zdolosc do rozwiazywania adresow. Protokol ARP, jak pamietamy z rozdzialow 4. i 5., rozwiazuje adresy IP na adresy MAC.

Jednym przypadkiem, w którym mozemy miec problem z ARP, jest sytuacja, gdy do pamieci podrecznej ARP na potrzeby wydajnosci zostało dodane rozwiazywanie statyczne. Jesli jednak adapter sieciowy dla systemu, dla którego wprowadzono adres IP, został wymieniony, odwzorowanie moze powodowac problemy. Mozemy sprawdzic, czy ten problem wystepuje za pomoca narzedzia ARP, szukajac w tablicy ARP wpisów statycznych.

Tablica ARP zawiera odwzorowania zdalnych adresow IP na MAC. Niektore klienty lub serwery posiadaja w tablicach ARP wpisy statyczne. Jesli wpisy statyczne sa stosowane, nalezy sprawdzic rozwiazywanie adresow (a jesli odwzorowania nie sa juz wazne, powinny zostac odrzucone). Niemal we wszystkich systemach operacyjnych moze my sprawdzic tablice ARP poleceniem arp -a. Wynik bedzie wygladal mniej wiecej tak:

Interfejs: 24.112.92.45 on Interface 0x2000002	Adres internetowy	Adres Fizyczny	Typ
	24.112.92.1	00-01-4f-16-08-00	dynamiczny
	24.112.92.10	00-80-97-ea-e5-67	dynamiczny
	24.112.92.11	00-80-c6-df-07-1a	dynamiczny
	24.112.92.12	00-e0-5f-23-67-a8	dynamiczny
	24.112.92.14	00-80-e5-1f-32-0f	dynamiczny
	24.112.92.16	00-a0-78-a3-a1-ff	dynamiczny
	24.112.92.17	00-05-c6-15-87-80	dynamiczny
	24.112.92.18	00-50-21-f6-43-ec	dynamiczny
	24.112.92.19	00-e0-09-35-31-d3	dynamiczny
	24.112.92.20	00-80-4d-64-5a-45	dynamiczny
	24.112.92.21	00-80-dc-a6-0f-72	dynamiczny
	24.112.92.23	00-80-cf-1f-f8-c1	dynamiczny
	24.112.92.26	00-00-c8-12-d5-09	dynamiczny
	24.112.92.28	00-60-85-17-af-3f	dynamiczny
	24.112.92.31	00-20-29-f6-43-f9	dynamiczny
	24.112.92.32	00-80-45-d6-08-85	dynamiczny
	24.112.92.34	00-00-45-6d-28-16	dynamiczny
	24.112.92.35	00-50-56-a1-23-4f	dynamiczny
	24.112.92.36	00-e0-4c-08-f1-35	dynamiczny
	24.112.92.38	00-80-fe-05-a4-04	dynamiczny
	24.112.92.42	00-50-c6-dd-53-22	dynamiczny
	24.112.92.47	00-e0-80-45-48-3a	dynamiczny
	24.112.92.49	00-e0-01-16-0f-45	dynamiczny
	24.112.92.50	00-e0-ae-18-24-f6	dynamiczny
	24.112.92.51	00-50-04-8f-57-a2	dynamiczny
	24.112.92.53	00-03-4f-10-2e-09	dynamiczny
	24.112.92.54	00-04-05-de-f8-87	dynamiczny
	24.112.92.56	00-05-45-73-0f-20	dynamiczny
	24.112.92.58	00-50-37-16-24-2e	dynamiczny
	24.112.92.59	00-e0-27-45-68-1f	dynamiczny
	24.112.92.61	00-80-16-88-05-23	dynamiczny
	24.112.92.62	00-05-08-ed-f5-fa	dynamiczny
	24.112.92.64	00-05-1f-a1-a6-e5	dynamiczny
	24.112.92.65	00-e0-08-15-0f-1f	dynamiczny

W powyzszym listingu wszystkie systemy posiadaja adresy dynamiczne. Aby oczyiscic tablice, wystarczy zaczekac kilka minut. Wpisy w tablicy ARP zwykle szybko ulegaja przedawnieniu, aby zapewnic uzycie aktualnego adresu.

Przyjrzelismy się już podstawowej łączności; pora zająć się kolejnym ważnym problemem, z którym Czytelnik będzie miał do czynienia — rozwiązywaniem nazw.

Znajdowanie problemów z rozwiązywaniem nazw

Kluczowym obszarem w znalezaniu problemów jest rozwiązywanie nazw. Ogólnie mówiąc, mamy do czynienia z dwiema nazwami i każda z nich jest rozwiązywana w inny sposób. Pierwsza jest oczywiście nazwa hosta — albo w postaci prostej nazwy hosta, albo nazwy pełnej zlozonej (FQDN); druga jest nazwa NetBIOS dla sieci Microsoft lub sieci używających narzędzia Samba.

Znajdowanie problemów z rozwiązywaniem nazw hostów

Rozwiązywanie nazw hostów jest zaskakująco proste. W istocie większość błędów powodowanych jest przez literówki: albo w komputerze, który chce się połączyć, albo w bazie danych serwera DNS. Najszybszą metodą ustalenia, czy mamy do czynienia z problemem z rozwiązywaniem nazwy jest próba pingowania nazwy. Jeśli możemy pingować serwer według jego nazwy, wszystkie elementy składowe sieci działają poprawnie. Zwykle jest to pierwszy krok w rozwiązywaniu problemów. Jeśli nie możemy skontaktować się z serwerem pingując jego nazwę, należy wykonać kroki opisane w poprzednim podręczniku. Proste badanie serwera poleceniem ping może wyglądać następująco:

```
Badanie www.dilbert.com [65.114.4.69] z użyciem 32 bajtów danych:
```

```
Odpowiedz z 65.114.4.69: bajtów=32 czas=168ms TTL=235
Odpowiedz z 65.114.4.69: bajtów=32 czas=171ms TTL=235
Odpowiedz z 65.114.4.69: bajtów=32 czas=164ms TTL=235
Odpowiedz z 65.114.4.69: bajtów=32 czas=163ms TTL=235
```

```
Statystyka badania dla 65.114.4.69:
```

```
Pakiety: Wyslane = 4, Odebrane = 4, Utracone = 0 (0% utraconych),
Szacunkowy czas bladzenia pakietów w milisekundach:
    Minimum = 163ms, Maksimum = 171ms, Średnia = 166ms
```

W tym przypadku serwer jest czynny. Jeśli uzyskamy taka odpowiedź, jak poniżej, możemy mieć problem z rozwiązywaniem nazwy:

```
Nieznany host fred.flintstone.com.
```

W tym przypadku system daje do zrozumienia, iż nazwy serwera nie można znaleźć, wobec tego serwera nie da się pingować. Innym problemem może być rozwiązanie nazwy na błędny adres. Oto przykład:

```
Badanie www7.AlzheimerCalgary.com [207.236.154.14] z użyciem 32 bajtów danych:
```

```
Uplynął limit czasu zadania.
Uplynął limit czasu zadania.
Uplynął limit czasu zadania.
Uplynął limit czasu zadania.
```

```
Statystyka badania dla 207.236.154.14:
```

```
Pakiety: Wyslane = 4, Odebrane = 0, Utracone = 4 (100% utraconych),
```

Szacunkowy czas bladzenia pakietów w milisekundach:
Minimum = 0ms, Maksimum = 0ms, Srednia = 0ms

Mozemy podjac kilka dzialan, aby ustalic, czy problem jest lokalny, czy wystepuje w serwerze DNS. Najpierw nalezy sprawdzic plik *hosts* i upewnic sie, czy nie zawiera wpisu dla hosta, którego chcemy zlokalizowac. Standardowo plik *hosts* jest sprawdzany w pierwszej kolejnosci, aby zredukowac ruch w sieci. Plik ten mieści sie w katalogu *Windows* dla Windows 9x i Me, w katalogu *Winnt\system32\drivers\etc* dla NT i 2000, lub w katalogu */etc* dla wiekszosci implementacji Uniksa.

Jesli host jest wymieniony w pliku *hosts*, musimy sie upewnic, czy nazwa zostala zapisana prawidlowo, i czy adres IP jest poprawny. Jezeli tak — a nadal mozemy pingowac serwer uzywajac IP — problem najprawdopodobniej lezy w warstwie aplikacji albo serwer zostal przeniesiony i inny host zajal jego adres.

Przyjmujac, iz nazwa serwera, z którym chcemy sie skontaktowac, nie znajduje sie w pliku *hosts*, sprawdzamy, czy uzywamy wlasciwego adresu serwera DNS i czy jest on osiagalny. Wystarczy znalezc adres IP uzywanego serwera DNS i sprawdzic polece niem ping. W przypadku systemow operacyjnych Windows mozemy ponownie uzyc polecenia *ipconfig /all*. Informacje o serwerze DNS zostana wyswietlone w Windows 2000 w danych adapterow, zas w pozostalych systemach Windows — w konfiguracji systemu. W przypadku Uniksa i pochodnych najlepiej sprawdzic w dokumentacji (niektóre systemy przechowuja adres serwera nazw w pliku, a inne w pamieci).

Zakladajac, iz adres serwera nazw jest prawidlowy, musimy sprawdzic w serwerze DNS adres serwera, z którym chcemy sie polaczyc. Jesli uzywany przez nas serwer nazw powinien posiadac dany adres (inaczej mowiac, jesli posiada pelnomocnictwa dla strefy), mozemy tam sprawdzic, czy nazwa szukanego serwera i jego adres IP sa poprawne. Jesli tak — i nadal mozemy pingowac serwer za pomoca adresu IP — musimy szukac problemu w warstwie aplikacji. Jesli nie mozemy pingowac adresu, zapewne serwer lub czesc sieci nie dziala.



W Windows 2000 i niektórych odmianach Uniksa usługa DNS stala sie dynamiczna. Inaczej mowiac, serwery i klienty rejestruja sie same w serwerze DNS; administrator nie musi wprowadzac tych informacji recznie. Moze okazac sie konieczne wymuszenie ponownej rejestracji przez serwer odwzorowania nazwy na IP.

Jesli serwer DNS nie posiada pelnomocnictw dla strefy, funkcjonuje w rozwiazywaniu nazw jak serwer buforujacy. Inaczej mowiac, musi otrzymac adres hosta od innego serwera DNS, z naszej sieci lub z Internetu. W tym przypadku problem moze dotyczac zdalnego serwera DNS lub sieci po drodze do niego. Tutaj warto szybko upewnic sie, czy wszystkie czesci sieci funkcjonuja, az do miejsca, gdzie zadanie rozwiazania nazwy przekracza dopuszczalny czas oczekiwania. Jesli mozemy pingowac hosta (musimy znac IP) i tylko rozwiazywanie nazw nie dziala, mozemy tymczasowo dodac odwzorowanie nazwy na IP do pliku *hosts*.



Poniewaz takie statyczne odwzorowanie moze prowadzic do innych problemow, gdy zdalny system zmieni polozenie i adres IP, przypisanie adresu IP do nazwy nalezy jak najszybciej usunac.

Jesli nadal nie znalezlismy problemu, mozemy wykorzystac narzecze nslookup, które pozwala zobaczyć, jak serwer DNS rozwiązuje poszczególne nazwy. Narzecze to jest dostepne w Microsoft Windows NT i 2000 oraz wiekszosci implementacji Unixa. Nslookup moze dzialac w dwóch trybach: trybie zapytan, w którym wysylamy proste zapytanie do serwera, oraz w trybie interaktywnym, który pozwala, miedzy innymi, za pomoca standardowego polecenia ls wyswietlic informacje o zawartosci serwera. Poniżej przedstawiony zostal przyklad standardowego trybu zapytan:

```
nslookup www.GolfCanada.com
Server: localhost
Address: 127.0.0.1

Name: www.GolfCanada.com
Address: 48.53.66.7
```

Standardowe zapytania obejmują zarówno wyszukiwanie w przód, jak i wstecz. Tryb interaktywny pozwala nieco scislej zdefiniowac, jakiego typu rekordu szukamy. Na przykład, wyszukiwanie rekordu komputera wymienajacego poczta (MX — *mail exchanger*) wyglada tak:

```
> ls -t mx Scrimger.org
[cyclops.scrimtech.com]
Scrimger.org.      MX      10    mail.Scrimger.org
>
```

W wiekszosci przypadków, jesli nazwa hosta nie rozwiązuje sie, lecz posiadamy adres IP, mozemy po prostu uzywac adresu IP i nie przejmowac sie wcale nazwa. Nie jest tak w przypadku nazw NetBIOS. W swiecie TCP/IP polaczenie adresu IP i numeru gniazda lub portu sluzy do identyfikacji uslugi, ktorej chcemy uzyc. Adres IP i port razem tworza *gniazdo*, które stanowi jeden z punktów koncowych polaczenia (drugim jest adres IP i numer portu klienta). Wyobrazmy sobie próbe dostarczenia przesyłki w centrum Nowego Jorku, gdy dysponujemy jedynie ulica i numerem budynku — dochodzac do wiezowca szybko zdamy sobie sprawe, iż bez numeru apartamentu nie jessem w stanie wykonac zadania. To samo dotyczy danych przesyłanych do hosta, w którym uruchomionych jest dwadziescia lub trzydziestu usług.

Nazwa hosta nie jest az taka wzarna (o ile znamy adres IP i numer portu lub gniazdo), poniewaz TCP/IP opracowano do obslugi duzych rozproszonych sieci. W przeciwnieństwie do niego, NetBIOS został zaprojektowany do uzytku w sieci jednosegmentowej, wobec czego nie dziala w ten sposób — dla NetBIOS-u nazwa stanowi tozsamosc komputera.

Znajdowanie problemów w rozwiazywaniu nazw NetBIOS

Gdy NetBIOS byl opracowywany, nie bylo zbytniego zapotrzebowania na sieci rozlegle, poza istniejacimi rozwiazaniami mainframe. Wobec tego NetBIOS został opracowany tak, by korzystac z przyjaznej nazwy komputera w roli adresu, zamiast cegos w rodzaju adresu IP lub numeru wezla IPX. Wobec tego, aby dostac sie do wlasciwego serwera, musimy znac jego nazwe.

15-znakowa nazwa NetBIOS stanowi tozsamosc sieciowa systemu NetBIOS-owego. We wczesnych latach 80. zakladano, ze systemy te beda korzystac z NetBEUI (który został zaprojektowany specjalnie do pracy z nazwami NetBIOS) w roli protokolu trans-

portowego. Nazwy NetBIOS posiadaja szesnasty bit, który jest numerem usługi. Numer ten identyfikuje usługę, z którą łączymy się w komputerze, podobnie jak numer portu w TCP/IP. Dla każdej nazwy NetBIOS jest dostępnych tylko 256 numerów usług; jednakże pojedynczy komputer może mieć więcej niż jedna zarejestrowana nazwa NetBIOS. Dochodzimy więc do prostej prawdy, przedstawionej w tabeli 23.2.

Tabela 23.2. Porównanie TCP/IP i NetBIOS-u

Rdzenny TCP/IP	NetBIOS
Identyfikator sieci	Adres TCP/IP
Identyfikator usługi	Numer portu

Aby uzyskać łączność za pomocą rdzennego TCP/IP, musimy połączyć komputer z właściwym IP używając właściwego portu. Wobec tego w sieci NetBIOS musimy połączyć się z właściwą nazwą komputera i właściwym numerem usługi.

Jest tu jeszcze jeden problem. NetBIOS nie jest przeznaczony, z założenia, do użytku poza pojedynczym segmentem sieci i dokonuje całego rozwiązywania nazw za pomocą rozgłoszeń. Ponieważ rozgłoszenia te żałalyby sieć, gdyby routery je w ogóle przepuszczaly, routery nie przekazują rozgłoszeń. Wobec tego wszelkie funkcje nazewnicze rdzennego NetBIOS-u są ograniczone do pojedynczego segmentu, co neguje przeznaczenie TCP/IP.

Ponieważ funkcje nazewnicze NetBIOS-u są ograniczone do pojedynczego segmentu, ta usługa w sieci wielosegmentowej musi zostać zmodernizowana przez dodanie nowej funkcjonalności. Do rozwiązania tego problemu wykorzystywano kilka metod. Pierwsza było użycie prostego pliku — podobnego do pliku hostów — o nazwie *lmhosts* (*lm* od *LAN Manager*). Ten prosty plik wymieniał serwery i ich adresy IP oraz posiadał zdolność do wskazania, które z serwerów były serwerami uwierzytelniającymi (kontrolerami domeny). Drugim rozwiązaniem było utworzenie serwera, który zarządzal funkcjonalnością nazw NetBIOS.

Na początku, gdy TCP/IP i NetBIOS były używane razem, zwykle liczba serwerów była na tyle mala, iż utrzymywanie pliku nie sprawiało problemu. To jednak zaczęło się zmieniać — liczba aktualizacji pliku stała się trudna w obsłudze i administracja zaczęła być problemem, przede wszystkim dlatego, że plik *lmhosts*, podobnie jak *hosts*, musiał być zapisany w każdym komputerze klienckim.

To doprowadziło do utworzenia skoncentrowanego pliku *lmhosts*. W tym scenariuszu lokalny plik *lmhosts* w każdym klienckim wymieniał serwery uwierzytelniające i adres serwera centralnego (lub kilku), które posiadały pełną kopię pliku *lmhosts*. Dzięki temu klient mógł pobierać odwzorowania nazw z serwera centralnego w miarę potrzeb; nie musiał regularnie aktualizować swojej lokalnej kopii pliku *lmhosts*.

W koncu Microsoft, gdy wydał Windows NT, udostępnił usługę podobną do DNS-u, lecz zajmującą się nazwami NetBIOS. Usługa ta, nosząca techniczną nazwę *NetBIOS Name Server* (NBNS), w Windows NT otrzymała nazwę *Windows Internet Naming Service* (WINS). WINS pozwala na dynamiczna rejestracje nazw, jak też na odnawianie, rozwiązywanie i zwalnianie nazw.

Gdy mamy znaleziony problem z rozwiązywaniem nazw NetBIOS, możemy w większości postępować tak, jak w przypadku DNS-u. Ponownie należy sprawdzić konfigurację stacji roboczej i zweryfikować, czy adres IP serwera NBNS (WINS) jest poprawny. Możemy spróbować pingować serwer, aby sprawdzić, czy przynajmniej sam serwer działa oraz skontrolować, czy szukana nazwa jest rzeczywiście zarejestrowana w serwerze.

Jednym z kłopotów ze znajdywaniem problemów z nazwami NetBIOS jest ten, iż sposób użycia przez klienta narzędzi do rozwiązywania nazwy nie jest stały. Istnieją w istocie cztery sposoby używania narzędzi do rozwiązywania nazw przez klienta NetBIOS. Konkretna metoda, jakiej klient używa, jest definiowana przez typ wezła i może zostać sprawdzona polecienniem `ipconfig /all`. W wyniku zobaczymy wiersz, który określa typ wezła: rozgłoszeniowy, równorzędny, mieszany lub hybrydowy. Zazwyczaj będzie to wezel rozgłoszeniowy (dla komputerów nie posiadających skonfigurowanego adresu serwera WINS) lub hybrydowy (jeśli adres serwera WINS jest skonfigurowany). Typ wezła jest prawdopodobnie pierwsza rzeczą, która powinniśmy sprawdzić, jeśli podstawowe informacje wydają się poprawne.

Następnie należy sprawdzić pamięć podrzczna NetBIOS — czy nie znajdują się w niej wstępnie załadowane wpisy. W pewnych sytuacjach wpisy takie są używane do przyśpieszenia połączenia z określonym serwerem, jeśli jednak jego adres IP ulegnie zmianie, zawartość pliku `lmhosts` (z którego ładowane są informacje) musi również zostać zmodyfikowana. Możemy sprawdzić pamięć podrzczną nazw za pomocą polecenia `nbtstat -c`, które wyświetla zawartość pamięci podrzcznej. Wynik może wyglądać następująco:

```
C:\WINNT\system32>nbtstat -c

External:
Node IpAddress: [207.236.145.40] Scope Id: []

NetBIOS Remote Cache Name Table

Name          Type      Host Address   Life [sec]
-----
WEB           <1C>    GROUP        192.168.7.8   -1
MINOTAUR     <03>    UNIQUE       192.168.7.8   -1
MINOTAUR     <00>    UNIQUE       192.168.7.8   -1
MINOTAUR     <20>    UNIQUE       192.168.7.8   -1

Internal:
Node IpAddress: [192.168.1.2] Scope Id: []

NetBIOS Remote Cache Name Table

Name          Type      Host Address   Life [sec]
-----
WEB           <1C>    GROUP        192.168.7.8   -1
CYCLOPS       <03>    UNIQUE       192.168.1.1   -1
CYCLOPS       <00>    UNIQUE       192.168.1.1   -1
CYCLOPS       <20>    UNIQUE       192.168.1.1   -1
```

W tym przypadku wartości w kolumnie `Life` są ustawione na `-1`, co oznacza, iż wszystkie nazwy zostały wstępnie załadowane. Możemy spróbować usunąć polecenie `#PRE` z pliku `lmhosts` i przeładować pamięć podrzczną za pomocą polecenia `nbtstat -R`. Klient użyje serwera WINS do znalezienia serwera, którego szuka, i problem może

sie rozwiaze. Systemy operacyjne Microsoftu zawieraja plik o nazwie `lmhosts.sam`, który objasnia wszelkie opcje, które mozemy umiescic w pliku `lmhosts`.

Zakladajac, iz cala konfiguracja, która dotad sprawdzalismy, jest w porzadku, mamy jeszcze kilka nowych problemów, którym musimy stawic czolo. W DNS-ie przestrzen nazw jest zorganizowana w sposob hierarchiczny, to znaczy, istnieja rózne poziomy. Gdy chcemy rozwiazac nazwe, mozemy dojsc do poziomu głównego za pomoca FQDN i przejsc poziom po poziomie do systemu, który chcemy znalezc. Nie dotyczy to serwerów WINS lub NBNS. Zgodnie z zalozeniami, przestrzen nazw NetBIOS jest plaska, wobec czego baza danych WINS dla organizacji moze zawierac dosłownie dziesiatki tysiecy nazw, zwlaszcza ze wszystkie systemy, również klienty nie udostepniajace zadnych uslug, rejestruja swoje nazwy.

DNS uzywa hierarchicznej przestrzeni nazw, poniewaz liczba hostów, z którymi musi sie uporac, całkowicie przeciazylaby pojedynczy serwer. WINS stoi przed tym samym wyzwaniem. W przypadku uslugi WINS rozwiazaniem jest jednakze dodanie kilku serwerów równorzędnych i skonfigurowanie replikacji pomiedzy nimi. Oznacza to, ze czeescia rozwazywania problemów z WINS jest umiejetnosc rozwazywania problemów z replikacją.

Znajdowanie problemów z replikacją oznacza sprawdzanie w menedzerze WINS adresów IP dwóch zaangazowanych serwerów i zapewnienie, by przynajmniej jeden byl partnerem wypychajacym (*push*), a drugi sciagajacym (*pull*). W praktyce powinny byc jednym i drugim — o ile nie zaprojektowalismy infrastruktury WINS tak, by dzialala w konfiguracji satelitarnej.

Gdy wszystko inne zawiedzie, mozemy spróbować dodac odwzorowania nazw do pliku `lmhosts`. Jesli to rozwiaze problem, powinnismy przyjrzec sie uwaznie serwerom WINS, poniewaz moga miec uszkodzona baze danych lub nie dokonywac poprawnie replikacji.

Weryfikacja klienta i serwera

Ostatnim elementem procesu rozwazywania problemów jest kontrola wewnętrzna samego klienta i serwera. Z technicznego punktu widzenia klient jest pakietem oprogramowania uruchomionym w komputerze, zas serwer pakietem dzialajacym w serwerze, który moze byc tym samym lub innym komputerem. Jak w przypadku wszelkiego oprogramowania, istnieje mozliwosc, iz serwer, z którym chcemy sie polaczyc nie dziala, dziala nieprawidlowo lub nie jest aktualnie skonfigurowany do przyjmowania polaczen.

Ogólnie mówiac, szybkim testem sprawdzajacym w rdzennym srodowisku TCP/IP, czy usluga dziala, jest próba skontaktowania sie z inną uslugą w tym samym systemie. Jesli potrafimy polaczyc sie z Telnetem, lecz nie potrafimy z FTP, problem lezy po stronie serwera lub klienta FTP. Mozemy spróbować polaczyc sie z uslugą FTP z innej stacji roboczej lub z innym serwerem FTP z lokalnego komputera, co pozwoli ustalic, po której stronie sa problemy.

Tego samego testu mozemy uzyc z NetBIOS-em. Jesli nie potrafimy polaczyc sie z uslugą serwera, spróbujmy do systemu wyslac wiadomosc polecением `net send`. Mozemy tez uzyc polecenia `net view`, aby zobaczyć udzialy w innych systemach i

ustalic, czy klient działa. Jednym z najczęstszych problemów z NetBIOS-em są uprawnienia, więc możemy sprawdzić, czy mamy z nim do czynienia, próbując przypisać nazwę z wiersza poleceń, w którym zobaczymy komunikat o błędzie.

Proszę pamiętać, że problem niekoniecznie musi leżeć po stronie klienta — winien może być serwer. Czytelnik powinien też pamiętać, iż przypuszczalnie ma jako administrator więcej uprawnień od zwykłego użytkownika. Wobec tego po pomyslnym połączeniu warto sprawdzić uprawnienia.

Rozdział 24.

Monitorowanie sieci TCP/IP

W tym rozdziale:

- ◆ Monitorowanie sprzętu
- ◆ Narzędzia do monitorowania sieci
- ◆ Protokół SNMP
- ◆ Regulacja rozmiaru okna TCP

Czytelnik powinien być w tej chwili na etapie, na którym sieć została zainstalowana, wszystkie usługi działają poprawnie i użytkownicy są zadowoleni. Niestety, praca administratora systemów na tym się nie kończy. Zaczyna się eksploatacja i utrzymanie — czynności, które musimy podejmować, by zapewnić ciągłe funkcjonowanie sieci ze znamionowymi osiągami.

