

Podstawy Sieci Komputerowych

Charakterystyka warstwy sieciowej. Podstawy doboru trasy w sieciach IP cz.2.

Trasowanie statyczne i dynamiczne. Protokoły RIP, OSPF i BGP. Protokół IPv6

dr hab. inż. Konrad Gromaszek

Plan wykładu

- Wprowadzenie
- Urządzenia
- Tablice routingu
- Typy routingu
- Wybrane protokoły routingu
- Protokół IPv6
- Protokół ICMP
- Protokół IGMP

Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Wprowadzenie

- Routing = trasowanie (pl)
- *algorytm wyznaczania trasy do transmisji pakietu*
- spolszczona wymowa
 - mechanizm – routing (ang.), ruting (pl_ang), trasowanie
 - urządzenie – router (ang.), ruter (pl_ang), trasownik

Router

- Router
 - zapewnia przesyłanie pakietów pomiędzy własnymi interfejsami
 - komputer z odpowiednim oprogramowaniem
 - zwykle – dedykowane urządzenia z dedykowanym OS
 - obsługuje ruch pakietów w warstwie sieci modelu ISO/OSI
 - dostarcza mechanizmy efektywnej i niezawodnej transmisji pakietów w sieci
 - umożliwia filtrowanie pakietów
 - realizuje fragmentację pakietów (IPv4)

Rodzaje protokołów

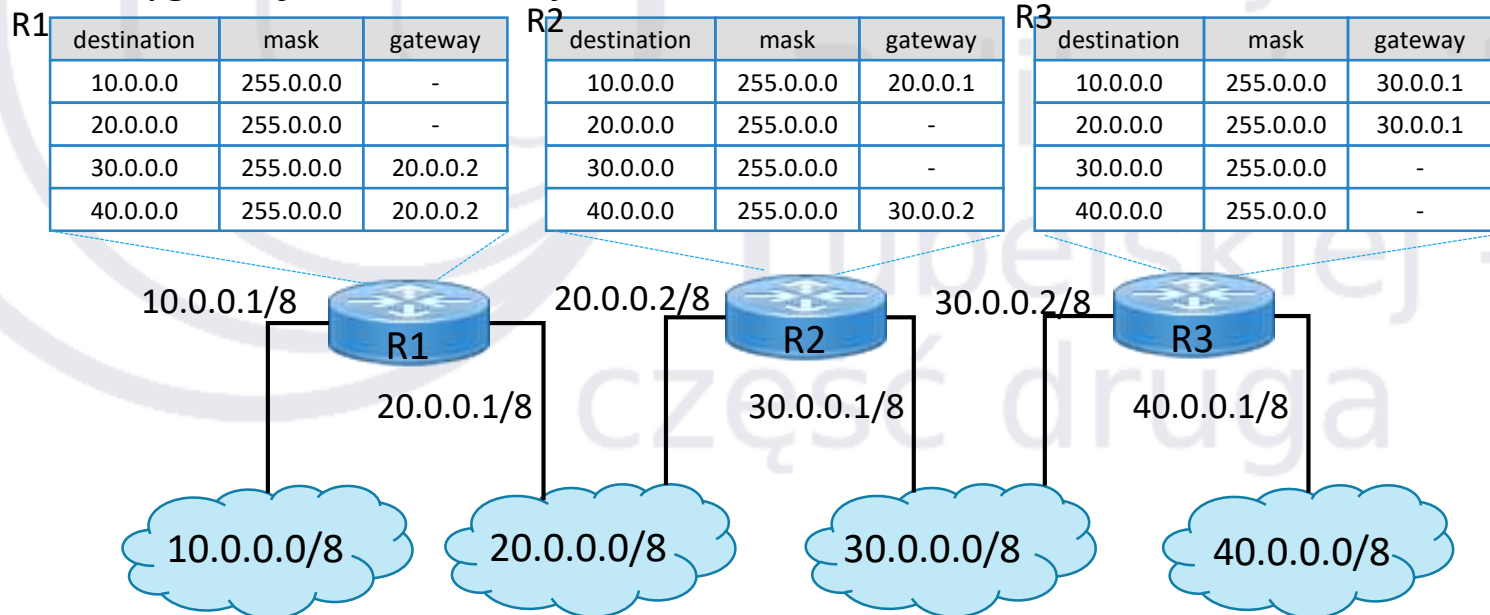
- Protokoły routowalne
- Zawierają informacje identyfikujące nadawcę i adresata
 - IP
 - Apple Talk
 - IPX
- Protokoły routujące
 - obsługują proces transmisji pakietu między urządzeniami sieciowymi
 - wybór odpowiedniej trasy dla pakietu
 - komunikacja między routerami – wymiana informacji o trasach