Do utrzymania sieci możemy podejść w dwojakim sposobie. Pierwszy polega na założeniu, że sieć będzie działać w oczekiwany sposób i reagowaniu na problemy na bieżąco. To podejście jest powszechnie spotykane, ponieważ administratorzy są często przeciążeni pracą. Drugie podejście obejmuje ciągłe monitorowanie sieci i zapewnianie, by serwery funkcjonowały z optymalną wydajnością. Jeśli dysponujemy wystarczającymi zasobami, ciągłe monitorowanie jest lepszym podejściem i często pozwala z wyprzedzeniem uporać się z problemami.

Administratorzy stosujący takie podejście wykorzystują dodatkowo informacje z procedur monitorujących, aby planować modernizację i zastępowanie serwerów oraz dalej dostosować sieć, dopóki każdy jej składnik nie osiągnie niemal 100-procentowej wydajności. Takie podejście wymaga jednak czasu i cierpliwości, a wiele firm nie potrafi go docenić. Jak już powiedziano, większość organizacji zajmuje się problemami dopiero wtedy, gdy wystąpią.

W wielu przypadkach monitorowanie daje bardzo dobre wyniki. W niniejszym rozdziale omówimy różnorodne narzędzia służące do monitorowania. Zaczniemy od omówienia, jakie kroki należy podjąć, aby nadzorować serwery oraz faktyczny ruch sieciowy. Ograniczymy się do narzędzi dostarczanych z różnymi systemami operacyjnymi; ponieważ dla platform Windows i Unix dostępne są dosłownie tysiące narzędzi różnych producentów, zamieszczenie ich tutaj byłoby niemożliwe.

Nastepnie omówimy funkcje protokołu SNMP (*Simple Network Management Protocol*). Rozdział konczy omówienie rozmiaru okna TCP (podstawowego narzędzia dostrajania TCP/IP) oraz maksymalnej jednostki transmisji MTU (*Maximum Transmission Unit*).

Monitorowanie sprzetu

Aby monitorowac serwer, musimy w pierwszej kolejnosci dokladnie dowiedziec sie, jakim typem serwera dysponujemy. Dla różnych serwerów wymagania sprzętowe sa różne, zas ich konfiguracje różnia sie nieznacznie. Zasoby serwera mozna podzielic na cztery podstawowe obszary wpływajace na wydajnosc. Polaczenie zasobów z tych czterech obszarów pozwoli ustalic, jak dobrze serwer spelnia swoje zadania, wobec czego monitorowanie powinno skoncentrowac sie na nastepujacych elementach:

- ◆ *Procesory* — liczba i szybkosc procesorów w systemie.
- ◆ *Pamiec* — objetosc i szybkosc fizycznej pamieci operacyjnej systemu.
- ◆ *Dyski* — typ dysków (SCSI lub IDE), ich pojemnosc, szybkosc przesyłu danych, czas dostepu i konfiguracja (macierz RAID lub jej brak).
- ◆ *Siec* — liczba kart sieciowych, ich szybkosc i sposob polaczenia z siecia (gigabitowy przelacznik, hub itp.).

Polaczenie zasobów z czterech obszarów sprzetu decyduje, jak dobrze komputer bedzie spelnial swoja role (na przyklad, serwera uwierzytelniajacego czy serwera aplikacji). Jesli stosujemy Windows 2000 i posiadamy mniej niz 256 MB RAM-u, inne elementy — dysk, siec i procesor — nie graja roli. Mozemy nawet stosowac macierz dyskowa RAID 0 + 1 (lustrzane zestawy paskowe), osiem kart sieciowych laczacych z kazdym segmentem sieci i osiem procesorów PIII Xeon 900 MHz — a system nadal nie bedzie dzialal wydajnie, poniewaz pamiec stanie sie waskim gardlem. Z drugiej strony, zastosowanie 2 GB RAM w systemie z jednym procesorem Duron 600 nie przyniesie poprawy wydajnosci w porównaniu z 512 MB RAM.

Musimy osiagnac równowage pomiedzy zasobami systemu. Przeanalizujemy teraz różne funkcje, jakie moze pelnic serwer. Prosze pamietac, ze w niektórych przypadkach serwer gra równoczesnie kilka рол; wówczas musimy zwrócić szczególna uwage na zapewnienie zasobów wystarczajacych do wszystkich zadan.

Wymogi dla serwerów uwierzytelniajacych

Proces uwierzytelniania moze obejmowac wysylanie i odbieranie informacji pomiedzy serwerem i uzytkownikiem, byc moze szyfrowanie wiadomosci, sprawdzanie podpisów przez mieszanie wiadomosci, deszyfrowanie wiadomosci i wyszukiwanie uzytkownika na liscie uzytkowników i hasel. Dodatkowo moze byc wymagane udostepnianie informacji o uzytkowniku innym serwerom w sieci (w przypadku Microsoft Networks).

Dla uwierzytelniania wymagana jest pewna moc obliczeniowa procesorów — szczególnie zdolnosc do obliczen matematycznych, poniewaz szyfrowanie i deszyfracja opiera sie na algorytmach. Dodatkowo wzarna jest zdolnosc do radzenia sobie z ruchem sieciowym dla faktycznego uwierzytelniania i udostepniania informacji innym serwerom

uwierzytelniającym. Objętość informacji o koncie użytkownika jest stosunkowo niewielka, wobec tego bazy danych kont powinny mieć rozmiary niewielkie w porównaniu z innymi bazami danych. Oznacza to, że dyski mogą nie być eksploatowane zbyt intensywnie. Ponieważ po kolejnej konfiguracji nie będzie zbyt wiele aktualizacji, większość informacji może być buforowana.

Powinniśmy upewnić się, czy serwer uwierzytelniający ma dobre wejście-wyjście sieciowe i rozsądnie wydajny procesor. Dodatkowo wymagana jest wystarczająca ilość pamięci RAM do buforowania danych kont. Dysk jest mniej ważny od pozostałych trzech obszarów, ponieważ raczej nie będzie wymagane tysiące aktualizacji na dzień.

Wymogi dla serwerów plików i drukowania

Serwer plików lub drukowania przesyła przez sieć pliki o różnych rozmiarach. Transfery plików mogą znacznie zyskać na buforowaniu: odczytach z wyprzedzeniem i zapisywaniu w tle z opóźnieniem. Tutaj ważna rolą odgrywa podsystem dyskowy, zwłaszcza w serwerze plików, ponieważ w nim będzie przechowywane faktyczne pliki. Z drugiej strony CPU nie jest nadmiernie obciążony, ponieważ przenoszeniem plików zajmuje się przede wszystkim BIOS. Trzeba będzie uporać się z kilkoma zagadnieniami bezpieczeństwa, lecz z pewnością nie na taka skalę, jak w serwerach uwierzytelniających.

Z czterech obszarów sprzętowych ważne jest połączenie sieciowe oraz podsystem dyskowy. Pamięć również będzie grała rolę, ponieważ z pewnością nie będziemy chcieli korzystać nadmiernie z pliku wymiany. Ponadto wykorzystanie pamięci do buforowania pozwoli zrównoważyć wydajność systemów sieciowego i dyskowego, gdy jeden z nich zostanie chwilowo przeciążony. Procesor nadal jest ważny, lecz w tym przypadku najmniej.

Wymogi dla serwerów aplikacji

Tutaj musimy wziąć pod uwagę, jakie typy aplikacji będą uruchamiane w danym serwerze. Na przykład, oparty na plikach system pocztowy jest serwerem plików, nie serwerem aplikacji. Z drugiej strony, aktywny system pocztowy, na przykład Microsoft Exchange lub Lotus Notes, wymaga o wiele więcej mocy obliczeniowej i pamięci (ponieważ takie są zasadniczo wymogiem w przypadku baz danych).

Większość serwerów aplikacji zalicza się do jednej z dwóch kategorii. Serwer aplikacji może być serwerem plików (na przykład w przypadku serwerów pocztowych lub WWW bez rozszerzeń, takich jak ASP lub ORBS) lub serwerem baz danych (np. serwer Microsoft Exchange lub serwer grup dyskusyjnych). Ponieważ serwery plików już omówiliśmy, przyjrzymy się wymogom dla serwera baz danych, pamiętając, że niezbędne zasoby zależą od konkretnego serwera.

Ogólnie mówiąc, serwer baz danych zawiera duże, a nawet olbrzymie pliki. Od takiego serwera często zada się przejście przez plik, znalezienia określonej porcji informacji i zwrotu wyniku do użytkownika. Wobec tego serwer baz danych musi posiadać wydajny podsystem dyskowy. Ponieważ wszelkie dane muszą zostać wczytane do pamięci RAM zanim procesor zajmie się ich obróbką, ważną rolą odgrywa podsystem pamięci. To, czy pamięć jest ważniejsza od procesora, zależy od ilości pracy, jaka serwer musi wykonać.

nac z danymi. Jesli kazda porcja danych musi zostac na dysku zaszyfrowana, procesor bedzie wazniejszy od pamieci; w przeciwnym razie RAM jest wazniejszy od procesora. W tym przypadku zapytania wysylane do serwera i odpowiedzi serwera sa zazwyczaj male w porownaniu ze zbiorem danych. Oznacza to, ze sieciowy skladnik systemu jest mniej wazny od pozostalych obszarow.

Nie istnieja rozwiazania dobre dla wszystkich. Musimy przeanalizowac okreslone wymogi danego serwera aplikacji i sieci oraz skonfigurowany sposob interakcji uzytkownika z serwerem. Jako przyklad wezmy biuro, w którym aplikacje biurkowe sa trzymane w serwerze, a nie w komputerach osobistych. W takim przypadku glownym zasobem sprzutowym dla serwerow plikow i drukowania bedzie siec, a nie dyski. Pamiec bedzie również wazniejsza od dyskow, poniewaz chcielibysmy w jak najwiekszym stopniu obslugiwac zadania z pamieci, nie odwolujac sie do dysku. Nie mozna jednak nigdy zakladac konfiguracji na podstawie ostatnio skonfigurowanego podobnego serwera aplikacji — musimy monitorowac systemy i sprawdzic, jak sprawuje sie dany system.

Narzedzia monitorujace

W trakcie projektowania sieci powinnismy wykonac kilka testow, by ustalic, jakie zasoby sprzutowe beda potrzebne. Natomiast po implementacji musimy ponownie przeprowadzic testy, aby upewnic sie, czy rzeczywista eksplotacja wyglada równie dobrze, jak poprzednie testy. Testowanie wymaga narzedzi, pomagajacych monitorowac uslugi udostepnione w serwerze. Narzedzia moga byc rózne, w zaleznosci od systemu operacyjnego. Na poczatek przyjrzymy sie narzedziu o nazwie Microsoft Performance Monitor (Monitor wydajnosci; w Windows 2000 nosi ono po prostu nazwe Performance lub Wydajnosc w wersji polskiej). Nastepnie opiszymy kilka popularnych narzedzi unikowych, ktore spełniaja te same zadania.

Monitor wydajnosci Microsoftu

Jak nazwa wskazuje, Monitor wydajnosci systemu Windows NT 4 jest przydatnym narzedziem sluzacym do monitorowania wydajnosci komputera lub uslugi. Narzedzie jest zaprojektowane do interakcji z dowolna usluga zainstalowana w systemie i moze byc uzywane bezposrednio z systemem. Monitor wydajnosci moze tez byc uzywany z zdalnego systemu, co zmniejsza wpływ na monitorowane wyniki.

Informacje dostepne w Monitorze wydajnosci moga pochodzić z zarejestrowanego pliku dziennika lub z danych biezacych. Poniewaz mozemy przeglądac dane z pliku dziennika, pozwala to rejestrowac informacje o wydajnosci przez określony czas, a później przeglądac i analizowac zgromadzone dane. W Windows NT zdefiniowanie w systemie harmonogramu rejestracji danych wydajnosci oznacza wykorzystanie polecenia AT i różnych parametrów i ustawien; w Windows 2000 zaplanowanie rejestracji danych wydajnosci przez system jest late do skonfigurowania w Dziennikach wydajnosci, a rejestracja moze nawet zostac uruchomiona warunkowo przez parametr wydajnosci. Co wiecej, informacje mozemy gromadzic z wiecej niz jednego komputera, tak by mozna było porównywac rózne komputery ze sobą. Informacje, które mozemy monitorowac, sa podzielone w logiczny sposob:

- ◆ *Komputer* — mozemy wybrac komputer przeznaczony do monitorowania.
- ◆ *Obiekt* — obszar lub aplikacja, ktora chcemy monitorowac.
- ◆ *Licznik* — po wybraniu monitorowanego obiektu zostaje wyswietlona lista wlasciwosci tego obiektu.
- ◆ *Wystapienie* — w pewnych przypadkach usluga lub aplikacja moze byc uruchomiona w komputerze w wiecej niz jednej kopii. Kazda uruchomiona kopia uslugi lub aplikacji jest uznawana za odrebre wystapienie. Mozemy wybrac monitorowanie wszystkich wystapien razem lub pojedynczego obiektu. Na przyklad, jesli komputer posiada cztery dyski, mozemy wybrac monitorowanie tylko dysku zawierajacego pliki przeznaczone do wydrukowania w serwerze drukowania.

Podczas pracy z Monitorem wydajnosci lub dowolnym innym narzedziem tego typu, nalezy pamietac, iz obciazenie systemu powodowane przez pojedyncza stacje robocza roznia sie pod wieloma wzgledami od obciazenia powodowanego przez piec lub dwadzieścia stacji roboczych. Pierwsza stacja kliencka moze zaladowac pamiec podreczna, uruchomic usluge, obudzic usluge itp. Przy monitorowaniu wzanne jest zarejestrowanie wpływu pojedynczego klienta na serwer, jak również dwóch, pieciu lub dwudziestu, aby uzyskac wyczucie zachowan obciazenia.

Prosze tez pamietac, iz nie wszyscy klienci wykorzystuja zasoby równoczesnie. Na przyklad, jesli posiadamy centrum usługowe, z którym lacz sie 150 uzytkowników z całego kraju, nie wszyscy zadaja informacji lub wysylaja dane do serwera równoczesnie — czesc czasu poswieca na rozmowe, czesc na pisanie i tak dalej. W przypadku serwera plików szacunkowa liczba równoczesnych polaczen jest jeszcze mniejsza, poniewaz jest malo prawdopodobne, by wszyscy usilowali równoczesnie zapisac plik.

Monitorowanie wydajnosci w systemach uniksowych

Swiat Microsoftu uzywa tylko jednego narzedzia — Monitora wydajnosci — które zajmuje sie prawie wszystkim, natomiast srodowiska uniksowe posiadaja wiele różnych narzedzi, lacznie z narzedziami dodatkowymi, które mozna znalezc np. w Internecie. Z tego powodu biezacy podpunkt omawia tylko podstawowe narzedzia, które powinny byc zawarte we wszystkich (albo prawie wszystkich) wersjach systemu Unix. Prosze pamietac, ze pomiedzy różnymi wersjami narzedzi moga istnieć różnice, wiec w razie otrzymania dziwnych wyników nalezy sprawdzic w dokumentacji przyczyny ich wystąpienia polecением `man` (`man` od *manual* jest poleceniem uniksowym, dostarczajacym pomocy na temat dowolnego wybranego polecenia lub funkcji. Na przyklad `man ifconfig` wyświetli opis polecenia `ifconfig`).

Ponizsza lista obejmuje wiekszosć popularnych polecen, które moga posluzyc do kontroli wydajnosci serwera uniksowego:

- ◆ `uptime` — polecenie `uptime` zwraca jeden wiersz tekstu, zawierajacy biezacy czas, informacje od jak dawna system jest uruchomiony oraz przecietne obciazenie systemu z ostatnich: jednej, pieciu i pietnastu minut. Srednie obciazenie oznacza srednia liczbe procesow, gotowych do uruchomienia podczas ostatniej minuty, pieciu minut i pietnastu minut.

- ♦ **w** — polecenie w wyświetla informacje o użytkownikach obecnie zalogowanych do serwera i ich procesach. W pierwszym wierszu wyświetla informacje z polecenia `uptime`, a następnie informacje o obciążeniu ze strony każdego zalogowanego użytkownika. Do informacji należa: nazwa logowania, nazwa terminala `tty`, czas jalowy, `JCPU`, `PCPU` i polecenie bieżącego procesu. `JCPU` oznacza czas procesora dla wszystkich procesów, które użytkownik wykonuje z danego połączenia, łącznie z procesami tła. `PCPU` oznacza czas procesora jedynie dla aktualnego procesu interaktywnego.
- ♦ **top** — przedstawia w czasie rzeczywistym obciążenie procesorów. Polecenie `top` wyświetla liste najbardziej obciążających procesor zadan w systemie i udostępnia tryb interaktywny, w którym można manipulować procesami. Zadania mogą być sortowane według wykorzystania procesora, wykorzystania pamięci lub czasu przebiegu. Do danych wyjściowych polecenia należa:
 - ♦ `uptime` — wyświetla te same dane, co polecenie `uptime`.
 - ♦ `processes` — całkowita liczba procesów uruchomionych podczas ostatniej aktualizacji oraz podział procesów na działające, `spione`, zatrzymane i „zombie”.
 - ♦ `CPU states` — pokazuje procentowe wykorzystanie procesorów w trybie użytkownika, trybie systemowym, przez zadania o obnizonym priorytecie i czas jalowy. Zadania o obnizonym priorytecie są również liczone w czasie użytkownika i systemu, więc suma przekracza 100%.
 - ♦ `mem` — statystyka wykorzystania pamięci, obejmująca pamięć całkowitą, pamięć wolną, dostępna przestrzeń w pliku wymiany (`swap`) i wykorzystana przestrzeń pliku wymiany.
- ♦ **ps** — polecenie `ps` dostarcza takich samych informacji, jak polecenie `top`, lecz dodatkowe parametry pozwalają za pomocą polecenia `ps` otrzymać więcej informacji o pojedynczym procesie.
- ♦ **pstree** — przedstawia widok uruchomionych procesów w strukturze drzewa, co pozwala na łatwe ustalenie relacji nadzędny-podzędny.
- ♦ **/proc** — podobnie jak wszystko inne w Uniksie, uruchomione procesy są traktowane jak pliki, tak że można nadać do nich uprawnienia (w przeciwnieństwie do traktowania procesów jako obiektów, jak to robią systemy Windows NT i 2000). `/proc` jest struktura pseudokatalogów, w której rezydują procesy. Mozemy tu znaleźć kilka plików, udostępniających szczegółowe informacje o uruchomionych procesach.
- ♦ **vmstat** — to polecenie udostępnia informacje o procesach, pamięci, stronicowaniu, blokowych procedurach we-wy oraz aktywności procesorów. Uruchomienie `vmstat` po raz pierwszy daje raport o średnich wartościach od ostatniego重启. Kolejne raporty dają informacje o okresie próbkowania o danej długości opóźnienia.
- ♦ **df** — polecenie `df` (*disk free*) wyświetla informacje o objętości dostępnej pamięci.
- ♦ **free** — to polecenie wyświetla całkowity rozmiar wolnej i zajętej pamięci fizycznej i pamięci wymiany w systemie, jak również pamięci wspólnej i buforów używanych przez jadro.

Dostępnych jest wiele różnych poleceń mogących pomóc w sprawdzaniu wydajności systemów uniwersalnych. Oprócz wymienionych powyżej dostępne są dosłownie setki innych.

Trzeba jednak pamiętać, by równowazyc zasoby systemowe i wyposażyc serwer w zasoby sprzętowe wystarczające do efektywnego odgrywania różnych wymaganych ról. Nie ma sensu wyposażanie w 1 GB RAM-u komputera, który będzie służył do edycji tekstu.

Omówiliśmy już narzędzia służące do monitorowania wydajności serwera, więc przyszła pora, by przyjrzeć się różnym narzędziom używanym do monitorowania sieci.

Narzędzia do monitorowania sieci

Jak Czytelnik zapewne się domysla, nawet najlepszy serwer na świecie na niewiele się przyda, jeśli sieć będzie na tyle przeciążona, iż nikt nie zdobędzie do niego dostępu — lub jeśli poziom błędów w sieci zmusi systemy do stałego ponawiania transmisji. W tym podrozdziale przyjrzymy się kilku narzędziom, które mogą posłużyć do kontroli wydajności sieci.

Najpierw wróćmy do początku rozdziału 23., w którym omówiliśmy łączenie ze zdalnym serwerem WWW „z dystansu”. Z podobnego dystansu powinniśmy monitorować sieć. Weźmy pod uwagę sieć, która składa się z pięciu podsieci, zawierających użytkowników podłączonych razem przez ruter do podsieci szkieletowej, w której znajdują się serwery. W tym przypadku zdolność przesyłania przez każdy serwer danych prosto do przełącznika gigabitowego Ethernetu z maksymalną predkością nie da zbyt dużego zysku, jeśli sieć po drugiej stronie przełącznika jest „zatkana”, lub jeśli jedna lub więcej sieci po drugiej stronie routera jest nasyconych.

Wiele z narzędzi służących do rozwiązywania problemów jest również przydatnych do zapewnienia odpowiedniej wydajności sieci. Zaczniemy od narzędzia najprostszego — polecenia ping. Jak na razie używaliśmy go, aby kontrolować dostępność połączenia i identyfikować źródło problemów w lokalnym stanie. Polecenie ping może posłużyć też do identyfikacji źródła problemów sieciowych.

Monitorowanie sieci za pomocą polecenia ping

Ping jest najbardziej popularnym narzędziem w środowisku TCP/IP, jednakże wielu użytkowników nie wykorzystuje w pełni jego możliwości. Oprócz prostego sprawdzania połączeń, ping może zostać użyty do ustalenia maksymalnego rozmiaru przesyłanego segmentu (MTSS — *Maksimum Transmit Segment Size*), który wskazuje, jak duży pakiet można przesłać dana trasa bez fragmentacji. Wykorzystamy to później, gdy będziemy mówić o rozmiarze okna TCP/IP. Ustalenie MTSS wymaga użycia opcji -l (długość) i -f (nie fragmentuj) oraz zwiększania rozmiarów pakietu, dopóki będzie mógł przejść przez połączenie bez fragmentacji.

Polecenie ping możemy też zastosować podczas tworzenia środowiska tras w sieci. Jest to szczególnie ważne, gdy planujemy wykorzystanie w sieci tras statycznych. Za pomocą opcji -j (luzne trasy źródłowe) i (lub) -k (scisłe trasy źródłowe) możemy pró-

bowac przeslac pakiety przez rózne zestawy ruterów i różnymi trasami, aby ustalic najlepszy zestaw ruterów dla transmisji miedzy dwoma punktami. Ping z tymi opcjami moze tez posluzyc do upewnienia sie, czy stosowany protokół wyboru tras generuje najlepsze mozliwe trasy; jesli nie, mozemy zmienic koszty skojarzone z przechodzeniem przez niektóre rutery, aby zapewnic wybór faktycznie najlepszej trasy.



Opcja **-j** (luzne trasy źródłowe) jest tez dostepna w narzedziu traceroute (tracert).

Opcja **-r** (rejestruj trase) moze posluzyc w poleceniu ping do szybkiego ustalenia trasy. Opcja ta jest również przydatna przy monitorowaniu i optymalizacji topologii tras.

Monitorowanie sieci za pomocą polecenia netstat

Jak wiemy, polecenie netstat moze byc uzyte do poznania polaczen sieciowych z innymi systemami i portów, na których nasz system nasluchuje (uslug). Dostepne sa dwie opcje, mogace zmienic sposób działania polecenia netstat. Opcja **-n** powstrzymuje netstat przed rozwiazywaniem adresów IP i numerów portów, wyswietlajac wyniki jedynie w postaci liczbowej. Skraca to czas wymagany na wyswietlenie informacji. Opcja **-n** bedzie przydatna, gdy ustawimy wartosc interwalu, aby odswiezac informacje okresowo, gdyz system nie bedzie musial rozwiazywac adresów IP na nazwy.

Opcja **-p** pozwala wyswietlic tylko jeden protokół: TCP, UDP lub IP. Pozwala ona tez przyspieszyc wyswietlanie i usunac z ekranu uboczne informacje. Na przyklad, w poniższym listingu zastosowane zostały opcje **-a**, **-n** i **-p**, aby szybko uzyskac okreslone informacje.

```
C:\>netstat -a -n -p TCP
```

Aktywne polaczenia

Protokół	Adres lokalny	Obcy adres	Stan
TCP	0.0.0.0:21	0.0.0.0:0	LISTENING
TCP	0.0.0.0:25	0.0.0.0:0	LISTENING
TCP	0.0.0.0:53	0.0.0.0:0	LISTENING
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:443	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1026	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1029	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1030	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1032	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1033	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1034	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3372	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING
TCP	192.168.1.2:20	192.168.1.1:4509	TIME_WAIT
TCP	192.168.1.2:20	192.168.1.1:4511	TIME_WAIT
TCP	192.168.1.2:21	192.168.1.1:4512	ESTABLISHED
TCP	192.168.1.2:139	0.0.0.0:0	LISTENING
TCP	192.168.1.2:1034	192.168.1.1:1433	ESTABLISHED

TCP	217.236.145.40:21	21.31.26.199:3161	CLOSE_WAIT
TCP	217.236.145.40:21	62.155.219.72:1612	CLOSE_WAIT
TCP	217.236.145.40:21	194.236.217.102:1414	CLOSE_WAIT
TCP	217.236.145.40:21	212.179.232.173:3984	CLOSE_WAIT
TCP	217.236.145.40:21	217.83.134.140:3704	CLOSE_WAIT
TCP	217.236.145.40:139	0.0.0.0:0	LISTENING
TCP	217.236.145.40:3389	24.112.92.45:3704	ESTABLISHED
TCP	217.236.145.42:21	24.70.96.200:61186	ESTABLISHED
TCP	217.236.145.42:80	206.98.210.9:42221	ESTABLISHED
TCP	217.236.145.42:80	206.98.210.9:42423	ESTABLISHED
TCP	217.236.145.42:80	216.154.50.72:1191	ESTABLISHED
TCP	217.236.145.42:80	216.154.50.72:1192	ESTABLISHED
TCP	217.236.145.42:80	24.42.182.247:3656	ESTABLISHED
TCP	217.236.145.44:80	24.42.182.247:3662	ESTABLISHED
TCP	217.236.145.44:80	24.69.255.202:43919	ESTABLISHED
TCP	217.236.145.44:80	24.69.255.203:29360	ESTABLISHED
TCP	217.236.145.44:80	24.69.255.203:29394	ESTABLISHED
TCP	217.236.145.44:80	24.69.255.204:20832	ESTABLISHED
TCP	217.236.145.44:80	24.69.255.204:20835	ESTABLISHED
TCP	217.236.145.44:80	24.69.255.205:7256	ESTABLISHED
TCP	217.236.145.44:80	24.71.223.140:43896	ESTABLISHED
TCP	217.236.145.44:80	205.200.178.107:3160	ESTABLISHED
TCP	217.236.145.44:80	205.200.178.107:3162	ESTABLISHED
TCP	217.236.145.44:80	205.200.178.107:3167	ESTABLISHED

W powyzszym listingu wynik jest wyswietlony w formacie numerycznym i koncentruje sie na protokole TCP. Wpisy z adresem obcym 0.0.0.0:0 oznaczaja porty nasluchujace — usluga otwarla port i nasluchuje klientow, ktore chca sie polaczyc. Dodajac na koniec polecenia liczbe (x), mozemy odswiezacz dane co x sekund. Odswiezanie informacji moze byc bardzo przydatne, jesli obserwujemy polaczenie ze zdalnego systemu, ktorego adres IP znamy. Informacje na koncu wierszy pozwalaja zorientowac sie w stanie polaczenia. Ponizsza lista opisuje stany raportowane przez polecenie netstat:

- ◆ *CLOSED* — sesja TCP zostala zamknieta.
- ◆ *FIN_WAIT_1* — polaczenie jest wlasnie zamykane.
- ◆ *SYN_RECEIVED* — odebrano zadanie sesji.
- ◆ *CLOSE_WAIT* — polaczenie jest wlasnie zamykane.
- ◆ *FIN_WAIT_2* — polaczenie jest wlasnie zamykane.
- ◆ *SYN_SEND* — trwa zadanie sesji.
- ◆ *ESTABLISHED* — pomiedzy systemami jest aktualnie nawiiazana sesja.
- ◆ *LISTEN* — usluga otwarla pasywnie port.
- ◆ *TIMED_WAIT* — sesja wlasnie czeka na dzialanie ze strony drugiego komputera.
- ◆ *LAST_ACK* — system wyslal ostatnie potwierdzenie.