Typy routingu

- Routing statyczny
 - wpisy dotyczące tras dokonywane „ręcznie” przez administratora systemu
 - oparte o wiedzę i znajomość topologii sieci administratora
- Routing dynamiczny
 - trasy ustalane w oparciu o protokoły routingu
 - informacja o topologii sieci
 - informacja o zmianie tras

Tablica routingu

- Typ protokołu
- Odniesienie do punktu docelowego
- Metryka routingu
- Interfejs wyjściowy
- Czas wygaśnięcia/aktualizacji



Routing statyczny

- Routing statyczny
 - ustalany i wprowadzany przez administratora
 - stosowany w prostych/nieskomplikowanych sieciach
 - sprzętowy lub programowy
 - niekiedy stosowany razem z routingiem dynamicznym jako trasy zapasowe (wyższe metryki)
 - brak nadmiarowej komunikacji w sieci (ze wzgl. dynamiczną konfiguracją)
 - w przypadku awarii lub modyfikacji sieci, konieczność ręcznej interwencji
- Polecenia

Windows:

C:\users\Student\route print

route \[-p\] **add** <destination> **mask** <subnet mask> <gateway> **metric** <lowest number wins> **if** <interface>

Przykład:

route -p add 192.168.3.0 mask 255.255.255.0 10.0.0.1 metric 1 if 0x4

Linux:

ip route add / via dev X

route add/del -net <destination> netmask <mask> gw <gateway>

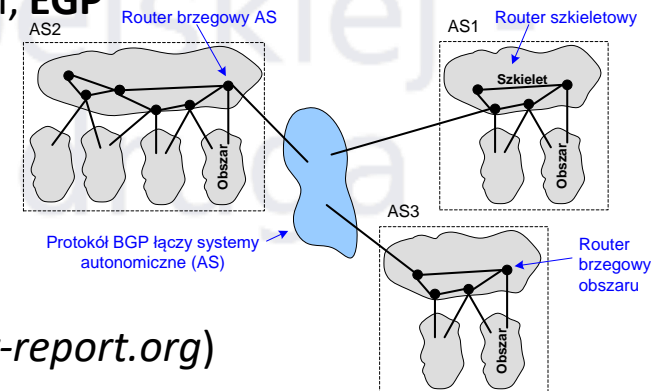
Przykład:

ip route add 10.10.10.0/24 via 192.168.1.1 dev eth0

route add -net 10.10.10.0 netmask 255.255.255.0 gw 192.168.1.1 dev eth0

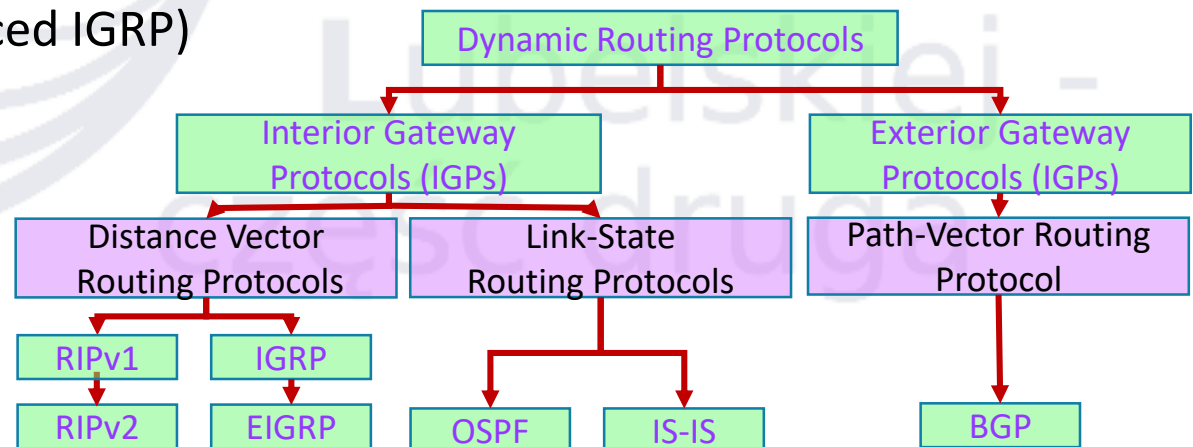
Routing dynamiczny

- **Autonomous system, AS** – system autonomiczny, opisany w dokumentach RFC 1771, RFC 1930 – zbiór adresów sieci IP pod wspólną kontrolą administracyjną, w którym utrzymywany jest spójny schemat trasowania (**routing policy**)
- Podział ze względu na zasięg działania
 - Protokoły wewnętrzne – Interior Gateway Protocol, **IGP**
 - Routing Information Protocol, **RIP** (v1/v2)
 - Open Shortest Path First, **OSPF**
 - Enhanced/Interior Gateway Routing Protocol, **EIGRP**
 - Protokoły zewnętrzne – Exterior Gateway Protocol, **EGP**
 - Border Gateway Protocol, **BGP**
 - Exterior Gateway Protocol, **EGP** (przestarzały)
- Numery AS są nadawane przez ICANN
 - 2 bajtowe (65 536 systemów) do 2007r.
 - 4 bajtowe po 2007
 - ok 70 tys. (na 29.03.2020, na podstawie www.cidr-report.org)



Routing dynamiczny – protokoły

- Otwarte
 - RIP (Routing Information Protocol)
 - OSPF (Open Shortest Path First)
 - BGP (Border Gateway Protocol)
- Własnościowe (CISCO)
 - IGRP (Interior Gateway Routing Protocol)
 - EIGRP (Enhanced IGRP)

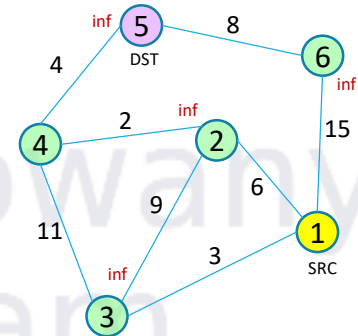


Wymagania stawiane protokołom routingu

- Optymalizacja – wybór ścieżki o najlepszych metrykach
- Szybkozbieżność – propagacja danych o zmianach tras routerów
- Elastyczność – uwzględnienie urządzeń o różnych parametrach, dynamika / zmienność warunków (przepustowość, opóźnienia)
- Odporność na błędy – obsługa awarii łączy
- Prostota i niski narzut – złożoność obliczeniowa, obciążenie sprzętu

Rodzaje routingu

- Podział ze względu na sposób wyznaczania trasy
 - Protokoły *wektora odległości* (***distance vector***)
 - routery wymieniają się gotowymi trasami
 - przesyłają informację o sieci docelowej wraz z jej kosztem (metryka)
 - metryka
 - prosta – liczba przejść przez kolejne routery do punktu docelowego (hop)
 - złożone/ zaawansowane – np. przepustowość, czas dostępu, koszt
 - stosunkowo prosty algorytm obliczania ścieżki (zaleta)
 - najkrótsza droga nie musi oznaczać najkrótszego czasu transmisji (wada)
 - Protokoły *stanu łącza* (***link state***)
 - uwzględniają „koszt” przeskoku
 - bazują na algorytmie Dijkstry
 - efektywniejsze algorytmy wyznaczania tras niż DVR (zaleta), ale także bardziej złożone obliczeniowo
 - routery znają całą topologię oraz wymieniają się informacjami o stanie łącza (samodzielne przeliczanie trasy)
 - Protokoły *hybrydowe* (łączą cechy dwóch powyższych)
 - Protokoły *path-vector*
 - opisują trasy przy użyciu atrybutów

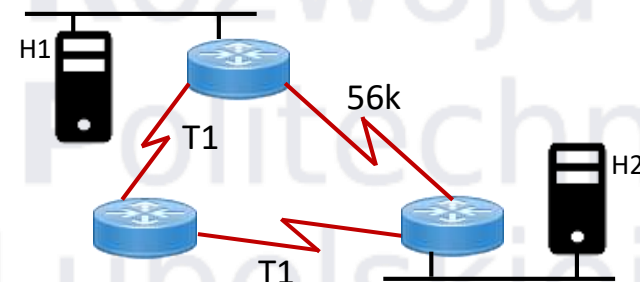


Metryka routingu

- **Metryka routingu**

- miara opisująca ***koszt*** transmisji pakietu określoną trasą
- abstrakcyjna ilościowa wartość, wskazująca odległość do danej sieci
- wartość liczbowa => *im mniejsza wartość, tym lepiej*