Polecenie `netstat -r` pokaze tablice tras, natomiast aktualne statystyki Ethernetu mozna wyswietlic poleceniem `netstat -e`. Wynik bedzie wygladal podobnie, jak ponizszy listing:

```
C:\>netstat -e
Statystyki interfejsu

          Odebrano      Wyslano
Bajty        4081977305    3955519850
Pakiety unicast   18629760    21377538
Pakiety inne niz unicast   51740     11234
Odrzucone           0         0
Bledy             0         0
Nieznanne protokoly   s 2349364
```

C:\>

Statystyki Ethernetu zawieraja krótka klasyfikacje ruchu obecnego w sieci — jak jest intensywny i jaki jego odsetek jest nieprawidlowy (pozycje *Odrzucone* i *Bledy*). Aby uzyskac bardziej szczegółowe informacje o protokolach TCP/IP, mozemy uzyc opcji **-s** — uzyskamy statystyki poszczególnych protokolów, jak poniżej:

```
C:\>netstat -s
Statystyki IP

          Otrzymane pakiety      = 18647515
          Otrzymane bledy nagłówka = 0
          Otrzymane bledy adresu = 0
          Przekazane datagramy = 0
          Otrzymane nieznane protokoly = 0
          Otrzymane pakiety nastepnie odrzucone = 0
          Otrzymane pakiety nastepnie dostarczone = 18647847
          Zadania wyjściowe = 21404452
          Odrzucenia routingu = 0
          Odrzucone pakiety wyjściowe = 0
          Pakiet wyjściowy bez trasy = 0
          Wymagane ponowne asemblerwanie = 31
          Pomyslne ponowne asemblerwanie = 3
          Niepowodzenia ponownego asemblerowania = 23
          Datagramy pomyslnie pofragmentowane = 0
          Datagramy nie pofragmentowane = 0
          Utworzonye fragmenty = 0
```

Statystyki ICMP

	Odebrano	Wyslano
Komunikaty	4172	4066
Bledy	10	0
Miejsce docelowe nieosiagalne	2776	3228
Przekroczeno czas	488	20
Problemy z parametrami	0	0
Elementy wygaszajace zródła	90	0
Przeadresowania	15	0
Echa	742	76
Odpowiedzi echa	6	742
Daty powstania	0	0
Odpowiedzi dat powstania	0	0
Maski adresów	0	0
Odpowiedzi masek adresów	0	0

Statystyki TCP

Aktywne otwarcia	= 56720
Pasywne otwarcia	= 400635

Niepomyślne próby połaczenia	= 4056
Resetowane połaczenia	= 118408
Bieżące połaczenia	= 16
Otrzymywane segmenty	= 18218773
Wysłane segmenty	= 20675882
Retransmitowane segmenty	= 310091

Statystyki UDP

Otrzymywane datagramy	= 421972
Brak portów	= 48378
Błedy odbioru	= 0
Wysłane datagramy	= 418751

Powyzszy listing zawiera mnóstwo informacji, wiec przyjrzymy się kazdej sekcji z osobna.

Statystyki IP

Pierwsza sekcja sa statystyki IP. W przykładowym listingu statystyki te wyglądają całkiem niezłe. Pomimo pojawienia się kilku błędów, stosunek błędów do pakietów odebranych i dostarczonych jest tak niski (na poziomie ulamka procentu), iz moze byc zignorowany. Na przykład, całkowita liczba błędów wynosi 23 (wszystkie to błędy ponownego składania pakietów). Jesli porównamy te liczbę z 18,6 miliona pakietów odebranych bez błędów, zdamy sobie sprawę, iz odsetek błędów jest znikomy. W rzeczywistości wymienione błędy mozemy uznac za pakiety zniekształcone umyślnie lub nieumyślnie — próba włamania do serwera (komputer z przykładu jest serwerem WWW). Gdyby stosunek błędów do poprawnych danych byl znacznie wyższy, bylbы to powód do zaniepokojenia.

Statystyki ICMP

W sekcji statystyk ICMP widac, iz całkowita liczba wszelkich pakietów ICMP pozostała na dosyć niskim poziomie. Ogólnie mówiąc, te wartości wskazują, czy istnieje jakiś problem. W tym przykładzie liczba komunikatów ICMP jest dosyć niska w porównaniu z całkowita liczba pakietów wysłanych za pomocą IP.

Dla pełnego obrazu spójrzmy na poszczególne wartości. Pierwsza pozycja — *Błedy* — oznacza po prostu błędy. Sa to zwykle niepoprawne pakiety, w których informacje IP były w porządku, lecz informacje nagłówka ICMP były uszkodzone. Gdyby liczby w pozycji *Miejsce docelowe nieosiągalne* były wyższe, stanowiłyby to powód do niepokoju. Oznaczałoby to, iz serwer lub jeden z ruterów pomiędzy serwerem i hostem nie może znaleźć trasy do hosta docelowego.

Wiersz *Przekroczeno czas* podaje liczbę pakietów, które nie mogły zostać dostarczone z powodu przekroczenia limitu czasu w sieci. Jesli ta liczba jest wysoka w sieci wewnętrznej, oznacza to, ze prawdopodobnie jeden lub wiele segmentów jest przeciążonych pakietami, i ze routery w tym segmencie nie nadają z transmisją. Jesli połaczenie jest zewnętrzne, przeciążona jest sieć dostawcy usług lub Internet. Wiersz *Problemy z parametrami* wymienia liczbę pakietów, które zasadaly nie obsługiwanej funkcji, lub w których dane uległy uszkodzeniu.

Gdyby liczba komunikatów *Elementy wygaszajace zródła (Source Quench)* była znacząca, wskazywałoby to na problem z ruterem polozonym powyżej. Ruter wysyla pakiet tego typu, gdy nie jest w stanie obsłuzyć odbieranego ruchu i sygnalizuje, aby nadawca przestał wysyłać. *Przeadresowania ICMP* mówią nadawcy (w tym przypadku serwera), by użył innej bramy. Jeśli w tym wierszu zobaczymy dużą liczbę — ponad 50% — warto rozważyć zmianę bramy domysłnej dla danego serwera.

Echa i Odpowiedzi echa są lepiej znane jako *ping*, który te funkcje wykorzystuje. Jeśli zobaczymy bardzo dużą liczbę tych komunikatów, warto zarejestrować próbki ruchu sieciowego, ponieważ ping może służyć do najprostszego ataku Dos (*denial of service*). Komunikaty *Daty powstania (timestamp)* i *Maski adresów (address mask)* są zwykle spotykane w środowisku uniksowym; stanowią zadanie informacji (czas z serwera lub jego maska podsieci) — ponownie, jeśli liczba komunikatów jest znacząca, warto sprawdzić, kto wysyła komunikaty tego typu.

Statystyki TCP

W sekcji statystyk TCP wymienione są liczby otwartych aktywnych i pasywnych. *Otwarcie aktywne* oznacza połączenie wykonane przez serwer (w tym przykładzie do serwera SQL), natomiast *otwarcie pasywne* usługa otwarta (czekająca) na połączenia. Liczba *niepowodnych prób połączenia*, powodowanych przez błędy sieci lub zniekształcone pakiety, powinna być możliwie najniższa. *Resetowane połączenia* odnoszą się do liczby sesji, które serwer zamknął, zwykle z powodu upływu dopuszczalnego czasu połączenia. Wartość *Bieżące połączenia* jest przydatna, gdy wykonujemy testy obciążenia, aby sprawdzić, jak wielu użytkowników może połączyć się z systemem równoczesnie.

Liczba segmentów wysłanych i odebranych powinna być zbliżona do liczby wysłanych i odebranych pakietów IP. Wartości *Wyslane segmenty* i *Otrzymane segmenty* stanowią miary ruchu TCP, zorientowanego na sesje; jeśli więc komputer jest serwerem mediów, wysyłającymi dane grupowo w sposób ciągły, liczby te będą znacznie mniejsze od liczb IP. Ostatnia wartość, *Retransmitowane segmenty*, jest również ważnym wskaźnikiem ogólnego działania sieci. Jeśli liczba ta jest wysoka, duża część pakietów nie dociera do miejsca przeznaczenia. Oznacza to problemy z opóźnieniami lub błędami ruterów i wymaga bliższego zbadania. W przykładowym listingu wartość ta wynosi 310 091. Jeśli podzielimy ją przez liczbę segmentów wysłanych (20 675 882), stwierdzimy, że tylko 1,46% ruchu sieciowego trzeba wysyłać ponownie, co dla Internetu jest dobra wartość.

Statystyki UDP

Ostatnia sekcja przykładowego listingu są statystyki UDP. Wartości znajdują się tutaj zależnie od typu serwera. W naszym przykładzie komputer jest również serwerem DNS, wobec tego duże wartości nie stanowią problemu. Fakt, że około 10 procent pakietów (wartość *Brak portów podzielona przez Otrzymane datagramy*) idzie na nieokreślony port, może być problemem w sieci wewnętrznej. W Internecie oznacza to po prostu działalność hakerów usiłujących ustalić, które porty są otwarte w serwerze.

Jak widać, polecenie *netstat* pozwala szybko ustalić stan serwera. Oczywiście kontrola ta nie będzie wszechstronna, zwłaszcza jeśli serwer używa NetBIOS-u. Jak jed-

nakreś stwierdziliśmy w rozdziale 23., dostępna jest NetBIOS-owa wersja polecenia netstat o nazwie nbtstat.

Monitorowanie sesji NetBIOS za pomocą narzędzia nbtstat

nbtstat podaje nam informacje o sesjach NetBIOS w systemie, lacznie z informacjami o nazwach i rozwijaniu nazw.

Opcja -c, o której mówiliśmy w rozdziale 23., podaje nazwy z pamięci podręcznej nazw NetBIOS. Dodatkowo, kilka opcji pozwala zobaczyć nazwy zarejestrowane w naszym systemie lub innym systemie w sieci. Na przykład, opcja -n pozwala wyświetlić liste nazw zarejestrowanych w lokalnym komputerze. Wynik wygląda następująco:

```
C:\>nbtstat -n

HomeNet:
Node IpAddress: [192.168.0.1] Scope Id: []
NetBIOS Local Name Table

      Name          Type        Status
-----
HYDRA        <00>    UNIQUE    Registered
SCRIMGER     <00>    GROUP     Registered
HYDRA        <20>    UNIQUE    Registered
HYDRA        <03>    UNIQUE    Registered
SCRIMGER     <1E>    GROUP     Registered
INet~Services <1C>    GROUP     Registered
SCRIM        <03>    UNIQUE    Registered
IS~HYDRA.....<00>    UNIQUE    Registered
SCRIMGER     <1D>    UNIQUE    Registered
..__MSBROWSE__.<01>    GROUP     Registered
HYDRA        <01>    UNIQUE    Registered

Internet:
Node IpAddress: [48.53.66.7] Scope Id: []
NetBIOS Local Name Table

      Name          Type        Status
-----
HYDRA        <03>    UNIQUE    Registered
INet~Services <1C>    GROUP     Registered
SCRIM        <03>    UNIQUE    Registered
IS~HYDRA.....<00>    UNIQUE    Registered
HYDRA        <01>    UNIQUE    Registered
```

Polecenie nbtstat -n pomoże nam upewnić się, czy nazwa, z którą użytkownicy usiłują się połączyć, została zarejestrowana w sieci. Za pomocą opcji -a i -A możemy sprawdzić nazwy zarejestrowane w zdalnym komputerze. Opcja -a określa nazwę zdalnego komputera, zaś opcja -A pozwala podać adres IP zdalnego komputera. Wyniki są podobne do wyników dla opcji -n.

Opcja -r pozwala zobaczyć nazwy znalezione przez rozgłoszenia lub przez usługę WINS, lecz nie wyświetla nazw załadowanych wstępnie z pliku *lmhosts*.

Do monitorowania serwera opcja -s lub -S, wyświetlająca listę obecnie aktywnych sesji NetBIOS. Opcje te pozwalają monitorować przechodzenie sesji przez kolejne etapy.

Kolumna *Status* wskazuje aktualny stan polaczenia NetBIOS. W istocie, polaczenie moze znajdowac sie w jednym z wielu różnych stanów, wymienionych w ponizszej liscie:

- ◆ *Connected* — sesja NetBIOS zostala ustanowiona pomiedzy dwoma hostami.
- ◆ *Associated* — nasz system zazadal polaczenia i rozwiazal zdalna nazwe na adres IP.
Jest to otwarcie aktywne.
- ◆ *Listening* — usluga w naszym komputerze aktualnie nieuzywana (otwarcie pasywne).
- ◆ *Idle* — usluga, ktora otwarla port, zostala wstrzymana lub zawiesila sie.
Zadne czynnosci nie beda mozliwe az do wznowienia sesji.
- ◆ *Connecting* — na tym etapie nasz system usiluje utworzyc sesje NetBIOS.
System probuje wlasnie rozwiazac nazwe zdalnego hosta na adres IP.
- ◆ *Accepting* — usluga w naszym systemie zostala poproszona o otwarcie sesji i jest w trakcie procesu negocjacji sesji ze zdalnym hostem.
- ◆ *Reconnecting* — po odrzuceniu sesji (czesto z powodu przekroczenia limitu czasu) nasz system usiluje polaczyc sie ponownie.
- ◆ *Outbound* — wlasnie odbywa sie trojstronne potwierdzenie TCP, ktore ustanowi sesje w warstwie transportowej, sluzaca do nawiazania sesji NetBIOS.
- ◆ *Inbound* — jak *Outbound*, z tym ze polaczenie odbywa sie do uslugi w naszym systemie.
- ◆ *Disconnecting* — zdalny system zazadal przerwania sesji, wiec sesja zostaje wlasnie zamkniata.
- ◆ *Disconnected* — nasz system zada zakonczenia sesji.

Opcje `-s` i `-S` moga byc przydatne, jesli chcemy monitorowac aplikacje NetBIOS i obserwować kolejne stany sesji. Wciasz jednak patrzmy na siec z punktu widzenia serwera. Aby naprawde zrozumieć ruch sieciowy, z którym mamy do czynienia, trzeba zobaczyć same dane w sieci.

Przechwytywanie ruchu sieciowego za pomocą analizatorów pakietów

Podstawy dzialania analizatora pakietów (inaczej weszyciela pakietów) sa proste. Po-niewaz nosnik — Ethernet lub Token Ring — jest typu rozgloszeniowego, kazdy system w sieci odbiera wszystkie pakiety przesypane w sieci. W wiekszosci przypadków adres sprzetowy docelowego komputera nie jest dla nas interesujacy, wiec pakiet zostaje po cichu odrzucony. Zalozmy jednak, iz karta sieciowa nie jest wybredna i przyjmuje wszystkie odebrane pakiety i umiesci je w pliku.

Uzyskamy w ten sposob plik zawierajacy caly ruch krazacy w naszej sieci. Mozemy teraz przyjrzec sie pakietom i przeanalizowac informacje. W tym celu analizator pakietów musi przelaczyc karte sieciowa w tryb mieszany (*Promiscuous*), w którym caly ruch jest przyjmowany i przesypany w góre stosu. Dane te moga zostac nastepnie odczytane z pliku jako tekst lub zinterpretowane w narzedziu typu *Monitor sieci* (Network Moni-

tor). W następnych podpunktach zobaczymy dwa różne narzędzia: stosunkowo proste polecenie snoop systemu Solaris i bardziej złożony Monitor sieci Microsoft.

Program snoop

Snoop jest programem, który przelacza karte sieciową w tryb mieszany i przechwytuje wszystkie pakiety w sieci, w czasie rzeczywistym lub do pliku. W pierwszej kolejności musimy zdecydować, czy chcemy wyświetlać dane na monitorze w czasie rzeczywistym, czy chcemy przechwytywać pakiety do pliku. Realistycznie patrzyc, dane będą przewijane na ekranie zbyt szybko, aby były przydatne. Wobec tego musimy zapisać dane w pliku. Aby rozpoczęć proces przechwytywania, należy wydać polecenie:

```
#snoop -o nazwa_pliku
```

Wszystkie dane zostaną zapisane w postaci binarnej w pliku *nazwa_pliku*. W tym przypadku ruch będzie rejestrowany aż do zatrzymania programu, więc wygodnie będzie ustalić liczbę pakietów, jaka snoop ma zarejestrować (za pomocą opcji *-c*). Możemy też kontrolować szczegółowość informacji gromadzonych przez snoop, wybierając tryb *Operation*. Domyslnie snoop działa w trybie *Summary* (podsumowania), który dostarcza jedynie podstawowych informacji. Przykładowe wyjście w trybie *Summary* wygląda następująco:

```
86 0.01548 hydra -> TEST.FODDER.COM TELNET C port=5633
```

Możemy wybrać tryb *Verbose Summary* (szczegółowe podsumowanie) lub *Full Verbose* (szczegółowy). Aby uzyskać trybu *Verbose Summary*, należy dodać do polecenia opcję *-v*. Wynik będzie teraz wyglądać następująco:

```
86 0.01548 hydra -> TEST.FODDER.COM ETHER Type=0800 (IP), size = 58 bytes
86 0.01548 hydra -> TEST.FODDER.COM IP D=217.53.64.1 S=48.53.66.7
LEN=44, ID=5232
86 0.01548 hydra -> TEST.FODDER.COM TCP D=23 S=5633 Syn Seq=82349322
Len=0 Win=8760
86 0.01548 hydra -> TEST.FODDER.COM TELNET C port=5633
```

W tym przypadku widzimy, co zachodzi w każdej warstwie i możemy stwierdzić, iż wyjściowy pakiet pokazany powyżej jest pierwszym krokiem trójstronnego potwierdzania TCP. Na koniec, możemy (jesli chcemy lub musimy) uzyskać szczegółowe informacje o pakiecie przesyłanym przez sieć za pomocą opcji *-v*. Wynik będzie wyglądać następująco:

```
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 85 arrived at 11:35:27.37
ETHER: Packet size = 58 bytes
ETHER: Destination = 0:0:b5:0:17:e5, Sun
ETHER: Source      = 0:0:c:90:77:d8, Sun
ETHER: Ethertype = 0800 (IP)
ETHER:
IP:      ----- IP Header -----
IP:
IP:  Version = 4
IP:  Header length = 20 bytes
IP:  Type of service = 0x00
IP:          xxx. .... = 0 (precedence)
```

```
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... 0.. = normal reliability
IP:  Total length = 44 bytes
IP:  Identification = 6082
IP:  Flags = 0x4
IP:      .1.. .... = do not fragment
IP:      ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 255 seconds/hops
IP:  Protocol = 6 (TCP)
IP:  Header checksum = 6045
IP:  Source address = 48.53.66.7, hydra
IP:  Destination address = 217.53.64.1, test.fodder.com
IP:  No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 5633
TCP:  Destination port = 32 (TELNET)
TCP:  Sequence number = 82349322
TCP:  Acknowledgement number = 0
TCP:  Data offset = 24 bytes
TCP:  Flags = 0x02
TCP:      ..0. .... = No urgent pointer
TCP:      ...0 .... = No Acknowledgement
TCP:      .... 0... = No Push
TCP:      .... 0.. = No reset
TCP:      .... .1. = Syn
TCP:      .... ..0 = No Fin
TCP:  Window = 8760
TCP:  Checksum = 0x6de1
TCP:  Urgent pointer = 0
TCP:  Options: (4 bytes)
TCP:  - Maximum segment size = 1460 bytes
TCP:
TELNET:  ----- TELNET:  -----
TELNET:
TELNET:  ""
TELNET:
```

W rzeczywistych warunkach analiza zwykle zaczyna sie od trybu *Summary* lub *Verbose Summary*, aby dac pojecie, co dzieje sie w sieci, a nastepnie jest wykorzystywany tryb *Verbose*, który sluzy do analizy konkretnych problemów. Prosze pamietac, ze im wiecej danych rejestrujemy, tym bardziej obciążamy system dokonujacy rejestracji. Mozliwe jest przeciążenie systemu, zmuszajace do odrzucania pakietów.

Nastepne pytanie oczywiscie brzmi: jak czytac te informacje? Ponownie zastosujemy polecenie *snoop*, lecz tym razem z opcja *-i* oraz nazwa pliku, co pozwoli nam wyświetlic informacje. Mozemy nawet za pomoca opcji *-p* wskazac systemowi, które pakiety chcemy zobaczyć. To prowadzi do pytania: jesli rejestracja danych w trybie *Verbose* moze przeciązyc system, czy nie ma jakiegos sposobu na przechwycenie tylko potrzebnych nam danych?

Odpowiedz brzmi: jest. Mozemy wykorzystac mozliwosci filtrowania programu *snoop*, aby rejestrować tylko określone pakiety z sieci. Do mozliwych opcji nalezy filtrowanie wedlug adresu IP lub adresu MAC. Kierunek ruchu również moze byc filtrowany za pomoca polecen *to* (do) i *from* (od). Ponadto, mozemy nalozyc wiecej warunkow za pomoca operatorów *AND* i *OR* oraz filtrować ruch za pomoca operatorów *!* lub *NOT*. Do-

datkowo mozemy filtrowac pakiety wedlug protokolu transportowego, uzywajac parametrów `tcp`, `udp` lub `icmp` oraz filtrowac dane na podstawie portu.

Jak widac, snoop jest poteznym analizatorem pakietów, który pozwala przechwytywac i filtrowac ruch oraz na zyczenie analizowac pakiet po pakiecie. Nie jest on niesety dostepny dla platform Windows. W tym przypadku musimy skorzystac z Monitora sieci.

Monitor sieci

Zarowno snoop, jak i Monitor sieci sa analizatorami pakietów. Oba rejestruja i zapisuja ruch sieciowy i pozwalaja na późniejsza analize. Ponadto oba dysponuja trybem czasu rzeczywistego. Funkcja Monitora sieci jest taka sama, jak programu snoop, lecz forma jest odmienna; zas dla uzytkowników nie zaznajomionych z pakietami i ich strukturą Monitor sieci jest łatwiejszy w użyciu. Monitor sieci jest dostarczany z kilkoma analizatorami protokołów i wykonuje za uzytkownika spora czesc interpretacji pakietu, wyswietlajac zarejestrowane informacje w formacie łatwym do odczytania.

Prosze zdawac sobie sprawe, iz istnieja różne wersje Monitora sieci. Jedna jest dostarczana z Systems Management Server (SMS). Inna zawarta jest w Windows NT, lecz moze tylko przechwytywac dane do lub z systemu lokalnego i nie posiada mozliwosci odtwarzania lub edycji. Wersja dolaczana do SMS pozwala przelaczyc karte sieciowa w tryb mieszany i rejestrowac wszystkie dane w sieci.

Monitor sieci sklada sie z dwóch elementów. Pierwszym jest sterownik (w NT agent) monitora sieci (*Network Monitor Driver*), który faktycznie sluzy do rejestracji danych przeznaczonych do analizy. Drugim elementem jest sam Monitor sieci, który jest narzedziem pozwalajacym pracowac z zarejestrowanymi informacjami.

Korzystanie z Monitora sieci jest late. Otwórz Monitor sieci i — jesli o to zapyta — podaj, z której karty sieciowej chcesz rejestrowac pakiety. Monitor wyswietli wszystkie karty wedlug adresów MAC, wiec moze byc konieczne wydanie polecenia `ipconfig /all`, aby ustalic adres sprzetowy interesujacej nas karty. Po otwarciu narzedzia mozna za pomoca filtrowania zmniejszyc liczbe rejestrowanych pakietow lub po prostu wybrac z menu *Przechwytywanie/Rozpocznij*.

W trakcie przechwytywania wyswietlone sa cztery panele informacyjne, dostarczajace w czasie rzeczywistym informacji o rejestrowanych danych. Te panele to:

- ◆ *Wykres* — ten panel zawiera wykresy slupkowe, które dynamicznie wyswietlaja aktualna aktywnosc sieci. Obecne tu paski oznaczaja % wykorzystania sieci, ramki na sekunde, bajty na sekunde i transmisje grupowe na sekunde. Dla kazdego paska zaznaczony jest maksymalny poziom zarejestrowany dla danego wykresu.
- ◆ *Statystyka* — ten panel zawiera skumulowane statystyki sieci, podsumowujace ruch sieci w pieciu obszarach: statystyki sieci, statystyki rejestracji, statystyki na sekunde, statystyki karty sieciowej (MAC) oraz statystyki bledów karty sieciowej (MAC).
- ◆ *Statystyka sesji* — ten panel wyswietla statystyki sesji aktualnie dzialajacych w sieci.
- ◆ *Statystyka stacji* — ten panel pokazuje statystyki sesji, w których dany komputer uczestniczy.

W przeciwnieństwie do programu snoop, dane zapisywane sa do bufora w pamieci, co oznacza mniejsze prawdopodobienstwo utraty pakietów. Mozemy zmienic rozmiar bufora w *Przechwytywanie; Ustawienia bufora* pozwalaja kontrolowac, ile informacji chcemy zatrzymac. Gdy bufor zapelni sie, starsze dane zostaja odrzucone, by zrobic miejsce na nowe wpisy. Po zakonczeniu rejestracji mozemy albo zatrzymac przechwytywanie poleceniem *Przechwytywanie/Zatrzymaj*, albo wybrac *Przechwytywanie/Zatrzymaj i wyswietl*, aby przejsc bezposrednio do podgladu danych. Mozemy również zapisac dane do późniejszej analizy.

Na pierwszy rzut oka zarejestrowane dane wygladaja jak w trybie podsumowania programu snoop — lista wszystkich zarejestrowanych pakietów. Gdy wybierzemy dowolna ramke i klikniemy ja dwukrotnie, mozemy wejsc do ramki, aby zobaczyć wiecej szczegółów. Oryginalny panel podsumowania nadal bedzie wyswietlony na górze, jednakze pojawi sie dodatkowo okno szczegółów i okno z podgladem szesnastkowym. Okno szczegółów wyglada poczatkowo jak tryb Verbose Summary programu snoop, zas podglad szesnastkowy zawiera, jak nazwa wskazuje, surowe dane w postaci szesnastkowej. Gdy klikniemy wybrana czesc panelu szczegółów, w podgladzie szesnastkowym pojawia sie odpowiednie dane.

Sympatyczna funkcja jest mozliwosc dwukrotnego klikniecia informacji pokazanych w panelu szczegółów i przejścia do interesujacej nas czesci pakietu. Po otwarciu tych informacji mozemy klikac ciąg wpisów w panelu podsumowania i obserwować rozwijanie się ciągu zdarzeń.

Prezentacja zarejestrowanych danych jest podstawowa funkcja Monitora sieci. Dostepnych jest wiele sposobów filtrowania danych: przez edycje i ponowne wysłanie; przez ustalenie rozkładu protokołów w sieci z pomocą „ekspertów” i tak dalej. Mozemy nawet skonfigurować Monitor sieci do korzystania ze sterownika monitora w innym komputerze i zdalnej rejestracji danych.

Oczywiście istnieją niezliczone analizatory sieci — bardziej lub mniej rozbudowane. Musimy jednak znaleźć taki, który bedzie najlepiej sprawdzać się na naszej platformie. Informacje z analizatora pakietów mogą być z początku nieco przytłaczające, lecz są bardzo przydatne do dostrajania, rozwiązywania problemów i zabezpieczania sieci. Innym narzędziem, którego warto użyć do monitorowania sieci, jest protokół SNMP (*Simple Network Management Protocol*).

SNMP

SNMP (*Simple Network Management Protocol* — prosty protokół zarządzania siecią) pozwala zdalnie rozwiązywać problemy i monitorować koncentratory, routery i inne urządzenia. Za pomocą SNMP możemy gromadzić informacje o odległych urządzeniach bez konieczności obecności fizycznej przy urządzeniu i dla urządzeń bez interfejsu użytkownika. Wiele urządzeń sprzętowych, z którymi Czytelnik będzie mieć styczność, posiada pewna formę wbudowanego SNMP, tak że można nimi zarządzać zdalnie.

SNMP składa się z trzech odrebnego części: agenta, który jest częścią zarządzanego sprzętu lub oprogramowania; stacji zarządzającej, która pozwala monitorować sprzęt i oprogramowanie; oraz z bazy informacji zarządzania (MIB — *Management Information Base*).

tion Base), która udostępnia wspólny schemat nazewniczy pomiędzy agentami i stacjami zarządzającymi.

Za pomocą SNMP możemy centralizować monitorowanie sieci o praktycznie dowolnych rozmiarach. Protokół SNMP dla wszystkich swoich funkcji korzysta z UDP na portach 161 i 162, co oznacza, że ruch sieciowy jest utrzymywany na niskim poziomie, a dołączność pomiędzy hostami nie są wymagane sesje. Z drugiej strony, bezpieczeństwo to duży problem w SNMP. Pozwolenie każdemu na czytanie informacji SNMP, zamierzone lub niezamierzone, może być niebezpieczne, gdy SNMP może udostępnić mnóstwo informacji — zwłaszcza z systemów operacyjnych Microsoftu.

IETF usiłuje obecnie stworzyć bardziej bezpieczny protokół zarządzania, który powinien współpracować z równie dużą liczbą typów urządzeń jak SNMP. Póki co, bezpieczeństwo w SNMP jest zapewniane przez ustawienie *community name* — dosłownie nazwy zbiorczej.

Community name

Spolecznosc (*community*) w tym przypadku oznacza grupę zarządzającą zbiorem hostów, świadczących usługi SNMP. Spolecznosc składa się z przynajmniej jednej stacji zarządzającej i jednego lub wielu agentów. Spolecznosci otrzymują nazwy zbiorcze, które przypominają nazwy grup tworzonych na potrzeby zabezpieczeń. W większości implementacji domyślnej nazwa jest publiczna — należy ją zmienić, tak aby niepowolana osoba usiłująca dostarczyć się do informacji nie znalała *community name*.

Poza tymi nazwami w SNMP nie istnieją ustalone mechanizmy zabezpieczeń. Dane nie są szyfrowane. Jedna konfiguracja nie powstrzyma nikogo przed dostępem do sieci, odkryciem *community name* i adresów za pomocą weszyciela pakietów, a następnie wysyaniem do agentów sfalszowanych zadań odczytu danych. Dlatego też większość informacji dostępnych przez SNMP jest tylko do odczytu, co zapobiega nieautoryzowanym zmianom. Większość implementacji pozwala wyszczególnić, którym stacjom agent może odpowiadać oraz skonfigurować stacje, do której agent SNMP będzie wysyłać błędy uwierzytelnienia, zwane pulapkami (*trap*).

System zarządzania SNMP

System zarządzania jest głównym składnikiem SNMP. Ogólnie mówiąc, jest to program działający w systemie i pozwalający odpływać poszczególne agenty oraz odczytywać z nich wartości. Stacje zarządzające dodatkowo pozwalają ustawiać wartości obserwowane i automatyczne zapytania, umożliwiając obserwację sieci przez oprogramowanie i alarmując w razie problemów.

Przypominam, SNMP jest prostym protokołem i pozwala stacji zarządzającej wysyłać tylko kilka poleceń. Stacja wysyła zapytanie do agenta na port 161 UDP. Jeśli *community name* są takie same, a agent nie został skonfigurowany tak, aby nie odpowiadać stacji, zwraca informacje do stacji zarządzającej na ten sam port. Dla wszystkich urządzeń dostępnych są następujące polecenia:

- ◆ *get* — zadanie określonej wartości.

- ◆ *get-next* — zada wartosci nastepnego obiektu, co moze byc przydatne w przypadku kilku instancji tego samego obiektu (np. kilka adresow IP dla jednej karty interfejsu sieciowego).
- ◆ *set* — to polecenie ma za zadanie zmienic wartosc obiektu. Obiekty sa w wiekszosci tylko do odczytu z uwagi na brak zabezpieczen w SNMP.

W kazdym przypadku zostaje podany identyfikator obiektu (OID), aby wskazac agentowi, jaka wartosc chcemy zobaczymy lub ustawic. ID obiektu odnosi sie do MIB.

Agent SNMP

Agent SNMP jest zasadniczo odpowiedzialny za odpowiadanie na pytania stacji zarządzającej. Agent moze jednakze wyslac do stacji zarządzającej pulapke (*trap*), czyli wyjatek lub blad. Ta moze nastepnie podjac dzialania warunkowe, zwykle zanotowanie informacji do dziennika. W agencie SNMP mozna zazwyczaj skonfigurowac okreslone opcje:

- ◆ *Kontakt* — nazwisko osoby kontaktowej, która chcemy powiadomic o warunkach w stacji.
- ◆ *Lokalizacja* — pole opisowe dla komputera, pozwalajace znalezc system wysylajacy alert.

W konfiguracji agenta jest zwykle dodatkowe miejsce na ustawienie uslug, którymi agent zarządza. Lista typów obiektów, które agent moze monitorowac, wyglada następujaco:

- ◆ *Fizyczne* — agent zarządza urzadzeniami fizycznymi, na przyklad regeneratorami lub koncentratorami.
- ◆ *Aplikacje* — agent uzywa lub dostarcza aplikacji stosujacych TCP/IP.
- ◆ *Lacze danych/podsiec* — agent jest ruterem IP.
- ◆ *Dwupunktowe* — agent uzywa TCP/IP do komunikacji.

Jak widac, agent jest narzedziem podstawowym, co pozwala wbudowac go do oprogramowania sprzutowego w wielu urzadzeniach.

Baza informacji zarządzania

Baza informacji zarządzania (MIB — *Management Information Base*) jest po prostu zbiorem zmiennych opisujacych obiekty, którymi mozna zarządzac w agencie. Do odwołania do zmiennej uzywany jest identyfikator obiektu. Stacja zarządzająca moze odpytywac o wartosc zmiennej, a w niektórych przypadkach modyfikowac ja. Gdyby jednak kazdy producent po prostu wymyslal własne identyfikatory obiektów, przysporzyloby to klopotów — cos nazwane numerem 8 przez jednego producenta mogloby u innego producenta miec zupełnie inną nazwe.

Aby zapobiec takim typom niekonsekwencji, numeracja zmiennych MIB (ID obiektów) jest zarządzana przez International Standards Organization — ISO. Producent moze

zgłosic do ISO zapotrzebowanie na MIB i otrzymac punkt wyjściowy dla identyfikatorów swoich obiektów, co przypomina przydzielanie adresów IP. Następnie producent moze rozbudowywac przydzielony identyfikator obiektu, a nawet utworzyc hierarchie dla różnych linii produktów, a wewnatrz niej hierarchie produktów i tak dalej.

Kazdy obiekt posiada wiec całkowicie unikatowy identyfikator obiektu i nazwe. Na przykład, dla TCP/IP lub Internetu MIB II jest Iso.org.dod.internet.management.mibii, zas identyfikatorem obiektu 1.3.6.2.1. Oczywiście protokół SNMP uzywa do komunikacji numerów zamiast liczb, chociaż nazwy sa dla nas wyświetlane przez oprogramowanie zarządzające. W wielkości przypadków moześmy dodawać bazy MIB razem z obiekta przeznaczonymi do zarządzania — o ile dodajemy je zarówno do agenta, jak i stacji zarządzającej.

SNMP jest przydatnym narzędziem do zarządzania dużymi sieciami i jednym z niewielu, które działają na duże odległości. Jedynym problemem z SNMP jest bezpieczeństwo. Teraz, gdy poznaliśmy już narzędzia służące do monitorowania sieci, spójrzmy na rozmiar okna TCP. Ten parametr jest jedynym parametrem regulacji TCP/IP i wpływa na możliwe szybkości przesyłania danych.

Regulacja rozmiaru okna TCP/IP

TCP do transferu danych pomiędzy komputerami używa systemu okien przesuwanych. Kazdy komputer posiada okna nadawania i odbioru, których używa do buforowania danych i zwiększenia wydajności procesu komunikacji. Komunikacja jest bardziej wydajna, ponieważ okno moze „przesuwać się” nad danymi, co oznacza, iż danych nie trzeba dzielić na komunikaty. Zamiast tego dane w oknie zostają wysłane i potwierzone, a następnie okno przesuwa się dalej.

Okno odbioru pozwala komputerowi odbierać pakiety nie po kolejności i porządkować je podczas oczekiwania na kolejne. Taka reorganizacja moze być niebezpieczna, ponieważ TCP/IP nie gwarantuje kolejności dostaw pakietów. Zmiany warunków w sieci, zatory w routeraх i inne problemy mogą powodować późniejsze docieranie lub całkowita utratę pakietów. Ponieważ okno nadawania podczas nawijywania sesji otrzymuje rozmiar równy rozmiarowi okna odbioru, zmiana rozmiarów okna odbioru moze wpływać na sieciową wydajność systemu.

Proces przesuwania okna jest stosunkowo prosty. Podczas tworzenia sesji rozmiar okna nadawania zostaje ustalony na taki sam, jak rozmiar okna odbioru, zas obie stacje wymieniają i synchronizują numery sekwencji. Okno nadawania zostaje umieszczone na początku danych czekających w buforze na wysłanie. Dane z okna są kolejno pobierane i pakowane do segmentów TCP, w miarę wysyłania danych w każdym segmencie z odpowiednim numerem sekwencji.

System docelowy odbiera segmenty z warstwy IP i umieszcza je w oknie odbioru w kolejności numerów sekwencji zawartych w pakietach. Gdy okno odbioru zawiera uporządkowaną serię pakietów, zostaje wysłane potwierdzenie, wskazujące następny oczekiwany numer sekwencji. W trakcie wypełniania okno odbioru przesuwa się nad

odebranymi danymi; w miare odbierania potwierdzen okno nadawania jest przesuwane nad potwierdzonymi danymi i zostaja wyslane nastepne dane.

Zgodnie z ta logika, rozmiar okna ustala objetosc danych, jakie moga byc jednoczesnie obecne w sieci. Rozmiar okna powinien byc wielokrotnoscia maksymalnego rozmiaru przesyłanego segmentu (MTSS), który omówiliśmy przy okazji polecenia ping, aby zapewnić obecność w sieci tylko kompletnych segmentów. Jednakże zmiany rozmiaru okna moga prowadzić do poważnych problemów z lacznoscia.

Jesli ustawimy zbyt male okno, nie bedziemy nigdy mogli umiescic zbyt duzo danych w sieci. Byloby to w porzadku, gdyby dane byly rzeczy wiscie przesyłane z punktu A do B natychmiast. Jednakże skutek bedzie taki, iż wyslemy w siec niewielka porcje danych i bedziemy musieli czekac, az odbior zostanie potwierdzony, zanim nadamy kolejna porcje danych. Jesli okno bedzie zbyt duze, jego „przepchniecie” przez siec sprawi kłopoty, co skonczy sie na fragmentacji zwalniającej proces. Ryzykujemy tez ponowne wyslanie pakietu po uplywie czasu retransmisji tylko z tego powodu, iż odbiorca nie otrzymał jeszcze calego pakietu. Jak wszystko inne, powinnismy zmiany takie najpierw przewidzieć w laboratorium, zanim wprowadzimy je do srodowiska produkcyjnego.

W wiekszosci systemów operacyjnych rozmiar okna jest juz poprawnie dobrany dla sieci Ethernet. Zdarza się przypadki, w których bedziemy musieli ustalic najlepszy rozmiar okna. Jak powiedziano wczesniej, do tego celu sluzy polecenie ping z opcjami -f i -l. Opcja -f ustawia dla pakietu flagę zakazu fragmentacji, zas -l ustawia rozmiar pakietu.

Za pomocą polecenia ping z tymi parametrami mozemy zaczac pingowac zdalny serwer duzymi pakietami. Otrzymamy komunikat, iż pakiet musi zostac pofragmentowany, lecz zabrania tego opcja „nie fragmentuj”. W koncu otrzymamy odpowiedz ze zdalnego serwera. Teraz musimy próbować pomiedzy najmniejszą wartością, która nie zadziałała oraz największą, która zadziałała, aby zarezerwować dokładną wartość maksymalnego rozmiaru przesyłanego segmentu.

Poniższy listing przedstawia początek tego procesu. W koncu ustalony został maksymalny rozmiar segmentu wynoszący 1472 bajty.

```
C:\>ping 24.42.96.14 -n 1 -f -l 3000
Badanie 24.42.96.14 z użyciem 3000 bajtów danych:
Pakiet musi być podzielony na fragmenty, ale ustawiono opcję DF.

C:\>ping 24.42.96.14 -n 1 -f -l 2000
Badanie 24.42.96.14 z użyciem 2000 bajtów danych:
Pakiet musi być podzielony na fragmenty, ale ustawiono opcję DF.

C:\>ping 24.42.96.14 -n 1 -f -l 1000
Badanie 24.42.96.14 z użyciem 1000 bajtów danych:
Odpowiedź z 24.42.96.14: bajtów=1000 czas=201ms TTL=122

C:\>ping 24.42.96.14 -n 1 -f -l 1500
Badanie 24.42.96.14 z użyciem 1500 bajtów danych:
Pakiet musi być podzielony na fragmenty, ale ustawiono opcję DF.

C:\>ping 24.42.96.14 -n 1 -f -l 1250
Badanie 24.42.96.14 z użyciem 1250 bajtów danych:
Odpowiedź z 24.42.96.14: bajtów=1250 czas=221ms TTL=122
```

```
C:\>ping 24.42.96.14 -n 1 -f -l 1325
Badanie 24.42.96.14 z użyciem 1325 bajtów danych:
Odpowiedz z 24.42.96.14: bajtów=1325 czas=210ms TTL=122

C:\>ping 24.42.96.14 -n 1 -f -l 1400
Badanie 24.42.96.14 z użyciem 1400 bajtów danych:
Odpowiedz z 24.42.96.14: bajtów=1400 czas=241ms TTL=122

C:\>ping 24.42.96.14 -n 1 -f -l 1450
Badanie 24.42.96.14 z użyciem 1450 bajtów danych:
Odpowiedz z 24.42.96.14: bajtów=1450 czas=241ms TTL=122

C:\>ping 24.42.96.14 -n 1 -f -l 1475
Badanie 24.42.96.14 z użyciem 1475 bajtów danych:
Pakiet musi być podzielony na fragmenty, ale ustawiono opcje DF.
```

Następnymi czynnikami, które musimy wziąć pod uwagę przy ustawianiu rozmiaru okna TCP/IP, są niezawodność i szybkość sieci. Jeśli pozwolimy systemowi umieścić dużą liczbę pakietów w oknie nadawania, sieć musi być zdolna do niezawodnego przesłania ich do zdalnego komputera. Sieć musi również dążyć do przesłania wszystkich pakietów do zdalnego celu, zanim upłynie limit czasu dla retransmisji. Jeśli umieścimy dużą liczbę pakietów w oknie nadawania, równocześnie zostanie wysłana duża objętość danych. Jeżeli sieć nie poradzi sobie z nimi, skonczy się na retransmisjach, które najmniej nie zwiększą wydajności.

Liczba pakietów w oknie nadawania jest ustalana metodą prób i błędów. Dobra wartość wyjściowa jest osiem pakietów w oknie. Weźmy liczbę pakietów i pomóżmy przez maksymalny rozmiar przesyłanego segmentu. Uzyskany wynik jest rozmiarem okna odbioru, jaki należałoby zastosować. Rozmiar okna można zmienić w systemach uniwersalnych polecienniem `ifconfig`, zas w Windows przez modyfikacje Rejestru. Po ustawieniu wartości i ewentualnym restartie systemu, jeśli był wymagany, możemy spróbować przesłać dane pomiędzy dwoma hostami.

Teraz należy użyć polecenia `netstat`, aby określić liczbę zachodzących retransmisji. Jeśli nie było żadnych, możemy spróbować zwiększyć liczbę pakietów w oknie. Jeśli jest ich wiele, a transfer zachodzi powoli, należy zmniejszyć liczbę pakietów. Proces ten należy powtarzać, dopóki nie osiągniemy najbliższego możliwego transferu danych bez retransmisji. Nawiąsem mówiąc, do wszystkich testów powinniśmy użyć tego samego pliku, zas dla uzyskania dokładniejszych wyników należy użyć do testu dwóch systemów i skonfigurować w obu taki sam rozmiar okna.

Proszę jednak pamiętać, iż zoptymalizuje to jedynie ruch sieciowy pomiędzy dwoma hostami. Podczas komunikacji z innym hostem za pomocą systemu, w którym ustawiliśmy rozmiar okna, wyniki mogą być różne. Zawsze powinniśmy zanotować oryginalną wartość, by móc ją przywrócić.

Rozdzial 25.

Plany na przyszlosc

W tym rozdziale:

- ◆ IPv6
- ◆ Bezprzewodowy Internet
- ◆ Inteligentne urzadzenia domowe

Pora przyjrzec sie najnowszym trendom technologicznym i ich wpływowi na przyszłe środowisko pracy i domowe. Zaczniemy od opisu protokołu IPv6, który został po raz pierwszy zarekomendowany jako następca IPv4 w roku 1994. W ciągu najbliższych kilku lat IPv6 powinien zdobyć silna pozycje. Protokół ten nie zastąpi IPv4 za naszego życia, lecz stanie się dominującym protokołem dla określonych urządzeń, które pojawia się na rynku w najbliższej przyszłości. Następnie rzucimy okiem na świat bezprzewodowego Internetu. W miarę zwiększenia liczby użytkowników mobilnych i zmian w tradycyjnym środowisku pracy, będą opracowywane nowe technologie komunikacyjne, tak jak w latach 80. stworzono pager i telefon komórkowy. Wielu dzisiejszych producentów telefonów komórkowych i urządzeń PDA/PDM (*Personal Digital Assistant/Personal Digital Manager*) pracuje usilnie nad połączeniem obu technologii w jedno urządzenie. Powstały już takie urządzenia, jak telefon Neophone i Nokia Communicator. Dodatkowo inne urządzenia, jak np. Blackberry firmy Research in Motion, mogą pomóc w utrzymaniu połączenia 24 godziny na dobę, siedem dni w tygodniu. Urządzenia tego typu i inteligentne urządzenia domowe jako pierwsze zapewne wykorzystają przestrzeń nazw IPv6.

TCP/IP wkroczy również do gospodarstw domowych w obszarze urządzeń łączących się z Internetem. Zastosowania te daleko wykraczają poza urządzenia Web TV, co zaczyna być widoczne w statystykach odwiedzin stron WWW. NCR — producent urządzeń gospodarstwa domowego — opracował już Microwave Bank, który pozwala sprawdzić stan konta bankowego na ekranie dotykowym wbudowanym w drzwiczki mikrofalówki. Różni producenci są już w trakcie tworzenia lodówek, które pozwalają nawigować po Sieci z monitora wbudowanego w drzwi lodówki. Wygląda na to, że jest popyt na taki typ technologii, zwłaszcza że producenci są gotowi wydawać miliony dolarów na ich opracowanie. Urządzenia tego typu będą prawdopodobnie używać stosu IPv6.

Pod koniec rozdziału omówimy, jak uporać się z zachodzącymi zmianami technologicznymi. Zaczniemy teraz od omówienia IPv6, ponieważ większość nowych technologii będzie zależna od pojemności przestrzeni adresowej.

Wprowadzenie do IPv6

Jak juz wspomielismy, obecnym standardem IP jest IPv4. Prace nad IPv6 (inaczej IPng — *IP Next Generation*) zaczely sie w roku 1991, lecz protokol nie byl oficjalnie zalecany jako nastepca IPv4, az do spotkania IETF w Toronto w 1994 r. Jednym z glownych powodow opracowania IPv6 byl wzrost liczby hostow podlaczonych do Internetu. Wzrost ten zagrazal zajeciem caiej przestrzeni adresowej Internetu i zaczal utrudniac zarzadzanie architektura trasowania. Trzeba bylo wowczas zajac sie tymi powaznymi problemami.

Oczywiscie IPv6 nie byl jedynym zaproponowanym rozwiazaniem. Opracowano tez inne rozwiazania, to znaczy protokol tlumaczenia adresow NAT (*Network Address Translation*) i serwery proxy, ktore opieraja sie na NAT. Protokol NAT i serwery proxy pozwolily przedsiębiorstwom uzywac wewnetrznie prywatnych adresow sieci i nadal laczyc sie z Internetem. Zredukuwalo to znacznie liczbe potrzebnych adresow IP do kilku, potrzebnych dla uslug (np. serwerów proxy, poczty lub WWW), ktore sa niezbedne w przedsiębiorstwach dla kazdego podlaczonego komputera.

Dalsza redukcja uzycia adresow IP byla wciaz wymagana, poniewaz system klas adresow w IPv4 wymagal przydzielania firmom pelnych blokow adresow klasy C — przy najmniej 254 adresy. Na szczesie odkryto, iz podzial na podsieci moze odbywac sie zarówno na poziomie ISP, jak i na poziomie firmy, co pozwolilo ISP dzielic adresy klasy C tak, by kazda firma otrzymala przybliziona liczbe potrzebnych adresow.

Idea bezklasowego trasowania domen internetowych (CIDR — *Classless Internet Domain Routing*), czyli laczenie w nadsieci, mogla tez posluzyc do laczenia kilku malych adresow — na przyklad, grupy adresow klasy C — w cos, co przypominalo z wygladu i dzialania klase B lub jej czesc. Pozwolilo to firmom, ktore potrzebowaly wiecej niz 254 adresow, laczyc posiadane adresy w pojedynczy blok. W efekcie *cali* firma na potrzeby trasowania mogla byc widziana jak pojedynczy adres sieci.

Problem z trasowaniem w Internecie został w duzym stopniu rozwiazany za pomoca masek podsieci o zmiennej dlugosci (VLSM — *Variable Length Subnet Masking*), ktore pozwolily na opracowanie hierarchii w trasowaniu internetowym i osiągniecie wiekszej wydajnosci wyboru tras.

Po wprowadzeniu tych modernizacji do swiata IPv4, koniecznosc wprowadzenia IPv6 zdawala sie malec. Jak jednak wspomniano wczesniej, IPv4 stanie sie przypuszczalnie bardziej popularny — nie we „wlasciwym” Internecie, lecz w Internecie bezprzewodowym i inteligentnych urzadzeniach domowych. Doprowadzi to do powstania duzych, niemal niezaleznych sieci, ktore beda laczyc sie z wlasciwym Internetem za pomoca routerow z podwójnym stosem protokolow (czyli udostepniajacych zarówno funkcjonalosc IPv4, jak i IPv6).