- Szerokość pasma
- Opóźnienie
- Obciążenie
- Niezawodność
- Liczba przeskoków
- Impulsy zegarowe
- Koszt



	Bandwidth	Adm. dist.
Bandwidth	Directly connected interface	0
	Static routes	5
Delay	OSPF internal routes	10
	IS-IS Level 1 Internal	15
	IS-IS Level 2 Internal	18
Hop count	RIP	100
Cost	Aggregate (route summary)	130
	OSPF external routes	150
	IS-IS Level 1 External	160
	IS-IS Level 2 External	165
	BGP	170

Protokół RIP

- Routing Information Protocol, RIP – zdefiniowany w RFC 1058 protokół bram wewnętrznych (IGP), w ramach jednego AS, należący do rodziny protokołów opartych o wektor odległości (*distance vector*)
- Aktualizacja tras:
- Rozgłaszana na adres broadcast
 - wysyłana co 30 sekund
 - metryka – liczba przeskoków
- Maksymalna liczba przeskoków na trasie – 15 (dla >16 przeskoków adres nieosiągalny)
- Ograniczenia:
 - nie wysyła informacji o masce podsieci (v.1)
 - nie obsługuje VLSM i CIDR
 - nie obsługuje uwierzytelniania

0	8	16	31
command(1)		version (1)	must be zero(2)
Address Family Ident. (3)		must be zero(2)	
IP address (4)			
must be zero(4)			
must be zero(4)			
metric(4)			

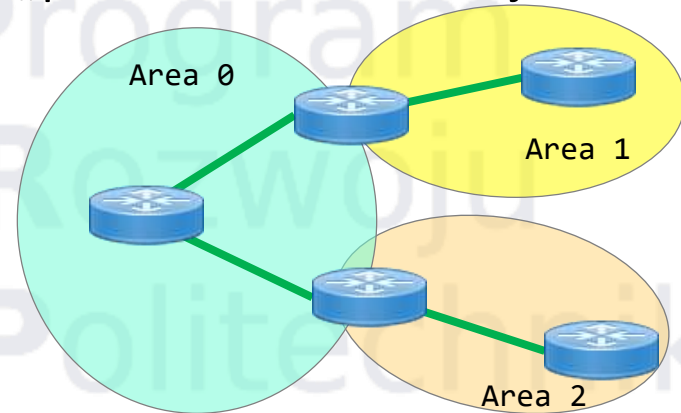
Protokół RIPv2


0	8	16	31
command(1)		version (1)	
Address Family Ident. (3)		route Tag(2)	
IP address (4)			
subnet mask (4)			
next hop(4)			
metric(4)			

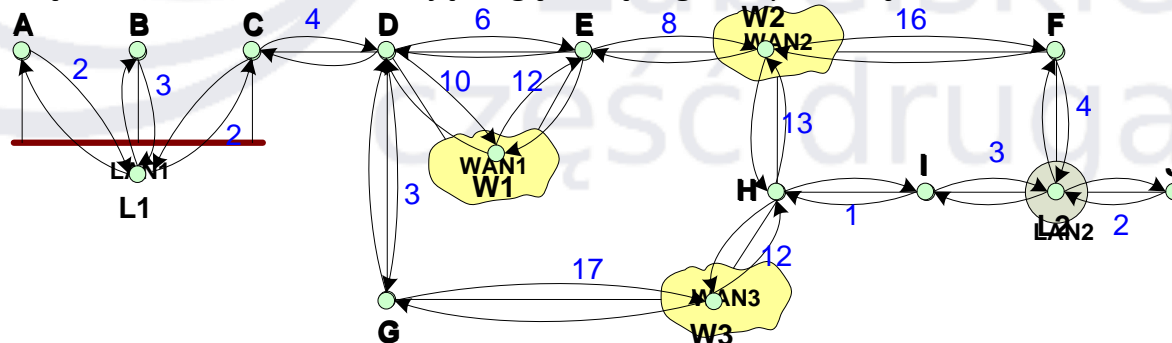
- Opracowany na początku lat 90., opisany w rekomendacji RFC 1723
- Obsługuje routing bezklasowy (VLSM)
- Przesyła informacje o masce podsieci
- Wysyła informacje uwierzytelniające
- Rozgłasza informacje wysyła na adres multicast (224.0.0.9)
- Przenosi informacje uzyskane za pomocą innych protokołów z sieci zewnętrznej
- (dodatkowe) pola ramki:
 - Route tag – informacja z routera wewnętrznego czy zewnętrznego
 - IP subnet mask – dla IP – maska podsieci
 - Next hop – adres routera następnego skoku