Dzisiejsze rozmiary Internetu powoduja, iz szanse na zaistnienie IPv6 we wlasciwym Internecie sa bardzo male — zwlaszcza, ze wszystkie rutery na swiecie musialyby zostac równoczesnie wymienione. Moze wystapic dlugi okres przejsciowy, w którym zgodnosc bedzie osiagana za pomoca narzedzi typu rutery dwuprotokolowe. To zasadniczo rozpocznie powstawanie Internetu II. Gdy przedsiebiorstwa potrzebujace mozli-

wosci, jakie daje IPv6, zaczna ze soba współpracowac, wówczas beda tworzyc własna lacznosc. Skonczy sie przypuszczalnie na powstaniu dwóch równoległych Internetów: jednego znanego nam dzisiaj oraz drugiego, uzywajacego IPv6 na potrzeby sieci bezprzewodowych i inteligentnych urzadzen domowych.

Zmiany w porównaniu z IPv4

IPv6 wprowadza kilka zmian do protokolu IPv4. Dzieki nim IPv6 jest znacznie bardziej elastyczny i niezawodny oraz udostepnia praktycznie nieograniczoną przestrzeń adresowa. Oto lista najwa znieszych zmian:

- ◆ *Rozszerzone mozliwosci trasowania i adresowania* — IPv6 zwiększa długosc adresu z 32 bitów do 128 bitów. Pozwala to na obsługę większej liczby poziomów hierarchii adresowania i znacznie większej liczby adresowalnych wezłów.
- ◆ *Dodatkowe pole zakresu w adresie grupowym* — w ten sposób został utworzony nowy typ adresu o nazwie *anycast address* („adres uniwersalny”). Może on służyć do identyfikacji zbioru wezłów, gdzie do jednego z wezłów jest dostarczany pakiet wysłany na taki adres.
- ◆ *Uproszczenie formatu nagłówka* — część pól nagłówka IPv4 odrzucono lub zmieniono na opcjonalne. Zmniejsza to koszt obsługi pakietu i dodatkowe obciążenie sieci do minimum (mimo zwiększonej długosci adresów).
- ◆ *Ulepszona obsługa opcji* — nowy nagłówek IP koduje opcje w sposób pozwalający na wydajniejsze przekazywanie i na mniej ograniczeń długosci opcji, co daje elastyczne możliwości wprowadzania w przyszłość nowych opcji.
- ◆ *Wsparcie dla jakości usług (Quality of Service)* — dodano nowa możliwość nadawania etykiet pakietom, należącym do określonego przepływu danych, dla którego nadawca wymaga specjalnego traktowania, na przykład innej niż standardowej jakości usług lub usług w czasie rzeczywistym.
- ◆ *Wsparcie dla uwierzytelniania i prywatności* — IPv6 obejmuje definicje rozszerzeń, które umożliwiają obsługę uwierzytelniania oraz integralności i poufności danych.

W nagłówku IPv6, oprócz podstawowych składników, mogą pojawić się rozszerzenia nagłówka. Projektanci w ten sposób umożliwiły przyszłe rozszerzanie protokołu bez konieczności definiowania go w całości od nowa.

Nagłówki rozszerzające się pomiędzy faktycznym nagłówkiem a protokołem warstwy transportowej. Wobec tego wszelkie urządzenie nie „rozumiejące” nagłówków rozszerzających będą je ignorować. W ten sposób routery nie będą musiały analizować rozszerzeń, co zmniejszy dodatkowe obciążenie ruterów. Nagłówki rozszerzające nie są już ograniczone do 40 bajtów; fakt oddzielenia tych nagłówków od nagłówka IP oznacza, że w zasadzie mogą mieć dowolną długosć; ponadto możliwe jest stosowanie wielu nagłówków. Do już zdefiniowanych nagłówków rozszerzających należą:

- ◆ trasowanie,
- ◆ rozszerzone trasowanie (jak np. luźny wybór trasy),
- ◆ fragmentacja,

- ◆ fragmentacja i ponowne składanie,
- ◆ uwierzytelnianie,
- ◆ integralność i zabezpieczenie uwierzytelniania,
- ◆ opakowanie,
- ◆ poufnosc,
- ◆ opcje hopów,
- ◆ opcje miejsca przeznaczenia.

Oczywiście z biegiem czasu będą opracowywane dalsze nagłówki specjalne, aby umożliwić łączność w bezprzewodowym Internecie i intelligentnych urządzen domowych.

Adresowanie IPv6

W IPv6 adres uległ drastycznej zmianie. Gdy spojrzymy na implementacje IPv6 po raz pierwszy, zauważymy, że adres ma teraz 128 bitów zamiast 32. Pozwala to na niewątpliwie wielka liczbę adresów: dokładnie mówiąc, 340 282 366 920 938 463 463 374 607 431 768 211 456.

Nadal stosowane są adresy unicast (bezpośrednia transmisja do innego komputera) i rozgłoszeniowe (wysyłanie informacji do wielu komputerów dostrojonych do danej transmisji). Dodatkowo nowy typ adresu (*anycast*) pozwala kierować pakiet pod adres IPv6. Dowolna liczba hostów ze skonfigurowanym adresem anycast będzie w stanie odpowiedzieć — przypomina to nazwę grupy NetBIOS i może służyć do identyfikacji systemów, świadczących usługi.

Adres IPv6 możemy wyrazić w jednym z trzech różnych formatów:

- ◆ Format preferowany przedstawia adres w osmiu polach po 16 bitów. Każda szesnastobitowa grupa jest przedstawiana w postaci ciągu czterech liczb szesnastkowych, na przykład:
1079:0005:AB45:5F4C:0010:BA97:0043:34AB.
- ◆ Możemy ukryć początkowe zera w dowolnym polu. Każde z osmiu pól musi jednak zawierać przynajmniej jedna cyfra. Przykład: 1079:5:AB45:5F4C:10:BA97:43:34AB.
- ◆ Wiele przydzielonych adresów IPv6 zawiera długie ciągi zer. Dla takich adresów jest dostępna specjalna składnia, w której 0 jest zastępowane „::”, na przykład adres 1090:0:0:0:0:876:AABC:1234 możemy zapisać jako 1090::876:AABC:1234.

Bezprzewodowy Internet

Prawie nie można dzisiaj wyjść gdziekolwiek, by nie natknąć się na kogoś rozmawiającego przez telefon komórkowy. Telefony komórkowe stały się częścią życia codziennego do tego stopnia, iż niektóre znane nam osoby nie posiadają telefonu domowego; zamiast tego każdy członek rodziny posiada telefon komórkowy. Istnieje niewiarygodny nacisk

na coraz wieksza funkcjonalnosc telefonów komórkowych i uczynienie z nich lacznika z reszta swiata.

Wiekszosc sprzedawanych dzisiaj telefonów wyposazonych jest w przegladarke WAP — przegladarke Sieci wbudowana bezpośrednio w telefon. W rzeczywistosci cyfrowy telefon komórkowy jest wezlem sieci cyfrowej. Oznacza to, iz sam telefon przetwarza sygnaly analogowe (glos) na cyfrowe, a nastepnie przesyła dane w sposób cyfrowy za pomoca radiowego nadajnika-odbiornika. Sygnaly wchodzi nastepnie do sieci cyfrowej operatora, gdzie sa trasowane do osoby, z która rozmawiamy. W pewnym momencie informacje cyfrowe zostaja z powrotem przekształcone na analogowe i nasz glos moze zostac uslyszany.

Komputery sa urzadzeniami cyfrowymi (co znaczy, ze do składowania i przetwarzania informacji uzywaja zer i jedynek). Jesli wiec posiadamy siec cyfrowa, cyfrowy telefon i cyfrowy laptop, powinnismy byc w stanie komunikowac sie przez polaczenie telefonu cyfrowego. Jesli dysponujemy PDA (np. Blackberry) z wbudowanym modemem, telefon cyfrowy nie jest nawet potrzebny. Dane przesypane sa z wykorzystaniem komutacji obwodów z predkoscia 9600 b/s lub przez komunikaty wysylane jako SMS (*Short Message Service*).

Wprowadzany jest wlasnie najnowszy dodatek do komutacji obwodów i SMS — GPRS (*General Packet Radio Service*). GPRS obiecuje stworzenie trzeciej generacji systemów bezprzewodowych. System ten jest wprowadzany, aby wydajnie przesyłac dane z duza szybkoscia przez istniejace infrastruktury bezprzewodowych sieci komórkowych. Sygnalizacja i dane GPRS nie sa przesylane siecia komórkowa, która sluzy jedynie do znalezienia danych profilu uzytkownika GPRS. W zaleznosci od implementacji, GPRS moze wykorzystywac od 1 do 8 szczelin czasowych kanalów radiowych, które moga byc uzytkowane wspólnie.

W GPRS dane uzytkownika sa dzielone na pakiety i przesypane odrebnymi sieciami PLMN (Public Land Mobile Network — publiczna ladowa siec urzadzen mobilnych) z wykorzystaniem szkieletu IP. Dane po wejsciu do sieci szkieletowej IP (IPv4 lub IPv6) moga byc przesypane w dowolne miejsce. Dzieki temu GPRS udostepnia znacznie wieksze szybkosci niz istniejace technologie (od 14 400 b/s do 115 000 b/s, w porownaniu z 9600 b/s). Technologia ta czyni np. z przenosnego telefonu urzadzenie dajace dostep do Internetu. Z uwagi na zdolnosc do wykorzystywania różnych przepustowosci, GPRS pozwala zarówno na impulsowe przesyłanie danych (np. w poczcie elektronicznej i przeglądaniu WWW), jak i na przesyły duzych ilosci danych, dzięki czemu wykorzystanie laptopa z technologią komórkową staje się praktyczne. Ponadto GPRS obsługuje Quality of Service, wobec czego dostawcy usług mogą oferować użytkownikom usługi priorytetowe. W najbliższej przyszłości prawdopodobnie otrzymamy lepszą grafikę i bedziemy mogli zrobic wiecej z przeglądarkami mobilnymi, dostepnymi w telefonach komórkowych. Zapewne bedziemy jednak chcieli nabyc lepszy telefon (ponieważ przeglądanie stron WWW na wyświetlaczu 2x40 znaków jest lagodnie mówiąc irytujące). Oczywiście byloby doskonale, gdybysmy mogli laczyc sie z Internetem bezpośrednio z PDA. Wprawdzie mozna polaczyc sie z PDA z wbudowanym modemem, na przykład Palm VIIx lub RIM Blackberry, lecz dla innych urzadzen mo zna zastosowac dodatkowy modem, który pozwoli polaczyc sie z Internetem.

Obecnie wyzwanie stanowi integracja funkcji PDA i lacznosci telefonu komórkowego w pakiet nadajacy sie do uzycia. Kilku producentów osiągnęło to już w mniejszym lub większym stopniu. Podobnie jak w przypadku wszelkich urządzeń pierwszej generacji, możemy oczekiwac ulepszeń i szybkiego spadku cen.

WAP (*Wireless Application Protocol*) to ogólny termin, stosujący się do zestawu protokołów zdefiniowanych z inicjatywy firm Unwired Planet, Motorola, Nokia i Ericsson. W najbliższej przyszłości możemy oczekiwac dużego postępu w technologii WAP na potrzeby rosnącej armii pracowników mobilnych.

Przy obecnych ograniczeniach przepustowości i pasma nie damy rady po prostu wybrać witryny WWW i zaczac surfować. Informacje przesyłane do naszej komórki lub PDA zaczem radiowym muszą być nieco odmienne od zwykłej zawartości stron WWW. Wprawdzie w większości przypadków możemy odwiedzić standardową stronę WWW, lecz za pomocą bramy WAP. Brama taka działa jak serwer proxy, pobierając zazadaną stronę i przesyłając do użytkownika telefonu. Proszę pamiętać, iż w tym procesie usuwana jest grafika i inne funkcje strony WWW, których nasz telefon nie obsługuje.

Podobnie jak w przypadku TCP/IP lub dowolnej innej technologii sieciowej, protokół WAP składa się z kilku warstw:

- ♦ warstwa sieciowa — *Wireless Datagram Protocol* (WDP)
- ♦ warstwa zabezpieczeń — *Wireless Transport Layer Security* (WTLS)
- ♦ warstwa transakcji — *Wireless Transaction Protocol* (WTP)
- ♦ warstwa sesji — *Wireless Session Protocol* (WSP)
- ♦ warstwa aplikacji — *Wireless Application Environment* (WAE)

Wireless Datagram Protocol

Podobnie jak warstwa fizyczna w stosie TCP/IP, warstwa WDP funkcjonuje ponad usługami danych operatora — które, podobnie jak warstwa dostępu do sieci, zarządzają przesyłem zer i jedynek składających się na dane. Izoluje to protokoły wyższych warstw od szczegółów położonej poniżej sieci.

Wireless Transport Layer Security

Protokół WTLS, oparty na SSL lub *Transport Layer Security* (który jest nową definicją SSL), jest przeznaczony do użytku z protokołami transportowymi WAP i zoptymalizowany do wykorzystania w wąskopasmowych kanałach łączności. WTLS posiada następującą funkcjonalność:

- ♦ *Integralność i prywatność danych* — WTLS udostępnia szyfrowanie danych, więc dane przesyłane pomiędzy serwerem i urządzeniem są bezpieczne, a ponadto zapewnia, iż dane nie zostaną po drodze zmodyfikowane.
- ♦ *Uwierzytelnianie* — jak pokazaliśmy w rozdziale 22., za pomocą protokołu SSL można uwierzytelnić zarówno serwer, jak i klienta. Funkcje te są zawarte w WTLS.

- ◆ *Ochrona przed blokada uslug* — poniewaz uzywany jest klucz sesji, ataki przez przekazywanie moga zostac wykryte i odrzucone.

Ta warstwa stosu protokolów jest opcjonalna i podlega kontroli uzywanej aplikacji. Ponadto WTLS mozna uzywac pomiedzy dwoma urzadzeniami lub pomiedzy urzadzeniem i serwerem.

Wireless Transaction Protocol

Protokół WTP jest zasadniczo polaczeniem TCP i UDP; w istocie potrafi udostepnic podobny poziom uslug. Dane moga byc przesypane w jedna strone zarowno w postaci wiarygodnej, jak i nie gwarantowanej (co przypomina TCP i UDP), z potwierdzeniem kazdego pakietu. Ponadto mozliwa jest wiarygodna forma komunikacji, podobnie jak w protokole TCP. Dzieki polaczeniu funkcjonalnosci TCP i UDP zmniejsza sie potrzeba ladowania dodatkowego protokolu.

W WTP dane przeznaczone do wyslania moga byc skoncentrowane. Dzieki temu zamiast wysylania pieciu segmentow potrzebny bedzie jeden; stopien koncentracji zalezy od polozonej pod spodem sieci. Ponadto mozliwe sa transakcje asynchroniczne i opoznione potwierdzenia.

Wireless Session Protocol

Protokół WSP jest odpowiednikiem warstwy sesji w stosie TCP/IP. Udostepnia on warstwie aplikacji interfejs do protokolów z nizszych poziomów. WSP moze albo uzyc polaczeniowej warstwy transakcji, albo komunikowac sie bezposrednio z warstwa transportowa. W chwili obecnej WSP swiadczy glownie uslugi dostosowane do aplikacji przegladarek, jednakze to ograniczenie ulegnie zmianie w miare dojrzewania i ewoluwania protokolow.

Wireless Application Environment

W chwili pisania tej ksiazki warstwa aplikacji przede wszystkim tworzy szkielet dla przyszlego rozwoju. Twórcy maja nadzieję udostepnic pojedyncze, miedzyplatformowe srodowisko aplikacji bezprzewodowych (WAE — *Wireless Application Environment*), które bedzie bez problemów obsługiwane przez wszystkich producentów. Dzieki temu twórcy oprogramowania beda w stanie wydawac aplikacje dzialajace na wielu różnych platformach. Jak dotad, dostepna jest jedynie mikroprzegladarka zwierajaca nastepujace skladniki:

- ◆ *Wireless Makup Language (WML)* — uproszczony jezyk znakowania informacji, który jest w istocie zastosowaniem jezyka XML. WML został zoptymalizowany dla recznych urzadzen mobilnych.
- ◆ *WMLScript* — uproszczony jezyk skryptowy podobny do JavaScript.
- ◆ *Wireless Telephony Application (WTA, WTAI)* — uslugi telefonii i interfejsy programistyczne.

- ◆ *Content Formats (formaty zawartosci)* — zbiór dobrze zdefiniowanych formatów danych, obejmujących obrazy, rekordy książki telefonicznej i informacje kalendarzowe.

Oczywiście protokoly te stanowią obecne wyposażenie telefonów. Z upływem czasu PDA i telefon stają się jednym urządzeniem i dostępnych będzie wiele opcji, co doprowadzi do stosowania większych systemów operacyjnych. Bitwa systemów operacyjnych o dominację w świecie inteligentnych urządzeń domowych już się toczy pomiędzy różnymi producentami.

Inteligentne urządzenia domowe

Co to znaczy inteligentne urządzenie domowe? Szczególnie mówiąc, większość urządzeń sprzedanych w ciągu ostatnich kilku lat posiada „inteligencję” — to znaczy, wbudowane komputery i jakiś rodzaj środowiska operacyjnego. Jednakże ostatnio nasilają się tendencje do wbudowywania coraz bardziej złożonych środowisk operacyjnych w urządzenia wszelkich typów. Do tego dochodzi łączność — inaczej mówiąc, tworzone są urządzenia zdolne do komunikacji między sobą nawzajem i ze sprzedawcami oraz producentami.

Zamierzeniem twórców wydaje się być udostępnienie użytkownikowi łączności zewnątrz. Chciają oni, aby klient posiadał „informacje pod palcami” (filozofia Billiego Gatesa), aczkolwiek co niektórzy twierdzą, iż posuwa się to do skrajności. Na przykład, czy naprawdę potrzebna jest nam łódówka, która może zamawiać artykuły spożywcze, piekarznicza zaczajająca się na podstawie wiadomości o położeniu naszego samochodu (otrzymanej z systemu GPRS) lub sprzęt grający, który może łączyć się z Internetem? Może kiedyś nawet dojdziemy do szczoteczki do zębów, które będą powiadając naszego dentystę o ubytkach w zebach.

Ta wizja zaczyna mieć orwellowskie podteksty, lecz w rzeczywistości urządzenia takie już nadchodzą. Wielu użytkowników może już surfować w Internecie ze swojego telewizora lub słuchać muzyki przesyłanej przez Internet. Możemy już kupić łódówkę lub kuchenkę mikrofalową z przeglądarką wbudowaną w drzwiczki. Oczywiście rozwój takich urządzeń wywołuje szereg pytań dotyczących prywatności. Możemy niemal zagwarantować, iż wkrótce zobaczymy reklamy przesypane do naszego tostera. Musimy więc zaufać producentom, iż nie będą gromadzić osobistych informacji o nas.

Przejdzmy teraz do bardziej praktycznego zagadnienia. Z jakiego systemu operacyjnego będą korzystać te urządzenia? Jakie będą możliwości tego systemu? W walce o rynek urządzeń domowych uczestniczą trzech głównych rywali: Windows CE Microsoftu, JavaOS for Appliances firmy Sun oraz system operacyjny firmy Lucent o nazwie Inferno. Jak na razie żaden z nich nie może czuć się zwycięzcą. Rynek (pomyślmy tylko o liczbie urządzeń w naszym domu) może zostać podzielony pomiędzy systemy operacyjne. Przyjrzyjmy się Windows CE jako przykładowi takiego systemu operacyjnego, ponieważ zdobył chyba przewagę na starcie.

Microsoft musiał dopiero wejść na rynek systemów wbudowanych i uczynił to z nie jednym, a trzema wbudowanymi systemami operacyjnymi. We wszystkich przypadkach

były to dōsć pełne wersje systemu Windows. Nasuwa się oczywiste pytanie: po co wiejcej niż jeden wbudowany system operacyjny?

Windows CE jest odmiana Windows 98 w pełni dostosowana do rynku systemów wbudowanych, zawierająca podstawowe cechy projektowe wymagane do działania, lecz zapisane w pamięci ROM. Obejmuje to modułowość — inaczej mówiąc, system operacyjny musiał zostać podzielony na około 200 różnych modułów, które mogą być połączone w dokładnie takie środowisko, jakiego potrzebuje producent. Modułowość systemu operacyjnego pozwala producentowi usunąć niepotrzebne składniki, dzięki czemu rozmiary systemu mogą być zmniejszone nawet do 400 kB. Oznacza to, iż można w miarę potrzeb usunąć lub zaadaptować interfejs.

Dwa pozostałe systemy wybrane przez Microsoft jako wbudowane systemy operacyjne to Embedded NT i Windows 2000 z Server Appliance Kit, aczkolwiek oba systemy są przeznaczone raczej do małych przenośnych komputerów osobistych i dedykowanych serwerów, w których system operacyjny jest wbudowany bezpośrednio w sprzęt.

Ogólnie mówiąc, Embedded NT i Windows 2000 z Server Appliance Kit są wykonane solidnie — nie musimy restartować lodówka nazbyt często. Trzeba za to jednak zapłacić — każdy z tych systemów operacyjnych ma znacznie większe zapotrzebowanie na pamięć (*footprint*) niż Windows CE.

Niniejszy podrozdział koncentruje się na systemie Windows CE, ponieważ to on jest przeznaczony dla inteligentnych urządzeń domowych. W dużym skrócie Windows CE jest odmiana Windows zaprojektowana do pracy w komputerach wbudowanych. Do głównych właściwości tego systemu zaliczają się:

- ◆ *Podział na składniki* — system operacyjny składa się z nieco ponad 200 składników, co pozwala producentowi wybrać potrzebne elementy oraz umożliwi ograniczenie objętości pamięci nawet do 400 kB.
- ◆ *Niezależność od procesora* — Microsoft zapewnia obsługę szerokiego wachlarza sprzętu: 180 procesorów i setek BSP (*Board Service Package*), magistral, nosników pamięci i sterowników urządzeń. Ponadto producenci mogą tworzyć obsługę własnego sprzętu za pomocą narzędzia Platform Builder.
- ◆ *działanie w czasie rzeczywistym* — system operacyjny obsługuje pracę w czasie rzeczywistym z takimi możliwościami, jak ograniczone deterministycznie czasy reakcji, skrócone opóźnienia przerwan, 256 poziomów priorytetów, obsługa przerwan zagnieżdzonych i ochrona pamięci wirtualnej.

Podstawowa zaleta Windows CE jest jednak zaznajomienie twórców produktów z „pla-cem zabaw”, w którym będą pracować. Microsoft zainwestował wiele zasobów w umożliwienie projektantom szybkiego przejścia do środowiska Windows CE — ponieważ kluczem do zwycięstwa w bitwie systemów operacyjnych jest dostępność programistów i aplikacji, gdy producenci ich potrzebują. Windows CE daje programistom następujące dodatkowe możliwości:

- ◆ *Srodowisko programistyczne* — Windows CE posiada bogate środowisko programistyczne ze zintegrowanymi narzędziami do rozwoju i testowania. Narzędzia te udostępniają takie samo środowisko wizualne jak Visual Studio i wspólny model programistyczny Win32.

- ◆ *Rozszerzalnosc* — podzial systemu operacyjnego na składniki obejmuje mozliwosc tworzenia przez producentow i strony trzecie składników dolaczanych do systemu operacyjnego.
- ◆ *Wsparcie Microsoftu* — wsparcie ze strony Microsoftu obejmuje obsluge globalna i siec partnerow obejmujaca 200 integratorow systemu, ponad 60 niezaleznych twórcow aplikacji i 28 producentow ukladów scalonych.
- ◆ *Standardowe uslugi* — Windows CE za wiera wiele standardowych uslug, do których programisci sa przyzwyczajeni podczas programowania w systemie Windows, w tym DCOM, ADO i MSMQ (*Microsoft Message Queue Service*).
- ◆ *Uslugi internetowe* — Windows CE zawiera mozliwa do dostosowania wersje Internet Explorera i serwer WWW oraz obsluge jezyka XML.
- ◆ *ActiveX Data Objects (ADO)* — obiekty ADO daja programiscie mozliwosc laczenia sie z uslugami danych oraz mozliwosc lokalnej pracy z danymi.
- ◆ *Internet Connection Sharing (ICS)* — ICS pozwala na opracowywanie urzadzen opartych na Windows CE, które moga wspólnie korzystac z dostepu do Internetu.
- ◆ *Uslugi TCP/IP* — Windows CE zawiera wsparcie dla uslug DNS i WINS, Telephony API (TAPI) oraz protokolu SNMP.

Dzieki tym wszystkim narzedziom dostepnym dla programistow prawdopodobnie pojawi sie wiele zastosowan CE w istniejacych urzadzeniach podrecznych. W tej dziedzinie Microsoft ma zdecydowana przewage, poniewaz wielu uzytkownikow i programistow jest juz przy zwyczajonych do srodowiska Windows.

Planowanie na przyszlosc

Najlepszym sposobem przygotowania sie na przyszlosc jest zbudowanie solidnej sieci, pozwalajacej na rozwój, jakiego spodziewamy sie w przedsiebiorstwie w ciagu kilku najblizszych lat. W trakcie tworzenia sieci prosimy pamietac o tematach poruszonych w tej ksiazce. Pomoga one rozbudowywac siec i dodawac nowe protokoly i uslugi. Rownie wzorne jest monitorowanie serwerow sieciowych, aby upewnic sie czy caly czas funkcjonuja prawidlowo oraz w miare potrzeb modernizowac je lub zastepowac nowymi.

Radzimy tez byc na biezaco ze zmianami, jakie zachodza na rynku. Jesli bedziemy wiec o nadchdzacych zmianach i zapoznamy sie z nowymi technologiami, mozemy zaczac tworzyc strategie zaimplementowania ich w naszej sieci.

Warto stworzyc laboratorium, ktore posluzy nam i wspolpracownikom do testowania nowych uslug wprowadzanych w sieci. Laboratorium powinno nasladowac faktyczna siec tak dokladnie, jak to mozliwe.

Wprowadzanie zmian w sieci moze byc stresujace. Jednakze jest pewne, iz przemysl komputerowy nie zatrzyma sie nigdy w miejscu, wiec musimy byc na biezaco ze zmianami. Planujmy sieci tak, by mogly dostosowac sie do zmian, testujmy protokoly w rzeczywistych warunkach sieciowych i implementujmy nowe uslugi i protokoly w sposob miarowy i kontrolowany. To wszystko sprawi, iz nasza siec bedzie dzialac bez problemow.

Dodatki

Dodatek A

Domeny DNS najwyzszego poziomu

W tym dodatku:

- ◆ Ogólne domeny DNS najwyzszego poziomu
- ◆ Specjalne domeny DNS najwyzszego poziomu
- ◆ Narodowe domeny DNS najwyzszego poziomu

Domeny DNS najwyzszego poziomu (TLD — *Top Level Domain*) sa zgrupowane w trzy kategorie: ogólne, specjalne i kody krajów. TLD ogólne i specjalne sa uzywane przez przemysl i rząd USA, z wyjątkiem domeny *.com*, której uzywaja chyba wszyscy. TLD krajów uzywane sa przez przemysl i rządy pozostałych krajów. Niniejszy dodatek wymienia zawartosc tych kategorii wedlug stanu z czerwca 2001 r. Prosze zwrócić uwagę, iz wiele krajów zdecydowalo sie zaimplementowac ogólne i specjalne TLD jako poddomeny swojej domeny krajowej. Aktualna liste domen najwyzszego poziomu mozna znalezc pod adresem www.alldomains.com.

Ogólne domeny najwyzszego poziomu

Domeny te (gTLD — *generic TLD*) uzywane sa w USA i innych krajach dla przedsiębiorstw nastawionych na dochód, organizacji sieciowych i organizacji bezdochodowych:

- ◆ *com* — organizacje komercyjne
- ◆ *net* — organizacje sieciowe
- ◆ *org* — organizacje bezdochodowe

Specjalne domeny najwyzszego poziomu

Domeny te (sTLD — *special TLD*) uzywane sa w USA przez organizacje rządowe i wojskowe:

- ◆ *gov* — departamenty rządu USA
- ◆ *int* — organizacje międzynarodowe
- ◆ *mil* — wojsko USA

Narodowe domeny najwyższego poziomu z poddomenami

Inne kraje poza USA posiadają własne domeny najwyższego poziomu i poddomeny. Ponizej przedstawiona została obecna lista narodowych TLD z nazwami krajów i domenami:

- ◆ *ac* — Wyspa Wniebowstapienia
 - ◆ *com.ac*
 - ◆ *edu.ac*
 - ◆ *gov.ac*
 - ◆ *mil.ac*
 - ◆ *net.ac*
 - ◆ *org.ac*
- ◆ *ad* — Andora
- ◆ *ae* — Zjednoczone Emiraty Arabskie
 - ◆ *com.ae*
 - ◆ *net.ae*
 - ◆ *org.ae*
- ◆ *af* — Afganistan
- ◆ *ag* — Antigua i Barbuda
- ◆ *ai* — Anguilla
- ◆ *al* — Albania
- ◆ *am* — Armenia
- ◆ *an* — Antyle Holenderskie
- ◆ *ao* — Angola
- ◆ *aq* — Antarctica
- ◆ *ar* — Argentyna
 - ◆ *com.ar*
 - ◆ *net.ar*
 - ◆ *org.ar*

- ♦ *as* — Samoa Amerykańskie
- ♦ *at* — Austria
 - ♦ *ac.at*
 - ♦ *co.at*
- ♦ *au* — Australia
 - ♦ *asn.au*
 - ♦ *com.au*
 - ♦ *conf.au*
 - ♦ *csiro.au*
 - ♦ *gov.au*
 - ♦ *id.au*
 - ♦ *info.au*
 - ♦ *net.au*
 - ♦ *org.au*
 - ♦ *oz.au*
 - ♦ *telememo.au*
- ♦ *aw* — Aruba
- ♦ *az* — Azerbejdżan
 - ♦ *com.az*
 - ♦ *net.az*
 - ♦ *org.az*
- ♦ *ba* — Bośnia i Hercegowina
- ♦ *bb* — Barbados
 - ♦ *com.bb*
 - ♦ *net.bb*
 - ♦ *org.bb*
- ♦ *bd* — Bangladesz
- ♦ *be* — Belgia
- ♦ *bf* — Burkina Faso
- ♦ *bg* — Bułgaria
- ♦ *bh* — Bahrajn
- ♦ *bi* — Burundi
- ♦ *bj* — Benin

- ◆ *bm* — Bermudy
 - ◆ *com.bm*
 - ◆ *edu.bm*
 - ◆ *gov.bm*
 - ◆ *net.bm*
 - ◆ *org.bm*
- ◆ *bn* — Sultanat Brunei
- ◆ *bo* — Boliwia
- ◆ *br* — Brazylia
 - ◆ *art.br*
 - ◆ *com.br*
 - ◆ *esp.br*
 - ◆ *ect.br*
 - ◆ *g12.br*
 - ◆ *gov.br*
 - ◆ *ind.br*
 - ◆ *inf.br*
 - ◆ *mil.br*
 - ◆ *net.br*
 - ◆ *oeg.br*
 - ◆ *psi.br*
 - ◆ *rec.br*
 - ◆ *tmp.br*
- ◆ *bs* — Bahamy
 - ◆ *com.bs*
 - ◆ *net.bs*
 - ◆ *org.bs*
- ◆ *bt* — Bhutan
- ◆ *bv* — Bouvet Island
- ◆ *bw* — Botswana
- ◆ *by* — Bialorus
- ◆ *bz* — Belize

- ♦ *ca* — Kanada
 - ♦ *ab.ca*
 - ♦ *bc.ca*
 - ♦ *mb.ca*
 - ♦ *nb.ca*
 - ♦ *ns.ca*
 - ♦ *nt.ca*
 - ♦ *on.ca*
 - ♦ *pe.ca*
 - ♦ *qc.ca*
 - ♦ *sk.ca*
 - ♦ *yk.ca*
- ♦ *cc* — Wyspy Kokosowe (Keeling)
- ♦ *cf* — Republika Środkowej Afryki
- ♦ *cg* — Republika Kongo
- ♦ *ch* — Szwajcaria
- ♦ *ci* — Cote d’Ivoire (Wybrzeże Kości Słoniowej)
- ♦ *ck* — Wyspy Cooka
 - ♦ *co.ck*
- ♦ *cl* — Chile
- ♦ *cm* — Kamerun
- ♦ *cn* — Chiny
 - ♦ *ac.cn*
 - ♦ *ah.cn*
 - ♦ *bj.cn*
 - ♦ *com.cn*
 - ♦ *cq.cn*
 - ♦ *edu.cn*
 - ♦ *gd.cn*
 - ♦ *gov.cn*
 - ♦ *gs.cn*
 - ♦ *gx.cn*
 - ♦ *gz.cn*

- ◆ *hb.cn*
- ◆ *he.cn*
- ◆ *hi.cn*
- ◆ *hk.cn*
- ◆ *jl.cn*
- ◆ *js.cn*
- ◆ *ln.cn*
- ◆ *mo.cn*
- ◆ *net.cn*
- ◆ *nm.cn*
- ◆ *nx.cn*
- ◆ *org.cn*
- ◆ *qh.cn*
- ◆ *sc.cn*
- ◆ *sx.cn*
- ◆ *tj.cn*
- ◆ *tw.cn*
- ◆ *xj.cn*
- ◆ *xz.cn*
- ◆ *yn.cn*
- ◆ *zj.cn*
- ◆ *co* — Kolumbia
 - ◆ *arts.co*
 - ◆ *com.co*
 - ◆ *edu.co*
 - ◆ *firm.co*
 - ◆ *gov.co*
 - ◆ *info.co*
 - ◆ *mil.co*
 - ◆ *nom.co*
 - ◆ *org.co*
 - ◆ *rec.co*
 - ◆ *store.co*
 - ◆ *web.co*

- ♦ *cr* — Kostaryka
 - ♦ *ac.cr*
 - ♦ *co.cr*
 - ♦ *ed.cr*
 - ♦ *fi.cr*
 - ♦ *go.cr*
 - ♦ *or.cr*
 - ♦ *sa.cr*
- ♦ *cu* — Kuba
 - ♦ *com.cu*
 - ♦ *net.cu*
 - ♦ *org.cu*
- ♦ *cv* — Cape Verde
- ♦ — Wyspa Bożego Narodzenia
- ♦ *cy* — Cypr
 - ♦ *ac.cy*
 - ♦ *com.cy*
 - ♦ *gov.cy*
 - ♦ *net.cy*
 - ♦ *org.cy*
- ♦ *cz* — Czechy
- ♦ *de* — Niemcy
- ♦ *dj* — Dzibouti
- ♦ *dk* — Dania
- ♦ *dm* — Dominika
- ♦ *do* — Dominikana
 - ♦ *art.do*
 - ♦ *com.do*
 - ♦ *edu.do*
 - ♦ *gov.do*
 - ♦ *mil.do*
 - ♦ *net.do*
 - ♦ *org.do*
 - ♦ *web.do*

- ◆ *dz* — Algieria
- ◆ *ec* — Ekwador
 - ◆ *com.ec*
 - ◆ *k12.ec*
 - ◆ *edu.ec*
 - ◆ *fin.ec*
 - ◆ *med.ec*
 - ◆ *gov.ec*
 - ◆ *mil.ec*
 - ◆ *org.ec*
 - ◆ *net.ec*
- ◆ *ee* — Estonia
- ◆ *eg* — Egipt
 - ◆ *com.eg*
 - ◆ *edu.eg*
 - ◆ *eun.eg*
 - ◆ *gov.eg*
 - ◆ *net.eg*
 - ◆ *org.eg*
 - ◆ *sci.eg*
- ◆ *eh* — Zachodnia Sahara
- ◆ *er* — Erytrea
- ◆ *es* — Hiszpania
- ◆ *et* — Etiopia
- ◆ *fi* — Finlandia
- ◆ *fj* — Fidzi
 - ◆ *ac.fj*
 - ◆ *com.fj*
 - ◆ *gov.fj*
 - ◆ *id.fj*
 - ◆ *org.fj*
 - ◆ *school.fj*

- ♦ *fk* — Falklandy (Malwiny)
- ♦ *fm* — Mikronezja
- ♦ *fo* — Wyspy Faroe
- ♦ *fr* — Francja
- ♦ *fx* — Francja (stolica)
- ♦ *ga* — Gabon
- ♦ *gb* — Zjednoczone Królestwo (Wielka Brytania)
- ♦ *gd* — Grenada
- ♦ *ge* — Gruzja
 - ♦ *com.ge*
 - ♦ *edu.ge*
 - ♦ *gov.ge*
 - ♦ *mil.ge*
 - ♦ *net.ge*
 - ♦ *org.ge*
 - ♦ *pvt.ge*
- ♦ *gf* — Gujana Francuska
- ♦ *gg* — Wyspa Guernsey
 - ♦ *ac.gg*
 - ♦ *alderney.gg*
 - ♦ *co.gg*
 - ♦ *gov.gg*
 - ♦ *guernsey.gg*
 - ♦ *ind.gg*
 - ♦ *ltd.gg*
 - ♦ *net.gg*
 - ♦ *org.gg*
 - ♦ *sark.gg*
 - ♦ *sch.gg*
- ♦ *gh* — Ghana
- ♦ *gi* — Gibraltar
- ♦ *gl* — Grenlandia
- ♦ *gm* — Gambia

- ◆ *gn* — Gwinea
- ◆ *gp* — Gwadelupa
- ◆ *gq* — Gwinea Równikowa
- ◆ *gr* — Grecja
- ◆ *gs* — Wyspy South Georgia i South Sandwich
- ◆ *gt* — Gwatemala
- ◆ *gu* — Guam
 - ◆ *com.gu*
 - ◆ *edu.gu*
 - ◆ *gov.gu*
 - ◆ *mil.gu*
 - ◆ *net.gu*
 - ◆ *org.gu*
- ◆ *gw* — Gwinea-Bissau
- ◆ *gy* — Gujana
- ◆ *hk* — Hongkong
 - ◆ *com.hk*
 - ◆ *net.hk*
 - ◆ *org.hk*
- ◆ *hm* — Heard i Wyspy McDonalda
- ◆ *hn* — Honduras
- ◆ *hr* — Chorwacja
- ◆ *ht* — Haiti
- ◆ *hu* — Węgry
 - ◆ *co.hu*
 - ◆ *info.hu*
 - ◆ *nui.hu*
 - ◆ *org.hu*
 - ◆ *priv.hu*
 - ◆ *tm.hu*
- ◆ *id* — Indonezja
 - ◆ *ac.id*
 - ◆ *co.id*

- ♦ *go.id*
- ♦ *mil.id*
- ♦ *net.id*
- ♦ *or.id*
- ♦ *ie* — Irlandia
- ♦ *il* — Izrael
 - ♦ *ac.il*
 - ♦ *co.il*
 - ♦ *gov.il*
 - ♦ *k12.il*
 - ♦ *muni.il*
 - ♦ *net.il*
 - ♦ *org.il*
- ♦ *im* — Wyspa Man
 - ♦ *ac.im*
 - ♦ *co.im*
 - ♦ *gov.im*
 - ♦ *lkd.co.im*
 - ♦ *net.im*
 - ♦ *nic.im*
 - ♦ *org.im*
 - ♦ *plc.co.im*
- ♦ *in* — Indie
 - ♦ *ac.in*
 - ♦ *co.in*
 - ♦ *ernet.in*
 - ♦ *gov.in*
 - ♦ *net.in*
 - ♦ *nic.in*
 - ♦ *res.in*
- ♦ *io* — Brytyjskie Terytorium Oceanu Indyjskiego
- ♦ *iq* — Irak
- ♦ *ir* — Iran

- ◆ *is* — Islandia
- ◆ *it* — Włochy
- ◆ *je* — Jersey
 - ◆ *ac.je*
 - ◆ *co.je*
 - ◆ *gov.je*
 - ◆ *ind.je*
 - ◆ *jersey.je*
 - ◆ *ltd.je*
 - ◆ *net.je*
 - ◆ *org.je*
 - ◆ *sch.je*
- ◆ *jm* — Jamajka
- ◆ *jo* — Jordania
 - ◆ *com.jo*
 - ◆ *gov.jo*
 - ◆ *edu.jo*
 - ◆ *net.jo*
- ◆ *jp* — Japonia
 - ◆ *ac.jp*
 - ◆ *ad.jp*
 - ◆ *co.jp*
 - ◆ *gov.jp*
 - ◆ *net.jp*
 - ◆ *org.jp*
- ◆ *ke* — Kenia
- ◆ *kg* — Kirgizja
- ◆ *kh* — Kambodza
 - ◆ *com.kh*
 - ◆ *net.kh*
 - ◆ *org.kh*
- ◆ *ki* — Kiribati
- ◆ *km* — Komory

- ♦ *kn* — Federacja Saint Christopher i Nevis (Saint Kitts and Nevis)
- ♦ *kp* — Korea Północna
- ♦ *kr* — Korea Południowa
 - ♦ *ac.kr*
 - ♦ *co.kr*
 - ♦ *go.kr*
 - ♦ *nm.kr*
 - ♦ *or.kr*
 - ♦ *re.kr*
- ♦ *kw* — Kuwejt
- ♦ *ky* — Wyspy Kajmany
- ♦ *kz* — Kazachstan
- ♦ *la* — Laos
 - ♦ *com.la*
 - ♦ *net.la*
 - ♦ *org.la*
- ♦ *lb* — Liban
 - ♦ *com.lb*
 - ♦ *gov.lb*
 - ♦ *mil.lb*
 - ♦ *net.lb*
 - ♦ *org.lb*
- ♦ *lc* — Saint Lucia
 - ♦ *com.lc*
 - ♦ *edu.lc*
 - ♦ *gov.lc*
 - ♦ *net.lc*
 - ♦ *org.lc*
- ♦ *li* — Liechtenstein
- ♦ *lk* — Sri Lanka
- ♦ *lr* — Liberia
- ♦ *ls* — Lesotho

- ◆ *lt* — Litwa
- ◆ *lu* — Luksemburg
- ◆ *lv* — Lotwa
 - ◆ *asn.lv*
 - ◆ *com.lv*
 - ◆ *conf.lv*
 - ◆ *edu.lv*
- ◆ *ly* — Libia
 - ◆ *com.ly*
 - ◆ *net.ly*
 - ◆ *org.ly*
- ◆ *ma* — Maroko
- ◆ *mc* — Monako
- ◆ *md* — Moldawia
- ◆ *mg* — Madagaskar
- ◆ *mh* — Wyspy Marshalla
- ◆ *mk* — Macedonia
- ◆ *ml* — Mali
- ◆ *mm* — Myanmar (Zwiazek Myanmar dawna Birma)
 - ◆ *edu.mm*
 - ◆ *com.mm*
 - ◆ *gov.mm*
 - ◆ *net.mm*
 - ◆ *org.mm*
- ◆ *mn* — Mongolia
- ◆ *mo* — Macao
 - ◆ *com.mo*
 - ◆ *edu.mo*
 - ◆ *gov.mo*
 - ◆ *net.mo*
 - ◆ *org.mo*
- ◆ *mp* — Mariany Północne
- ◆ *mq* — Martynika

- ♦ *mr* — Mauretania
- ♦ *ms* — Wyspa Montserrat
- ♦ *mt* — Malta
 - ♦ *com.mt*
 - ♦ *net.mt*
 - ♦ *org.mt*
- ♦ *mu* — Mauritius
- ♦ *mv* — Malediwy
- ♦ *mw* — Malawi
- ♦ *mx* — Meksyk
 - ♦ *com.mx*
 - ♦ *net.mx*
 - ♦ *org.mx*
- ♦ *my* — Malezja
 - ♦ *com.my*
 - ♦ *edu.my*
 - ♦ *gov.my*
 - ♦ *net.my*
 - ♦ *org.my*
- ♦ *mz* — Mozambik
- ♦ *na* — Namibia
 - ♦ *com.na*
 - ♦ *net.na*
 - ♦ *org.na*
- ♦ *nc* — Nowa Kaledonia
 - ♦ *com.nc*
 - ♦ *net.nc*
 - ♦ *oeg.nc*
- ♦ *ne* — Niger
- ♦ *nf* — Wyspa Norfolk
- ♦ *ng* — Nigeria
- ♦ *ni* — Nikaragua
 - ♦ *com.ni*

- ◆ *nl* — Holandia
- ◆ *no* — Norwegia
- ◆ *np* — Nepal
 - ◆ *com.np*
 - ◆ *net.np*
 - ◆ *ort.np*
- ◆ *nr* — Nauru
- ◆ *nu* — Niue
- ◆ *nz* — Nowa Zelandia
 - ◆ *ac.nz*
 - ◆ *co.nz*
 - ◆ *gen.nz*
 - ◆ *govt.nz*
- ◆ *om* — Oman
- ◆ *pa* — Panama
 - ◆ *ac.pa*
 - ◆ *com.pa*
 - ◆ *edu.pa*
 - ◆ *gov.pa*
 - ◆ *net.pa*
 - ◆ *org.pa*
 - ◆ *sld.pa*
- ◆ *pe* — Peru
 - ◆ *com.pe*
 - ◆ *net.pe*
 - ◆ *org.pe*
- ◆ *pf* — Polinezja Francuska
- ◆ *pg* — Papua-Nowa Gwinea
- ◆ *ph* — Filipiny
 - ◆ *com.ph*
 - ◆ *mil.ph*
 - ◆ *net.ph*
 - ◆ *ngo.ph*
 - ◆ *org.ph*

- ♦ *pk* — Pakistan
- ♦ *pl* — Polska
 - ♦ *com.pl*
 - ♦ *net.pl*
 - ♦ *org.pl*
- ♦ *pm* — Terytorium Wysp Saint-Pierre i Miquelon
- ♦ *pn* — Wyspy Pitcairn
- ♦ *pr* — Puerto Rico
- ♦ *pt* — Portugalia
- ♦ *pw* — Palau
- ♦ *py* — Paragwaj
 - ♦ *com.py*
 - ♦ *edu.py*
 - ♦ *net.py*
 - ♦ *org.py*
- ♦ *qa* — Katar
- ♦ *re* — Reunion (Wyspa Francuska)
- ♦ *ro* — Rumunia
- ♦ *ru* — Federacja Rosyjska
 - ♦ *com.ru*
 - ♦ *net.ru*
 - ♦ *org.ru*
- ♦ *rw* — Rwanda
- ♦ *sa* — Arabia Saudyjska
- ♦ *sb* — Wyspy Salomona
- ♦ *sc* — Seszele
- ♦ *sd* — Sudan
- ♦ *se* — Szwecja
- ♦ *sg* — Singapur
 - ♦ *com.sg*
 - ♦ *edu.sg*
 - ♦ *gov.sg*
 - ♦ *net.sg*
 - ♦ *org.sg*

- ◆ *sh* — Wyspa Swietej Heleny
 - ◆ *com.sh*
 - ◆ *edu.sh*
 - ◆ *gov.sh*
 - ◆ *mil.sh*
 - ◆ *net.sh*
 - ◆ *org.sh*
- ◆ *si* — Slowenia
- ◆ *sj* — Wyspy Svalbard and Jan Mayen (Norwegia)
- ◆ *sk* — Slowacja
- ◆ *sl* — Sierra Leone
- ◆ *sm* — San Marino
- ◆ *sn* — Senegal
- ◆ *so* — Somalia
- ◆ *sr* — Surinam
- ◆ *st* — Wyspy Swietego Tomasza i Ksiazeca
- ◆ *sv* — Salwador
 - ◆ *co.sv*
- ◆ *sy* — Syria
 - ◆ *com.sy*
 - ◆ *net.sy*
 - ◆ *org.sy*
- ◆ *sz* — Swaziland
- ◆ *tc* — Turcja
- ◆ *td* — Czad
- ◆ *tf* — Francuskie Terytoria Poludniowe
- ◆ *tg* — Togo
- ◆ *th* — Tajlandia
 - ◆ *ac.th*
 - ◆ *co.th*
 - ◆ *go.th*
 - ◆ *net.th*
 - ◆ *or.th*

- ♦ *tj* — Tadzykistan
- ♦ *tk* — Tokelau
- ♦ *tm* — Turkmenia
- ♦ *tn* — Tunezja
 - ♦ *com.tn*
 - ♦ *edunet.tn*
 - ♦ *ens.tn*
 - ♦ *fin.tn*
 - ♦ *gov.tn*
 - ♦ *ind.tn*
 - ♦ *info.tn*
 - ♦ *intl.tn*
 - ♦ *nat.tn*
 - ♦ *net.tn*
 - ♦ *org.tn*
 - ♦ *rnr.tn*
 - ♦ *rns.tn*
 - ♦ *rnu.tn*
 - ♦ *tourism.tn*
- ♦ *to* — Tonga
- ♦ *tp* — Timor Wschodni
- ♦ *tr* — Turcja
 - ♦ *bbs.tr*
 - ♦ *com.tr*
 - ♦ *edu.tr*
 - ♦ *gov.tr*
 - ♦ *k12.tr*
 - ♦ *mil.tr*
 - ♦ *net.tr*
 - ♦ *org.tr*
- ♦ *tt* — Trynidad i Tobago
- ♦ *tv* — Terytorium Tuvalu

- ◆ *tw* — Tajwan
 - ◆ *com.tw*
 - ◆ *edu.tw*
 - ◆ *gove.tw*
 - ◆ *net.tw*
 - ◆ *org.tw*
- ◆ *tz* — Tanzania
- ◆ *ua* — Ukraina
 - ◆ *com.ua*
 - ◆ *gov.ua*
 - ◆ *net.ua*
- ◆ *ug* — Uganda
 - ◆ *ac.ug*
 - ◆ *co.ug*
 - ◆ *go.ug*
 - ◆ *or.ug*
- ◆ *uk* — Wielka Brytania
 - ◆ *ac.uk*
 - ◆ *co.uk*
 - ◆ *gov.uk*
 - ◆ *ltd.uk*
 - ◆ *mod.uk*
 - ◆ *net.uk*
 - ◆ *nhs.uk*
 - ◆ *org.uk*
 - ◆ *plc.uk*
 - ◆ *police.ik*
 - ◆ *sch.uk*
- ◆ *um* — US Minor Islands
- ◆ *us* — Stany Zjednoczone
- ◆ *uy* — Urugwaj
 - ◆ *com.uy*
 - ◆ *edu.uy*

- ♦ *net.uy*
- ♦ *org.uy*
- ♦ *uz* — Uzbekistan
- ♦ *va* — Watykan
- ♦ *vc* — Saint Vincent i Grenadyny
- ♦ *ve* — Wenezuela
 - ♦ *arts.ve*
 - ♦ *bib.ve*
 - ♦ *co.ve*
 - ♦ *com.ve*
 - ♦ *edu.ve*
 - ♦ *firm.ve*
 - ♦ *gov.ve*
 - ♦ *info.ve*
 - ♦ *mil.ve*
 - ♦ *net.ve*
 - ♦ *nom.ve*
 - ♦ *org.ve*
 - ♦ *rec.ve*
 - ♦ *stroe.ve*
 - ♦ *tec.ve*
 - ♦ *web.ve*
- ♦ *vg* — Wyspy Dziewicze (Brytyjskie)
- ♦ *vi* — Dziewicze Wyspy Stanów Zjednoczonych
 - ♦ *co.vi*
 - ♦ *net.vi*
 - ♦ *org.vi*
- ♦ *vn* — Wietnam
- ♦ *vu* — Vanuatu
- ♦ *wf* — Terytorium Zamorskie Wallis i Futuna
- ♦ *ws* — Samoa Zachodnie
- ♦ *ye* — Jemen

- ◆ *yt* — Mayotte
- ◆ *yu* — Jugosławia
 - ◆ *ac.yu*
 - ◆ *co.yu*
 - ◆ *edu.yu*
 - ◆ *org.yu*
- ◆ *za* — Republika Południowej Afryki
 - ◆ *ac.za*
 - ◆ *alt.za*
 - ◆ *co.za*
 - ◆ *edu.za*
 - ◆ *gov.za*
 - ◆ *mil.za*
 - ◆ *net.za*
 - ◆ *ngo.za*
 - ◆ *nom.za*
 - ◆ *org.za*
 - ◆ *school.za*
 - ◆ *tm.za*
 - ◆ *web.za*
- ◆ *zm* — Zambia
- ◆ *zr* — Zair
- ◆ *zw* — Zimbabwe