Protokół OSPF

- ***Open Shortest Path First, OSPF*** – tł.pl: „pierwszeństwo ma najkrótsza ścieżka”
- Opisany w RFC 2328
- Cechy
 - trasowanie najmniejszym kosztem
 - dobra skalowalność (w przeciwieństwie do RIP)
 - wybór optymalnej ścieżki
 - szybkozbieżny
 - wspiera *load balancing*
 - brak ograniczenia skoków do 15
 - przeznaczony do sieci zawierających do 500 routerów w obszarze trasowania
- Hierarchiczna struktura sieci
 - obszar zerowy (backbone) – Area 0
 - obszary podrzędne
 - eliminacja pętli dzięki wymianie tras pomiędzy obszarami za pomocą Area 0
 - zróżnicowanie algorytmów
 - wewnątrz obszaru – *link state* – (inf. o stanie łączy)
 - między obszarami – *distance vector* (gotowe trasy – wymieniane przez routery brzegowe)



- rów i łączy w graf skierowany, gdzie do ścieżka obliczana jest w oparciu o wagi: jednym w każdym kierunku samej sieci i po węźle na każdy router i są w grafie pominięte
- 



Protokół IGRP

- Interior Gateway Routing Protocol – protokół trasowania bramy wewnętrznej
- Cechy
 - Algorytm typu *distance-vector*
 - Metryka wykorzystywane przez routery do wyboru ścieżki
 - szerokość pasma
 - obciążenie
 - opóźnienie
 - niezawodność
 - Rozgłaszanie informacji o dostępności tras (wraz z parametrami łącz)
 - cykliczne – co 90 sekund
 - po zmianie stanu sieci
 - Protokół własnościowy Cisco
 - Brak wsparcia dla VLSM (zastąpiony przez EIGRP)

```
Router A> show interface serial 0
Serial0 is up, line protocol is up
Hardware is HD64570
Internet address is 172.16.4.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
```

```
Router B> show interface ethernet 0
Ethernet0 is up, line protocol is up
Hardware is Lance, address is 0010.7b80.bad5 (bia 0010.7b80.bad5)
Internet address is 172.16.2.1/24
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
```



Metryka

$$\text{IGRP}_{\text{metric}} = \{k1 \times \text{BW} + [(k2 \times \text{BW}) / (256 - \text{load})] + k3 \times \text{delay}\} \times \{k5 / (\text{reliability} + k4)\}$$
$$\text{IGRP}_{\text{metric}} = (10\,000\,000 / 1544) + (20000 + 1000) / 10$$
$$\text{IGRP}_{\text{metric}} = 6476 + 2100 = 8576$$

Protokół EIGRP

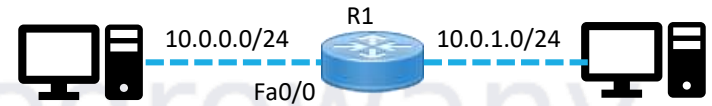
- **Enhanced Interior Gateway Routing Protocol, EIGRP**
- Hybrydowy protokół trasowania
 - wymienia informację jedynie z trasą i metryką
 - metrykę oblicza w oparciu o dodatkowe parametry
- Protokół własnościowy (Cisco) – sprzęt, licencja
- Cechy
 - mechanizm przeliczania tras: maszyna DUAL FSM (Diffused Update Algorithm Finite State Machine)
 - złożona metryka (Composite metric) uwzględniająca
 - przepustowość
 - obciążenie
 - opóźnienie
 - niezawodność
 - MTU
 - liczbę przeskoków
 - używany w sieciach do 50 routerów
 - płaska struktura sieci z podziałem na systemy autonomiczne
 - wykorzystuje protokół RTP do transportu pakietów
 - łatwa konfiguracja, obsługa VLSM, szybka zbieżność

```
RouterC# show ip eigrp topology 192.0.2.0/24
IP-EIGRP (AS 65535): Topology entry for 192.0.2.0/24
State is Passive, Query origin flag is 1, 7 Successor(s), FD is 13056
Routing Descriptor Blocks:
192.0.2.1 (Ethernet2/7), from 192.0.2.1, Send flag is 0x0
  Composite metric is (13056/12800), Route is External
  Vector metric:
    Minimum bandwidth is 500000 Kbit
    Total delay is 310 microseconds
    Reliability is 200/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
```

Metryka

$$\text{EIGRP}_{\text{metric}} = \{k1 \times BW_E + [(k2 \times BW_E) / (256 - \text{load})] + k3 \times \text{delay}\} \times \{k5 / (\text{Reliability} + k4) \times 256\}$$

Protokół EIGRP



- EIGRP
 - wykorzystuje dodatkowe 2 tablice:
 - tablice sąsiadów dostępne bezpośrednio/ pośrednio
 - tablica topologii trasy
 - dostępność tras sprawdza okresowo
 - aktualizacja transmituje zmiany (nie całą tablicę routingu) tylko zmiany
 - wymiana informacji wyłącznie przy zmianach (pełna tablica jedynie na początku)
 - wsparcie mechanizmu *load-balancing*
 - stosuje autentyfikację MD5 i SHA-2 między routerami
- komunikaty EIGRP:
 - typy: *Hello, Update, Query, Reply, Acknowledgement*
 - wartości metryk: 90 (dla AS), 170 (spoza AS)

```
R1# configure terminal
R1(config)# router eigrp 1
R1(config-router)# network 10.0.0.0 0.0.0.255
R1(config-router)# no auto-summary
```

R1# show ip protocols

```
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  Automatic network summarization is in effect
  Automatic address summarization:
    Maximum path: 4
  Routing for Networks:
    10.0.0.0/24
  Routing Information Sources:
    Gateway         Distance         Last Update
  Distance: internal 90 external 170
```

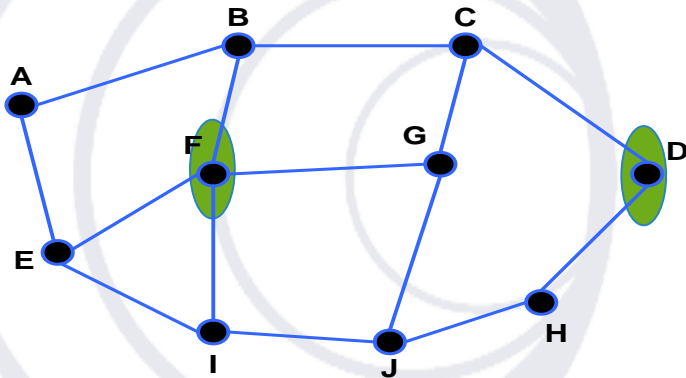
Protokół BGP

- **Border Gateway Protocol, BGP** jest protokołem routingu z rodziny DVR pomiędzy systemami autonomicznymi, opisany w RFC 1771, 1772, 1773, 1774, 1665), zaprojektowany, by w ruchu pomiędzy AS można było egzekwować wiele różnych zasad (w tym politycznych) routingu → stosunkowo wolnozbieżny
- Metryki zastąpione atrybutami + algorytm wyboru: pochodzenie ścieżki, ścieżka, adres następnego skoku
- Komunikaty (zestawienie sesji i wymiana informacji)
 1. OPEN – rozpoczyna zestawienie sesji BGP
 2. UPDATE – informacje o routingu (nieaktualne trasy, atrybuty ścieżki i NLRI)
 3. NOTIFICATION – wysyłany w przypadku błąd w nagłówku wiadomości
 4. KEEPALIVE – podtrzymuje sesję BGP, zeruje licznik HOLD TIME przy braku UPDATE
 5. ROUTE-REFRESH – obsługuje dynamiczne żądania odświeżenia tras

Protokół BGP

- a) tablica routingu F, używającego ścieżki FGCD by dostać się do D
- b) informacje o trasach od sąsiadów (pełne ścieżki)

a)



b)

Informacje które F otrzymuje od swoich sąsiadów na temat D:

Od B: „Ja używam BCD”
Od G: „Ja używam GCD”
Od I: „Ja używam IFGCD”
Od E: „Ja używam EFGCD”

- Po otrzymaniu ścieżek F sprawdza, która jest najlepsza (odrzuca od I, E)
- Router BGP zawiera moduł sprawdzania i oceny tras do danego celu, trasa naruszająca zdefiniowane zasady automatycznie otrzymuje wynik nieskończony
- Funkcja oceniająca odległość nie jest składnikiem protokołu ! (=> admin)
- Łatwość rozwiązywania problemu naliczania do nieskończoności (np.: awaria G lub linii FG: **BCD**, IFGCD)