Skorowidz

3DES, 451, 455

A

accept, 150
Access Control List, *Patrz* ACL
ACK, 140
ACL, 214
Active Directory, 212, 360
Active Directory Services Interface, *Patrz* ADSI
Active Server Pages, *Patrz* ASP, *Patrz* ASP
ActiveX, 234
ActiveX Data Objects, *Patrz* ADO
AD, 347
adapter, 78
address mask, 492
address resolution), 94
Address Resoution Protocol, *Patrz* ARP
Administrator, 170
administrator dostepu, 91
ADO, 514
adres bramy, 111, 385
adres docelowy, 28, 87
adres hosta, 93, 371
adres internetowy, 93
adres IP, 28, 40, 44, 56, 93, 102, 163, 173, 177, 190, 371, 406, 412
 klasa, 56
adres nie trasowany, 227
adres portu docelowego, 62, 63
adres portu zródłowego, 62, 63
adres prywatny, 369
adres publiczny, 369
adres rozgłoszeniowy, 160
adres sieci, 93, 371
adres sieciowy, 111
adres sprzętowy docelowy, 96
adres sprzętowy nadawcy, 96
adres zródłowy, 88
adresowanie grupowe, 111, 403
adresy, 44
adresy zródłowy, 28
ADSI, 360
Advanced Research Projects Agency Network, *Patrz* ARPANET
AF_NET, 152
agent przekazujacy, 193
agent SNMP, 500
AH, 459
aktualizacje wyzwalane, 402
alfabet, 118
algorytm, 283
algorytm bezpiecznego mieszania, *Patrz* SHA-1
algorytm podpisu cyfrowego, *Patrz* DSA
algorytm powolnego startu, 139
alias, 195, 197, 198
AltaVista, 314
amplituda, 48, 66
Amplitude Shift Keying, *Patrz* ASK
ANCHOR, 309
AND, 108
Apache, 314
aparat wyszukiwania, 314
API, 148, 214
AppleTalk, 253
application, 336
Application Programming Interface, *Patrz* API
application proxy firewall, 238
application proxy software, 238
Application Service Provider, *Patrz* ASP
ARCnet, 65, 85, 91
ARP, 42, 85, 94, 100, 109, 113, 384, 430, 473
ARPANet, 431
ARPANET, 45, 305
ASBR, 408
ASCII, 257, 336
ASK, 66
ASP, 225, 313, 321
Asynchronous Transfer Mode, *Patrz* ATM
Asynchronous Transfer Mode Address Resolution Protocol, *Patrz* ATMARP
ATM, 82, 86, 92, 327, 433
ATMARP, 85, 94, 97
at-nbp, 146
ATRLS, 284
at-rtmp, 146
Attach, 331
audio, 66, 336
Authentication Header, *Patrz* AH

Autonomous System Border Route, *Patrz* ASBR
AXFR, 212

B

B2B, 324
B2C, 325
backbone, 49, 77
bajt, 93
bandwidth, 30
bazy danych SAM, 447
Berkeley, 149
Berkeley Internet Name Daemon, *Patrz* BIND
bezpieczeństwo, 359
bezpieczna powłoka ssh, 281
bezprzewodowy Internet, 508
bgp, 146
BGP, 29
bilet Kerberosa, 456
bilet przyznajacy bilety, 456
bind, 150
BIND, 161, 211, 213
binding, 147
bind-listen-connect- -accept, 148
bit, 47
Blackberry, 509
blok komunikatów serwera, *Patrz* SMB
blokada uslug, 447
BNC, 52, 71
BODY, 309
boot file, 211
boot information, 181
BOOTP, 157, 179, 414
bootpc, 145
bootps, 145
bootstrap, 179
BOOTstrap Protocol, *Patrz* BOOTP
Border Gateway Protocol, *Patrz* BGP
brama, 246
brama domyslna, 38, 107, 173
brama sieciowa, 38
bridge, 37, 368
Broadcast, 219
broadcast bus, 77
browser, 307
bruter, 37
BSD, 143, 149
buforowanie, 127
buforowanie drukowania, 293
bus topology, 32
business-to-business, *Patrz* B2B
business-to-consumer, *Patrz* B2C
bytecode, 319

C

C, 62
cache, 211
call message, 256
callback, 441
calkiem niezla prywatnosc, *Patrz* PGP
campus area network, 30, *Patrz* CAN
CAN, 30, 367, 429
carbon copy, 331
carriage return, 267
Carrier Sense Multiple Access/Collision Detection, *Patrz* CSMA/CD
Cascading Style Sheets, *Patrz* CSS
CD, 278
CDPD, 76
cell, 83, 93
cell switching, 69
Cellular Digital Packet Data, *Patrz* CDPD
Center for Nuclear Research, *Patrz* CERN
CERN, 153
certyfikat cyfrowy, 337
certyfikaty X.509, 453
CGI, 320
CHAddr, 184
challenge, 452
Challenge Handshake Authentication Protocol, *Patrz* CHAP
CHAP, 250
Character User Interface, *Patrz* CUI
chargen, 145
checkpoint, 27
CIAddr, 183
ciag wstepny, 87
CIDR, 109, 381, 506
Cisco, 110, 242
Citrix, 271, 289
Citrix MetaFrame, 285
class object, 348
classful routing, 109
Classless Inter-Domain Routing, *Patrz* CIDR
Classless Internet Domain Routing, *Patrz* CIDR, *Patrz* CIDR
client-side, 319
CLOSE_WAIT, 136
CLOSED, 135, 136
clustering, 425
cmip-agent, 146
cmip-man, 146
CNAME, 198
Collabra Discussions, 315
collision, 31
Common Gateway Interface, *Patrz* CGI
community name, 499
conference, 146

- connection establishment delay, 133
 connection establishment failure probability, 133
 connection release delay, 133
 connection release failure probability, 133
 connectionless, 126
 connection-oriented, 126
 contention, 89
 content-type, 336
 Control Channel, 266
 CPU, 483
 CRC, 29, 88, 89
 CSMA, 90
 CSMA/ CA, 90
 CSMA/CD, 31, 33, 79, 90, 367
 CSS, 312
 CUI, 272
 cyclical redundancy check, *Patrz* CRC
 czas dzierzawy, 188
 czas opóźnienia topologii, 404
 czas zwłoki, 408
 czas życia, 115
 częstotliwość, 48, 66
- D**
- daemon process, 162
 dane surowe, 29
 DARPA, 85, 305
 data encapsulation, 228
 Data Link, 87
 datagram, 64, 108, 109, 134, 150, 431
 daytime, 145
 DB-25, 52
 DDNS, 212, 416
 decentralizacja, 312
 default gateway, 38
 default route, 385
 defragmentacja, 61
 Demilitarized Zone, *Patrz* DMZ
 demodulacja, 74
 demon, 162, 257
 demultiplexowanie, 127
 denial of service, 447, 492
 DES, 323, 451, 453, 455
 DES CBC, 451
 DES Cipher Block Chaining, *Patrz* DES CBC
 DES XOR, *Patrz* DESX
 Designated Router, *Patrz* DR
 Destination Service Access Point, *Patrz* DSAP
 DESX, 451
 device file, 292
 df, 486
 DFS, 255
 DGRAM, 150
- DHCP, 44, 102, 157, 171, 179, 220, 412, 416, 464,
 472
 DHCPACK, 187, 188
 DHCPDISCOVER, 187
 DHCPNACK, 188
 DHCPOFFER, 187
 DHCPRELEASE, 188
 DHCPREQUEST, 187, 188
 DHTML, 312
 diagram, 115
 dial-on-demand, 395
 dial-up connection, 233
 DIB, 347
 digest, 449
 Digital Encryption Standard, *Patrz* DES
 Digital Signature Algorithm, *Patrz* DSA
 dioda laserowa, 73
 dioda świecąca, 73
 Directory Information Database, *Patrz* DIB
 Directory Information Shadowing Protocol, *Patrz*
 DISP
 Directory Information Tree, *Patrz* DIT
 Directory System Protocol, *Patrz* DSP
 disk free, 486
 DISP, 348
 Distance Vector Multicast Routing Protocol, *Patrz*
 DVMRP
 Distinguished Name, *Patrz* DN
 Distributed File System, *Patrz* DFS
 DIT, 348
 DLC, 88
 DMZ, 241
 DN, 348
 DNS, 26, 44, 131, 173, 197, 217, 412, 416, 465
 dobrze znane numery portów, 144
 docelowy adres IP, 64
 Document Object Model, *Patrz* DOM
 DOD/ARPA, 38
 DOM, 312
 domain, 145
 Domain Name System, *Patrz* DNS, *Patrz* DNS
 domena, 202
 domena komunikacyjna, 147
 domena NIS, 353
 domeny poziomu głównego, 203
 dostawca usług internetowych, 233
 dostęp zdalny, 271, 441
 dotted decimal notation, 93, 102
 DR, 409
 drukarka, 292, 294, 299
 drukarka lokalna, 291
 drukarka sieciowa, 214, 291
 drzewo informacji katalogowych, 348
 DSA, 347, 451
 DSAP, 88

- DSP, 347
DTP, 266
DUA, 347, 349
dumb terminal, 181
DVMRP, 121
Dynamic Host Configuration Protocol, *Patrz* DHCP, *Patrz* DHCP
dynamiczny DNS, 212
dynamiczny wybór tras, 396
dynamiczny wybór trasy, 57
dysk, 416
dysk twardy, 181
dzierzawa, 186
- E**
- EBCDIC, 257
Ecapsulating Security Payload, 460, *Patrz* ESP
echo, 145
ECMA, 319
ECMAScript, 319
e-commerce, 316, 324
EGP, 116, 397
ekran, 70
elastyczność, 359
element, 309
elementy wygaszające zródła, 492
EMF, 295
EMI, 70, 73
emoticon, 343
emulacja terminala, 442
end-to-end flow control, 59
Enhanced Metafile, *Patrz* EMF
Enterprise Resource Planning, *Patrz* ERP
ERP, 225
ESP, 459
ESTABLISHED, 135, 136
etc/printcap, 293
etc/exports, 261
etc/fstab, 163
etc/ftpusers, 162
etc/hosts, 161
etc/hosts.equiv, 162
etc/inetd.conf, 162
etc/networks, 162
etc/protocols, 162
etc/services, 162
etc/sysconfig/network, 163
eth0, 168
Ethernet, 31, 35, 85, 86, 89, 97, 160, 164, 286, 365, 372, 430, 433, 494
etiquette, 343
etykieta, 195, 197
EWAN, 276
ewolucja, 312
- Excite, 314
exec, 146
eXtensible HyperText Markup Language, *Patrz* XHTML
eXtensible Markup Language, *Patrz* XML
Exterior Gateway Protocol, *Patrz* EGP
eXternal Data Representation, *Patrz* XDR
- F**
- fala nosna, 66
FAQ, 343
FAT, 264
faza, 66
faza ustalenia adresu, 182
FCS, 87, 88, 89
FDDI, 65, 81, 392
FDM, 69
Fiber-Optic Ring, 34
File Transfer Protocol, *Patrz* FTP
filtr wejściowy, 298
FIN, 62, 141
FIN_WAIT_1, 135
FIN_WAIT_2, 136
finger, 145
fingerprint, 324
firewall, 248, 322
flooding, 406, 407
Folder list, 335
FONT, 310
footprint, 513
formaty zawartości, 511
FORWARD_IPV4, 163
fotodioda, 73
FQDN, 159, 197, 412, 474
fragmentacja, 61, 114
frame, 85
FRAME, 309
Frame Check Sequence, *Patrz* FCS
frame relay, 69, 431
free, 486
FreeBSD, 281
FreeSSH, 281
Frequency Division Multiplexing, *Patrz* FDM
Frequency Shift Keying, *Patrz* FSK
Frequently Asked Questions, *Patrz* FAQ
FrontPage, 314, 315
FSK, 66
ftp, 145
FTP, 26, 44, 60, 143, 162, 195, 266, 446, 452
FTP ASCII, 267
FTP format binarny, 267
FTP PASV, 266
FTP PORT, 266
ftp-data, 145

Fully Qualified Domain Name, *Patrz* FQDN
funkcja mieszania, 250

G

garbage collection, 402
gateway, 246
Gateway, 468
GATEWAY, 163
gateway mode, 161
GDI, 294
General Packet Radio Service, *Patrz* GPRS
Generic Routing Encapsulation, *Patrz* GRE
GET, 311
Get Msg, 335
GIAddr, 184
GIF, 310
GKS, 153
Global Catalog, 361
gluche terminale, 181
gluchy terminal, 285
gniazdo, 40, 147
gniazdo datagramów, 147
gniazdo potokowe, 147
goofy, 197
Google, 314
gopher, 145
Gopher, 306
gppitnp, 145
GPRS, 509
grafika rastrowa, 295
graphics, 145
Graphics Device Interface, *Patrz* GDI
GRE, 252
grupa dyskusyjna, 339
grupa moderowana, 340
grupa robocza, 23, 215
grupowanie, 425
grupy dyskusyjne, 315
grymas, 343
grzbiet, 66
GUI, 281
gwiazda, 80

H

haker, 234, 324, 447
hash functions, 250
Hash Message Authentication Code, 459, *Patrz*
 HMAC
hash value, 250
hasla nie szyfrowane, 452
haslo, 274, 278, 446
HEAD, 309, 311
header, 60
hierarchiczne drzewo katalogów, 359

hipertekst, 306
HLEN, 62, 96, 183
HMAC, 451, 459
hop, 383, 385
Hopy, 183
host bastionowy, 235
hostname, 145
HOSTNAME, 163
hosts, 475
HOSTS, 161, 196, 199, 217, 219
HTML, 307, 308, 317
http, 145
HTTP, 26, 44, 307, 310, 458
HTTPd, 314
HTTPS, 458
HTYPE, 183
Hyper Text Transfer Protocol, *Patrz* HTTP
HyperText Markup Language, *Patrz* HTML
HyperText Transfer Protocol, *Patrz* HTTP

I

IAB, 38, 230
IANA, 104, 144, 227, 369, 470
IBM, 33
ICA, 289
ICANN, 202
ICMP, 42, 100, 116, 386
ICS, 247, 514
ICV, 459
ID hosta, 104, 380
ID lokalizacji, 375
ID podsieci, 379
ID sieci, 104
IDE, 482
identyfikator, 106
identyfikator cyfrowy, 338
identyfikator hosta, 104
identyfikator sieci, 104
IEEE, 31
IESG, 38
IETF, 38, 185
ifconfig, 165, 166, 467, 503
IGMP, 42, 100, 116, 119
igmpv3lite, 146
IGP, 397
IGRP, 397, 403, 405
IHL, 115
IIS, 313
image, 336
imap, 146
IMAP, 333
IMAP4, 333
inbound mapping, 243
Inbox, 330, 334, 335

- Independent Computing Architecture, *Patrz* ICA
Index Server, 313
informacje inicjujące, 181
information superhighway, 305
infostrada, 305
infrastruktura klucza publicznego, 449
Input Filter, 298
Institute of Electrical and Electronic Engineers, *Patrz* IEEE
integralność, 250
Integrated Services Digital Network, *Patrz* ISDN
Integrity Check Value, 459
interfejs, 78, 111
interfejs hosta, 78, 81
interfejs urządzenia graficznego, *Patrz* GDI
interfejs usług Active Directory, *Patrz* ADSI
International Organization for Standardization., *Patrz* ISO
International Telecommunication Union, *Patrz* ITU
Internet, 143, 305
Internet Architecture Board, *Patrz* IAB, *Patrz* IAB
Internet Connection Sharing, *Patrz* ICS
Internet Control Message Protocol, *Patrz* ICMP
Internet Control Messaging Protocol, *Patrz* ICMP
Internet Engineering Task Force, *Patrz* IETF
Internet Explorer, 304, 307
Internet Group Management Protocol, *Patrz* IGMP, *Patrz* IGMP
Internet Information Server, *Patrz* IIS
Internet Mail Access Protocol, *Patrz* IMAP
Internet News, 315
Internet Packet Exchange, *Patrz* IPX
Internet Printing Protocol, *Patrz* IPP
Internet Protocol, 99, *Patrz* TCP/IP
Internet Research Task Force, *Patrz* IRTF
Internet Service Provider, *Patrz* ISP
Internet Services Provider, *Patrz* ISP
Internet Wave, 327
internetowy protokół drukowania, *Patrz* IPP
interwał przywitania, 408
intranet, 153
IP, 99, 101, 108, 112
IP docelowy, 96
IP Header Length, *Patrz* IHL
IP nadawcy, 96
IP Next Generation, 230
ipconfig, 177, 465
 all, 466, 475
 release, 467
 renew, 467
iPlanet Web Server, 314
IPng, 230
IPP, 304
IPSec, 17, 253, 458
IPSec policy, 462
IPv4, 105, 229
IPv6, 230, 505, 506
ipx, 146
IPX, 38, 253, 477
irc, 146
IRD, 396
IRSG, 38
IRTF, 38
ISAKMP, 462
ISDN, 70, 251, 327, 437
ISO, 24
ISP, 48, 104, 225, 369, 370, 382, 434
ITU, 347
ITU-T, 454
izolująca sieć lokalna, 241
- J**
- Java, 234, 313, 319, 443
Java Server Pages, *Patrz* JSP
Java Virtual Machine, 290, 319
Java Web Server, 314
JavaScript, 319
jednolity lokalizator zasobów, *Patrz* URL
język hipertekstowego znakowania informacji, *Patrz* HTML
język programowania, 319
język znakowania, 308
JPEG, 310
JScript, 319, 321
JSP, 320, 321
- K**
- kabel koncentryczny, 70
kabel koncentryczny cienki, 70
kabel koncentryczny gruby, 71
kanal, 89
kanaly wirtualne, 93
karta inteligentna, 287
karta interfejsu sieciowego, 36
karta sieciowa, 416, 421
kartasieciowa, 22
karuzela, 426
kaskadowe arkusze stylów, *Patrz* CSS
katalog główny, 263
kerberos, 455
Kerberosa, 324
klient, 21
klasa A, 106
klasa B, 107
klasa C, 106
klasa IP, 105
klastrowanie, 425
klastry, 419
klawiatura, 273

- klient, 22, 134
 klucz deszyfrujacy, 283
 klucz LAN Manager, 453
 klucz NT, 453
 klucz prywatny, 249, 323, 337
 klucz publiczny, 249, 323, 337, 450, 454
 klucz sesji, 456
 klucz szyfrujacy, 283
 kluczowanie amplitudy, 66
 kluczowanie czeostoliwosci, 66
 kluczowanie fazy, 67
 kluczy sesji, 459
 kod Manchester, 87
 kod posredni, 319
 kod zrodowy, 314
 kolejka do drukowania, 293
 kolizja, 31, 89
 komorka, 83
 kompresja, 26
 komputer wieloadresowy, 111
 komunikacja logiczna, 125
 komunikat, 61
 komunikat odkrycia, 187
 komunikat odpowiedzi, 256
 komunikat oferty, 187
 komunikat wywolania, 256
 komutacja, 56, 368
 komutacja komorek, 69
 komutacja komunikatów, 56, 68
 komutacja obwodów, 56, 68
 komutacja pakietów, 56, 69, 306, 431
 komutacja ramek, 431
 koncentrator, 33, 36, 52
 koncentrator aktywny, 52
 koncentrator pasywny, 52
 kontrola bledów, 59
 kontrola dostepu, 349
 Kontrola dostepu, 250
 kontrola dostepu do nosnika, 29
 kontrola lacza logicznego, 29
 kontrolowanie przeciazien, 127
 konwersja, 103
 korekcja bledów, 75
 korzen, 263, 358
 kroczenie po drzewie, 210
 ksiazka teleadresowa, 347
- L**
- l0phcrack, 447
 L2F, 252
 L2TP, 251, 252, 395, 439
 LAN, 30, 48, 53, 75, 93, 112, 429
 laptop, 441, 446
 laser, 75
- LAST_ACK, 136
 Layer 2 tunneling protocol, *Patrz* L2TP
 Layer-2 Tunneling Protocol, *Patrz* L2TP
 LC3, 447
 LDAP, 324, 347, 350, 360
 lease, 186
 LED, 73, 468
 lekki klient, 284
 Light Emitting Diode, *Patrz* LED
 Lightweight Directory Access Proto, *Patrz* LDAP
 limit okna, 139
 line feed, 267
 line printer daemon, *Patrz* lpd
 linia dzierzawiona, 48, 437
 Linux, 17, 149, 281, 392
 lista dyskusyjna, 315, 339
 listen, 150
 LISTEN, 136
 LLC, 87
 lmhosts, 478, 479
 LMHOSTS, 217
 local area network, *Patrz* LAN
 localhost, 167
 lockd, 258
 Logical Link Control, 29, *Patrz* LLC
 login, 146
 lokalny adresu zwrotny, 105
 loopback, 105, 111, 166
 loopback mode, 161
 Lotus Notes, 314
 lpc, 292, 302
 lpd, 292
 lpq, 302
 lpr, 301
 lprm, 302
 lpstat, 303
 LSH, 281
 luzne trasy zrodlowe, 488
- L**
- ladunek, 116
 lacze, 310, 326
 lacze wirtualne, 407
 laczenie podsieci, 390
 lacznica, 430
- M**
- MAC, 29, 53, 93, 250, 368
 macierz dyskowa, 425
 Macintosh, 143, 277, 287, 326
 magistrala, 77
 magistrala rozgloszeniowa, 78
 Mail Transfer Agent, *Patrz* MTA
 Mail User Agent, *Patrz* MUA

- mailing list, 315, 339
Maksimum Transmit Segment Size, *Patrz* MTSS
maksymalny rozmiar segmentu, 137
man, 301, 485
MAN, 30, 367, 429
mapa samochodowa, 102
marshalling, 154
maska podsieci, 101, 107, 173, 191, 392, 403, 406
maska sieci, 111, 160
master, 91
MAU, 390
Maximum Transmission Unit, *Patrz* MTU
MCSD, 15
MCSE, 15
MD5, 250, 451, 459
MDA, 330
Media Access Control, *Patrz* MAC
Menedżer uwierzytelniania, 287
Mesh, 34
message, 61, 336
Message Digest 5, 451
Message list, 335
metaplik, 326
metoda karuzelowa, 209
metoda odpytywania, 91
metoda sygnalizacji, 66
metoda sztafetowa, 91
metropolitan area network, *Patrz* MAN
metryka, 111, 404
MIB, 499, 500
Microsoft Certified Systems Engineer, *Patrz* MCSE,
Patrz MCSE
Microsoft DHCP Manager, 190
Microsoft Exchange, 483
Microsoft Internet Mail, 315
Microsoft Outlook Express, 329, 334
Microsoft Point-to-Point Encryption, *Patrz* MPPE
Microsoft Terminal Server, 285, 287
Microsoft Transaction Server, *Patrz* MTS
Microsoft Word, 313
MIME, 315, 333, 336
MindTerm SSH, 281
modem, 52, 74, 233, 442
modulacja, 74
modulo, 132
modul dodatkowy, 326
modul namiastkowy, 154
monitor, 273
monitor sieci, 497
monitor wydajności, 484
monitorowanie sprzętu, 482
montowanie, 163
Mosaic, 306
MOSPF, 121
most, 53, 368
most sieciowy, 37
mountd, 257
MPPE, 252
MS SQL, 416
MSAU, 34
MS-DOS, 287
msg-auth, 145
msp, 145
MSS, 137, 139
MTA, 330
MTS, 313
MTSS, 487
MTU, 114, 404
MUA, 330
Multi Station Access Unit, *Patrz* MSAU
multicast, 111
Multicast Extensions to OSPF, *Patrz* MOSPF
multicast packet, 42
multicasting, 120, 403
multihoming, 421
multipart, 336
multipleksowanie, 70, 127
Multipurpose Internet Mail Extensions, *Patrz* MIME
- N**
- n, *Patrz* MDA
nadsiec, 381
nagłówek, 25, 60, 89, 115, 309
nagłówek TCP, 139
nagłówek UDP, 139
nagłówek uwierzytelniający, *Patrz* AH
najczęściej zadawane pytania, *Patrz* FAQ
name, 145
name resolution, 161
narzędzia monitorujące, 484
NAT, 226, 242, 506
NAT dynamiczny, 244
NAT nakładany dynamiczny, 244
NAT przeciążony dynamiczny, 244
NAT statyczny, 243
nazwa domeny, 160
nazwa hosta, 159
nazwa kanoniczna, 198
nazwa komputera, 159
nazwa wyróżniająca, 348
NBNS, 216, 478
nbtstat, 479, 493
NCP, 45, 306
NCR, 505
NCSA, 306
NDS, 23, 347, 358
neighbour process, 408
Neophone, 505
net, 218, 343

- net send, 480
 net view, 480
 NetBEUI, 253
 NetBIOS, 40, 44, 412, 474, 477
 NetBIOS Name Server, *Patrz* NBNS
 netbios-dgm, 146
 netbios-ns, 146
 netbios-ssn, 146
 netiquette, 343
 Netscape Messenger, 315, 331, 334
 Netscape Navigator, 307
 netstat, 488, 489
 netwall, 146
 NetWare, 215
 Network Access Layer, 85
 Network Address Translation, *Patrz* NAT, *Patrz* NAT
 Network Basic Input Output System, *Patrz* NetBIOS
 Network Basic Input -Output System, *Patrz* NetBIOS
 Network Control Protocol, *Patrz* NCP
 Network Directory Service, 345
 Network File System, *Patrz* NFS
 network gateway, 38
 Network Interface Card, *Patrz* NIC, *Patrz* NIC
 Network Interface Layer, 85
 Network layer, 46
 Network News Transfer Protocol, *Patrz*NNTP
 network redirector, 27
 NETWORKING, 163
 netykieta, 343
 newsgroup, 339
 newsgroups, 315
 NFS, 255
 nfsd, 257
 NIC, 22, 31, 36, 53
 nickname, 145, 159
 nieceka, 66
 NIIT, 15
 NIS, 347, 353
 NIS namespace, 353
 NIS+, 347, 355
 nntp, 146
 NNTP, 329, 342, 452
 node, 42, *Patrz* wezel
 Nokia Communicator, 505
 non-routable, 227
 Nortel Networks, 110
 nosnik fizyczny, 429
 nosnik transmisji, 47, 89
 notacja dwójkowa, 102
 notacja dziesietna, 102
 Notatnik, 315
 Novell NetWare eDirectory, 23
 Novell Networks, 24
 NT Directory Service, 360
 NTDS, 360
 NTFS, 261, 264
 NTLM, 453
 ntp, 146
 ntsysv, 292
 numer potwierdzenia, 62, 140
 numer sekwencji, 140
 NVT, 273
- O**
- Oakley Quick Mode, 462
 obiekt uzytkownika, 358
 obszar, 457
 obwod wirtualny, 83
 obwód wirtualny, 432
 OC12, 434
 ochrona, 133
 oczekiwanie na transmisyje, 136
 oczekiwanie na zakonczenie polaczenia, 135
 oczekiwanie na zamkniecie, 136
 odciski palca, 324
 odpornosc, 133
 odpowiedz, 311
 odpowiedz echo, 118
 odpytywanie, 54
 odsylacz, 326
 odwzorowanie wejsciowe, 243
 oglaszanie rutera, 397
 okno nadawania, 137
 okno odbioru, 137
 okno przeciazenia, 138
 okno przesuwne, 137
 okno TCP/IP, 28
 oktet, 93, 104
 on-demand streaming, 325
 online, 434
 OP, 183
 open computing, 45
 Open Connection, 277
 Open Shortest Path First, *Patrz* OSPF, *Patrz* OSPF
 Open Systems Interconnection, *Patrz* OSI
 OpenSSH, 281
 opóźnienie nawiazania polaczenia, 133
 opóźnienie przejścia, 133
 opóźnienie zwolnienia polaczenia, 133
 Oracle, 416
 OS/2, 143
 osadzanie, 326
 OSI, 24, 31, 46, 155, 347
 warstwa aplikacji, 26
 warstwa fizyczna, 29
 warstwa lacza danych, 29

warstwa prezentacji, 26
warstwa sesji, 27
warstwa sieciowa, 28
warstwa transportowa, 28
OSPF, 29, 114, 397, 403, 405
ostatnie potwierdzenie, 136
otrzymane segmenty, 492

P

packet filter firewall, 237

Packet Internet Groper, *Patrz PING*

Packet InterNet Groper Utility, *Patrz PING*

packet sniffer, 447

packet switching, 306

Padding, 461

pakiet, 182

pakiet danych, 24

pakiet DHCP, 190

pakiet protokołów, 45

pakiet SYN odebrany, 136

pakiety grupowe, 42

pamiec, 416

pamiec podreczna, 247

PAN, 30

Panel sterowania, 169, 170

PAP, 252

parallel line printer, 292

PCM, 67

PCNFS, 279

pcnfsd, 258

peer-to-peer, 23

pełnodupleksowa transmisja danych, 132

Permanent Virtual Circuit, 432, *Patrz PVC*

personal area network, *Patrz PAN*

Personal Web Server, *Patrz PWS*

PGP, 323, 339, 454, 457

Phase Shift Keying, *Patrz PSK*

Physical layer, 46

PI, 266

pieczetowanie wiadomosci, 450

pierscien, 50

PIM, 121

PIN, 338

ping, 167, 387, 468, 487

PING, 42, 118

PKI, 449, 450, 453, 457

PLEN, 96

plik strefy, 207

plik urzedzenia drukarki, 292

plik wskazówek głównych, 203

PLMN, 509

plug-and-play, 179

plug-in, 326

pobierz wiadomosci, 335

poczta elektroniczna, 315

podcerwien, 75, 76

poddomeny, 205

podpis cyfrowy, 250, 324, 339

podpisywanie wiadomosci, 449

podsiec, 42, 184, 375, 390

podszywanie sie, 282

podzial czasu, 69

podzial częstotliwosci, 69

podzial horyzontu, 400

podzial pasma, 68

Point-of-Presence, *Patrz POP*

Point-to-Point Protocol, *Patrz PPP*

point-to-point tunneling protocol, *Patrz PPTP*

Point-to-Point Tunneling Protocol, *Patrz PPTP*

poisoned reverse, 401

polling, 91

polaczenie, 311

polaczenie dwupunktowe, 48, 437

polaczenie kanalu sterujacego, 266

polaczenie nawiazane, 135, 136

polaczenie wielopunktowe, 48

polaczenie zamkniete, 135, 136

POP, 249, 330, 332, 452

pop2, 145

POP2, 332

pop3, 145

POP3, 44, 330

port, 40, 144

port docelowy, 132

port protokolu, 130

port zrodlowy, 132

portmapper, 258

porzadkowanie, 127

POST, 311

Post Office Protocol, *Patrz POP3*

potok, 150

potokowa transmisja obrazu, 325

potrójny DES, *Patrz 3DES*

potwierdzenie trójkierunkowe, 135

poufnosc, 249

Powielanie, 127

PPP, 233, 251

PPTP, 251, 395, 439

praca zdalna, 442

prawdopodobienstwo niepowodzenia nawiazania

polaczenia, 133

prawdopodobienstwo niepowodzenia przesylu, 133

prawdopodobienstwo niepowodzenia zwolnienia

polaczenia, 133

preamble, 87

Pretty Good Privacy, *Patrz PGP*

print queue, 293

printer, 146

printtool, 297

- priority, 133
 priorytet, 133
 priorytet na zadanie, 32
 proces znajdowania sąsiadów, 408
 procesor, 416
 prosty protokół przesyłania poczty, *Patrz* SMTP
 prosty protokół transferu plików, *Patrz* TFTP
 protection, 133
 Protocol Independent Multicast Protocol, *Patrz* PIM
 protocol suite, 45
 protokół, 26
 protokół dwupunktowy, *Patrz* PPP
 protokół informacyjny trasowania, *Patrz* RIP
 protokół interfejsu łącza szeregowego, *Patrz* SLIP
 protokół komunikacyjny sterowania siecią Internet, *Patrz* ICMP
 protokół kopiowania pomiędzy komputerami unikowymi, *Patrz* UUCP
 protokół przesyłania hipertekstu, *Patrz* HTTP
 protokół rozwiązywania adresów ATM, *Patrz* ATMARP
 protokół rozwiązywania nazw, *Patrz* ARP
 protokół systemu katalogowego, *Patrz* DSP
 protokół transferu plików, *Patrz* FTP
 protokół trasowania grupowego na podstawie wektorów odległości, *Patrz* DVMRP
 protokół tunelowania dwupunktowego, *Patrz* PPTP
 protokół tunelowania w warstwie 2., *Patrz* L2TP
 protokół urzędu pocztowego, *Patrz* POP
 protokół wykorzystania najkrótszej ścieżki, *Patrz* OSPF
 protokół zarządzania grupami internetowymi, *Patrz* IGMP
 protokół zdalnego kopiowania, *Patrz* rcp
 proxy, 238, 246
 prywatne adresy IP, 227
 przeglądarka, 307, 309
 przekazywanie ramki, 69
 przekazywanie zetona, 54
 przekierowanie, 118
 przekierowywanie danych, 27
 przekierowywanie zadań, 26
 przelaczanie komórek, 433
 przełącznik, 33, 37, 68, 368
 przepustowość, 30, 73, 92, 133
 przeskok, 383, 385
 przeskoki częstotliwości, 77
 przestrzeni nazw NIS, 353
 przestrzeń nazw, 202
 przesunięcie czasu, 191
 przesył danych, 126
 przewód elektryczny, 47
 przyrost, 377
 ps, 486
 PSH, 62, 140
 PSK, 67
 PSTN, 251
 pstree, 486
 PTR, 209
 Public Key Infrastructure, *Patrz* PKI
 pulpa adresów zakresu, 190
 pull, 479
 pulpit zdalny, *Patrz* RDP
 Pulse Code Modulation, *Patrz* PCM
 punkt kontrolny, 27
 punkt obecności, 249
 push, 479
 Putty, 281
 PVC, 249
 PWDump, 447
 PWS, 313
- ## Q
- qmtp, 146
 QoS, 64, 133
 qotd, 145
 Quality of Service, *Patrz* QoS
 QVTTerm, 276
- ## R
- R, 351
 radio, 76
 radio wąskopasmowe, 75
 RADIUS, 250, 253
 RAID, 482
 RAM, 230, 482, 483
 ramka, 75
 RARP, 94, 96, 181
 RAS, 251
 RAW, 295
 RC2, 451
 RC4, 451
 RC5, 451
 rcp, 268
 rcvfrom, 150
 RDN, 348
 RDP, 285, 288
 Real Audio, 327
 realm, 457
 Red Hat, 292
 RedHat, 163
 regenerator, 36, 52
 Rejestr, 503
 rekin, 22
 rekord poczatku pełnomocnictwa, *Patrz* SOA
 rekord zasobu, 198
 rekordy wskazników, 209
 Relative Distinguished Name, *Patrz* RDN
 relay agent, 193

- Remote Authentication Dial-In User Service, *Patrz*
RADIUS
Remote Copy Protocol, *Patrz* rcp
Remote Desktop, *Patrz* RDP
Remote execute, *Patrz* rexec
Remote login, *Patrz* rlogin
Remote Procedure Call, *Patrz* RPC
Remote shell, *Patrz* rsh
repeater, 36
replay attack, 460
replikacja, 348, 358, 422
reply message, 256
Request for Comments, *Patrz* RFC
residual error rate, 133
resilience, 133
resolwer, 202, 210
retransmitowane segmenty, 492
Reverse Address Resolution Protocol, *Patrz* RARP
rexec, 271, 284
RFC, 38
RG-58, 71
RIAddr, 184
RIP, 29, 114, 131, 396, 398, 405
RIPE-MD-160, 250
Rivest Shamir Adlemann, *Patrz* RSA
RJ-45, 52
rlogin, 271, 278
r-narzedzie, 271
ROM, 512
root, 144, 162, 281, 303, 348
root hints, 203
root servers, 203
round robin, 209, 426
route, 110, 165, 389, 467
 add, 389
 delete, 389
 flush, 389
 modify, 389
 print, 389
route discovery, 57
route print, 468
route tag, 402
router, 37
router programowe, 110
routing, 29, 383
Routing Information Protocol, *Patrz* RIP, *Patrz* RIP
routing table, 57
rozmiar okna, 134
rozmiar okna odbioru, 141
rozproszony system plików, *Patrz* DFS
rozszerzalność, 359
rozszerzenie protokołu OSPF o adresowanie
 grupowe, *Patrz* MOSPF
rozwiązywanie adresu, 94
RPC, 143, 151, 153, 256, 295
rpcinfo, 155, 261
RPS, 149
RSA, 283, 323
rsh, 271, 280
RST, 62, 140
rsync, 146
rtelnet, 145
ruch sieciowy rozgłoszeń, 37
ruter, 30, 37, 57, 99, 114, 192, 235, 384, 390, 401,
 406
ruter brzegowego systemu autonomicznego, 408
ruter brzegowy obszaru, 407
ruter desygnowany, 409
ruter wewnętrzny, 407
r-utilities, 271
rywalizacja, 53
- S**
- S/MIME, 316, 337
saft, 146
SAMBA, 27, 214, 279
SCO, 149
scope, 186
SCSI, 482
search engine, 314
Secure Hash Algorithm, *Patrz* SHA-1
Secure shell, *Patrz* ssh
Secure Shell, 281
Secure Sockets Layer, *Patrz* SSL
Secure/MIME, 316
Secure/Multipurpose Internet Mail Extensions, *Patrz*
 S/MIME
SecureSSH, 324
Security Accounts Manager, 447
Security Association, 461
segment, 30, 58, 135, 366, 421, 430
segment SYN wysłany, 135
segment synchronizacji, 134
send, 150
Send, 331
send window, 137
sendto, 150
Sequence Packet Exchange, *Patrz* SPX
Serial Line Interface Protocol, *Patrz* SLIP
Server Message Block, *Patrz* SMB
server-side, 320
Service Pack, 322
serwer, 23, 134
serwer nazw, 202
serwer specjalistyczny, 23
serwer terminali, 284, 288, 443
serwery buforujące, 424
serwery hierarchiczne, 422
Serwlet, 321

- sftp, 281
 SFU, 261
 SGML, 308, 317
 SHA, 250
 SHA-1, 451
 Shielded Twisted Pair, *Patrz* STP
 Short Message Service, *Patrz* SMS
 SIAAddr, 184
 Sieci lokalna, *Patrz* LAN
 sieciowe uslugi katalogowe, 345
 sieciowy programu przeadresujacy, 27
 sieciowy system plików, *Patrz* NFS
 siec, 416
 siec komórkowa, 76
 siec komputerowa, 21
 siec lokalna, 30
 siec miejskia, 30
 siec osobista, 30
 siec prywatna, 230
 siec prywatna, 227
 siec publiczna, 230
 siec rozlegla, 30, *Patrz* WAN
 siec równorzędnaorzedne, 22
 siec scentralizowana, 23
 siec szkieletowa, 392, 413, 440
 siec zdecentralizowana, 22
 Simple Mail Transfer Protocol, *Patrz* SMTP, *Patrz* SMTP
 Site Analyst, 313
 skalownosc, 359
 skretka, 72, 92
 skrypt, 319
 skrzynka odbiorcza, 334
 SlackWare, 163
 sliding window, 137
 SLIP, 233
 slowo kluczowe, 309
 Smart Card, 287
 SMB, 214, 293, 295
 SMS, 509
 smtp, 145
 SMTP, 26, 27, 44, 60, 143, 329, 330, 331, 452
 smux, 146
 snmp, 146
 SNMP, 60, 131, 329, 498
 snmptrap, 146
 snoop, 495, 498
 spp, 146
 SOA, 208
 SOCK_DGRAM, 152
 SOCK_STREAM, 152
 socket, 40, 132, 147, 150, 152
 Sockets, 197
 socks, 146
 software router, 110
 Solaris, 17, 281
 SoundEdit, 326
 SoundForge, 326
 Source Quench, 492
 Source Service Access Poin, *Patrz* SSAP
 split horizons, 400
 spolecznosc, 499
 spoofing, 282
 spooling, 292
 SPX, 38
 SQL Server, 366
 SRI-NIC, 196
 SSAP, 88
 ssh, 145, 271, 281
 SSH1, 281
 SSH2, 281
 SSL, 44, 323, 324, 458
 stan inicjacji, 187
 stan polaczenia, 57
 stan wyboru, 187
 stan zadania, 187
 standard Ethernet, 71
 Standard Generalized Markup Language, *Patrz* SGML
 standard uniwersalnej magistrali szeregowej, *Patrz* USB
 standardowe algorytmy szyfrowania, 450
 standardowy uniwersalny jezyk znakowania, *Patrz* SGML
 stany TCP, 135
 Start of Authority, *Patrz* SOA
 statd, 258
 stateful, 126
 stateful packet inspection firewall, 238
 statyczny wybór tras, 57, 388
 statystyki ICMP, 491
 statystyki IP, 491
 statystyki TCP, 492
 statystyki UDP, 492
 STDA, 356
 STDS, 347, 356
 sterowanie przeplywem, 54, 127
 sterowaniem przeplywem, 59
 sterownik, 160
 stopa bledów, 69, 133
 store-and-forward, 68
 STP, 72
 strategia dzierzawy, 188
 STREAM, 150
 streaming, 325
 StreetTalk, 356
 StreetTalk Directory Assistant, *Patrz* STDA
 strefa, 205
 strefa podstawowa, 205
 strefa wtórna, 205

- strefa wyszukiwania w przód, 206
strefa zdemilitaryzowana, 241
strefy wyszukiwania wstecz, 207
stub module, 154
Subject, 331
subnetting, 375
suma kontrolna, 63, 64, 67, 132
Sun, 153
Sun Ray, 271, 285
Sun Ray appliance, 285
SunOS, 149, 258
sunrpc, 145
supernetting, 381
super-thin client, 287
switch, 68
Switched Virtual Circuit, 432
sygnal analogowy, 47
sygnal cyfrowy, 47
SYN, 62, 134, 141
SYN segment, 134
SYN_RCVD, 136
SYN_SENT, 135
SYNACK segment, 134
systat, 145
system rozproszony, 153
system wieloadresowy, 421
szacowanie ryzyka, 445
szkielet, 49, 77
szybkosc transmisi, 70
szyfrowanie, 26, 323, 439
szyfrowanie kluczem symetrycznym, 450
- S**
- sciana przeciwpozarowa, 234
swiatlowod, 36, 73
swiatlowód, 47, 92
- T**
- tablica tras, 110, 384, 398
tacacs, 145
TACACS, 235
tag, 308
talk, 146
TCP, 62, 116
 nagłówek, 62
TCP/IP, 21, 40, 45, 85, 105, 160, 255, 266, 271, 329, 331, 363, 459, 464
 warstwa aplikacji, 40, 59
 warstwa fizyczna, 47
 warstwa interfejsu sieciowego, 42, 53
 warstwa internetowa, 41, 56
 warstwa transportowa, 41, 57
tcpmux, 145
TDM, 69
- TELEcommunication NETwork, *Patrz Telnet*
telewizja kablowa, 434
telnet, 145
Telnet, 44, 60, 195, 271, 272, 316, 442, 452
Telnet wirtualny terminal sieciowy, 273
Telnet zasada negocjacji, 273
temat, 331
Terminal Access Controller Access Control System, *Patrz TACACS*
Terminal server, 284
Terminal Server, 288
terminator, 71, 79
text, 336
tftp, 145
TFTP, 60, 128, 182, 268
thicknet, 71
thin client, 284, 289
thin Ethernet, 70
thinnet, 70
three-way handshake, 28, 135
throughput, 133
ticket-granting ticket, 456
TIFF, 311
time, 145
Time Division Multiplexing, *Patrz TDM*
time to live, *Patrz TTL*
TIME_WAIT, 136
timestamp, 492
TI-RCPs, 149
TI-RPC, 151
TI-RPCs, 143
TLD, 203
TLI, 143, 149
tlumienie, 73
token, 33, 54, 79, 91
Token Bus, 31
Token Ring, 33, 65, 79, 86, 91, 97, 430, 494
top, 486
Top Level Domain, *Patrz TLD*
top-level domains, 202
topologia, 22, 406
topologia fizyczna sieci, 48
topologia Gwiazda-magistrala, 35
topologia Gwiazda-pierscien, 35
topologia gwiazdy, 33, 49
topologia magistrali, 32, 49
topologia oczkowa, 34, 50
topologia pierscienia, 33, 50
topologia sieci, 32
topologie hybrydowa, 34
traceroute, 119, 166, 436, 472, 488
tracert, 119, 436, 472, 488
Tracert, 118
transfer failure probability, 133
transfer stref, 420, 423

- transit delay, 133
 transmisja analogowa, 66
 transmisja asynchroniczna, 74
 transmisja cyfrowa, 67
 transmisja danych, 429
 transmisja synchroniczna, 74
 transmisja szerokopasmowa, 68
 transmisje grupowe, 120
 transmisji potokowej na zadanie, 325
 Transmission Control Protocol, *Patrz* TCP/IP
 trap, 499
 trasa domyslna, 385
 trasowanie, 101, 109, 111, 383, 395, 403
 trasowanie bezklasowe, 109
 trasowanie DHCP, 192
 trasowanie klasowe, 109
 trasowanie rozplywowe, 406
 trasy domyslne, 110
 trasy dynamiczne, 114
 trasy statyczne, 113
 triggered updates, 402
 Triple DES, *Patrz* 3DES
 Trivial File Transfer Protocol, *Patrz* TFTP, *Patrz* TFTP
 troubleshooting, 463
 trójkierunkowe potwierdzenie, 28
 trójnikи, 52
 trusted host, 162
 tryb bramowy, 161
 TTL, 42, 59, 64, 113, 115, 121, 397, 424, 472
 TTSSH, 281
 tunel IP, 252
 tunelowanie, 251, 253, 434, 438
 typ danych surowych, 295
 typ protokolu, 96
 typ sprzetu, 96
 typ tras, 110
 typ uslugi, 40, 115
- U**
- UCB, 271
 UCT, 191
 udostepnianie polaczenia internetowego, 247
 UDP, 28, 58, 116, 130, 459
 ulistproc, 146
 ultralekki klient, 287
 unicast, 120
 Uniform Resource Locator, *Patrz* URL, *Patrz* URL
 Universal Coordinated Time, *Patrz* UCT
 Universal Serial Bu, *Patrz* USB
 uniwersalne rozszerzenia poczty internetowej, *Patrz* MIME
 Unix, 143, 149, 215, 281
 Unix-to-Unix Copy Protocol, *Patrz* UUCP
- unmarshall, 154
 unnamed root, 202
 Unshielded Twisted Pair, *Patrz* UTP
 uproszczony protokol dostepu do katalogu, *Patrz* LDAP
 uprzatania pamieci, 402
 uptime, 485
 urd, 146
 URG, 62, 140
 URL, 235, 307, 311
 USB, 286
 User Datagram Protocol, *Patrz* TCP/IP, *Patrz* UDP
 user object, 358
 usluga sieciowa, 27
 uslugi polaczeniowe, 58
 UTP, 72
 UUCP, 233
 uwierzytelnianie, 249, 402
 uwierzytelnianie uzytkowników, 451
 uzytkownik, 274
- V**
- Variable Length Subnet Masking, *Patrz* VLSM
 VAX, 153
 VBScript, 321
 VC, 93
 VCI, 93
 VeriSign, 454, 458
 vi, 164
 video, 336
 Virtual Channel, *Patrz* VC
 Virtual LAN, 37
 Virtual Private Network, *Patrz* VPN, *Patrz* VPN
 VLSM, 392
 VPI, 93
 VPN, 226, 235, 248, 323, 439
 funkcja mieszania, 250
 integralnosc, 250
 kontrola dostepu, 250
 podpisy cyfrowe, 250
 poufnosc, 249
 uwierzytelnianie, 249
- W**
- W3C, 305, 311, 318
 WAE, 511
 waga opóznien, 404
 waga przepustowosci, 404
 WAIS, 306
 walking the tree, 210
 WAN, 30, 367, 430
 WAP, 510
 warstwa aplikacji, 26, 40, 47, 59
 warstwa fizyczna, 29, 46, 47, 66

- warstwa interfejsu sieciowego, 42, 46, 53
warstwa internetowa, 41, 47, 56, 99
warstwa lacza danych, 29
warstwa prezentacji, 26
warstwa sesji, 27
warstwa sieciowa, 28
warstwa transportowa, 28, 41, 47, 57, 125
WDB, 103
Web TV, 505
WebCrawler, 314
webster, 146
wektor odleglosci, 57
well-known port number, 144
wewnętrzny protokół bramowy, *Patrz IGP*
weszyciel pakietów, 447
wezel, 22, 42, 147, 202, 286, 425
wezel hybrydowy, 219
white pages, 347
wiazanie, 147
Wide Area Information Server, 306
wide area network, *Patrz WAN*
Wideo, 325
wieloadresowosc, 421
wielouzytkownikowy rdzen serwera, 285
Windows 2000, 17, 23, 169, 255, 260, 275, 313, 439, 466
Windows 95, 110, 169
Windows 98, 169
Windows CE, 513
Windows Me, 169
Windows NT, 23, 101, 169, 215, 275, 313, 453
Windows Sockets, 149
Windows XP, 169
WinNuke, 447
WINOPCFG, 465
WINS, 173, 218, 412, 416, 465, 478
Winscp, 281
WinSock, 143, 149, 152
Wireless Application Environment, 511
Wireless Application Protocol, *Patrz WAP*
Wireless Datagram Protocol, 510
Wireless Session Protocol, 511
Wireless Transaction Protocol, 511
Wireless Transport Layer Security, 510
wirtualna siec lokalna, 37
wirtualne sieci prywatne, 248
wirus, 225
WML, 511
wodociag, 30
World Wide Web Consortium, 305
WS_FTP, 316
WSA, 152
wskaznik do pilnych danych, 141
wskaznik pilnosci, 63
wspolny interfejs bramy, *Patrz CGI*
- wspoloperatywnosc, 312
WTA, 511
WTAI, 511
WWW, 17, 26, 101, 119, 195, 247, 305, 370, 443, 446, 491, 505
wybor trasy, 29
wykrywanie trasy, 57
wymiana klucza, 283
wypelnienie, 461
WYSIWYG, 314
wyslane segmenty, 492
WYSWIG, 315
wyszukiwanie, 349
wyszukiwarka, 314
Wyslij, 331
wzgledna nazwa wyróżniajaca, 348
- X**
- X Window, 287
X.500, 347
XDR, 256, 257
XHTML, 312, 317, 318
XID, 183
XML, 312, 317
xns-courier, 146
XOR, 449
- Y**
- Yahoo, 314
yellow pages, 349
YIAddr, 183
- Z**
- zabezpieczanie sieci, 449
zabezpieczenie ladunku, *Patrz ESP*
zakotwiczenie, 309
zakres, 186, 189
zamkniece, 311
zapasowy ruter desygnowany, 409
zapora badajaca stan pakietów, 237
zapora filtrujaca pakiety, 236
zapora firewall, 234, 322
zapora hybrydowa, 236
zapora przejscia w warstwie aplikacji, 238
zasada IPSec, 462
zasada kontroli dostepu, 235
zasieg zalewów, 407
zatrucie zwrotu, 401
zaufane hosty, 162
zdalna kolejka uniksowa lpd, 299
zestawianie, 154
zewnętrzna reprezentacja danych, *Patrz XDR*
zlacze, 51

złącze programowe aplikacji, 148
zmienna środowiska, 279
znacznik, 308, 310
znacznik trasy, 402
znak nowego wiersza, 282
znak powrotu karetki, 267
znak przesuwu o wiersz, 267
znak ucieczki, 279

Z

źródłowy adres IP, 64

Z

zadanie, 311
zadanie echo, 118
zeton, 33, 54, 79, 91, 250

Uwagi po składzie

Rozdział 1. Brak wypunktowania "W tym rozdziale" str. 21

Rozdział 6. — rysunek 6.5. na tym rysunku po stronie serwera ma być "czas" a nie "time"
str. 134

Rozdział 8. Brak wypunktowania "W tym rozdziale" str. 159

Rozdział 11. — rysunek 11.5 – napis wychodzi poza ramkę str. 240