Protokół BGP – konfiguracja

- Wymiana komunikatów na porcie 179 protokołu TCP
 - (niezawodność)
 - (niezbędne zestawienie sesji między routerami)
- Typy routingu
 - EBGp (exterior) – sesją między dwoma AS
 - IBGP (interior) – sesja między dwoma routerami brzegowymi jednego AS
- Po IBGP nie przesyłamy tras o których dowiedzieliśmy się z IBGP (zapobiegane pętlom)
- Administrative distance (stopień zaufania, mniej->lepiej)
 - 20 – EBGp
 - 200 – IBGP

Simple Internet Protocol Plus (IPv6)

- 1990 IETF podjął prace nad nową wersją IP i w RFC 1550 wydał prośbę o propozycje i dyskusje
- Główne założenia nowego protokołu:
 - obsługa miliardów hostów, nawet przy nieefektywnym przydziale adresów
 - zmniejszenie rozmiarów tablic routingu
 - uproszczenie protokołu, w celu szybszego przekazywania pakietów przez routery
 - zapewnienie wyższego bezpieczeństwa niż bieżący IP
 - podniesienie rangi typów usług głównie dla transmisji w czasie rzeczywistym
 - wspomaganie rozsyłania grupowego (możliwość definiowania zakresów)
 - możliwość przenoszenia hosta bez zmiany adresu
 - możliwość ewolucji protokołu w przyszłości
 - możliwość współistnienia starego i nowego protokołu przez szereg lat
- XII 1992 – wybrano 7 poważnych propozycji z 21
- 1993 – 3 najważniejsze propozycje opublikowano w IEEE Network, a po licznych poprawkach wybrano *Simple Internet Protocol Plus* (SIPP), oznaczone symbolem IPv6

Simple Internet Protocol Plus (IPv6)

- IPv6 całkiem dobrze spełnia postawione założenia RFC 2460 i 2466
- Nie jest zgodny z IPv4, ale zachowuje zgodność z innymi pomocniczymi protokołami Internetu, w tym TCP, UDP, ICMP, IGMP
- IPv6 posiada 16 bajtowe adresy
 - $2^{128} = 3,4 \times 10^{38}$ – 340 trylionów (gdzie IPv4: $2^{32} = 4,29$ mld)
 - ok. 6 mld/os => pokrycie Ziemi adresami IP = $6,7 \times 10^{38}/\text{m}^2$
- IPv6 ma uproszczony nagłówek do 7 pól (IPv4 – 13)
- IPv6 posiada lepszą obsługę opcji; w nowym nagłówku pola, które były uprzednio wymagane, są opcjonalne => routery mogą łatwiej pomijać nieistotne opcje, przyspieszając przetwarzanie pakietów
- Lepsze bezpieczeństwo, dzięki uwierzytelnianiu
- Przywiązano znaczną uwagę co do jakości usług (QoS)
- Konieczność zapewnienia zgodności – tunelowanie IPv6 w IPv4

IPv6 – budowa nagłówka

- Prostszy nagłówek 40B (=320b)
- Opcje dodatkowe w nagłówkach rozszerzających

(20b) pole dla pakietów wymagających oddzielnego traktowania (wymogach doręczania w czasie rzeczywistym; wykorzystywane do dostarczania multimediu)

(20b) pole dla pakietów wymagających oddzielnego traktowania nawiązywania pseudopołączeń pomiędzy źródłem a celem o określonych właściwościach/wymogach

(16b) wielkość pakietu bez nagłówka podstawowego (z ew. nagłówkiem opcjonalnym)

(8b) – typ następnego nagłówka (n. rozszerzający lub warstwy wyższej) odpowiada *Długość całkowita* z IPv4 (ale nie wlicza dł. nagłówka)

Wersja	Klasa ruchu	Etykieta przepływu		
Długość ładunku		Nastęny nagłówek	Limit przeskoków	
Adres źródłowy (16 bajtów)				
Adres docelowy (16 bajtów)				

(8b) odpowiada TTL – ilość przejść routerów przed odrzuceniem pakietu

(128b) 16 bajtowe adresy o notacji zapisu:

- osiem grup, rozdzielonych dwukropkami, po 4 cyfry szesnastkowe każda

1000:0000:0000:0000:0123:4567:89AB:CDEF

- możliwość optymalizacji zapisu (pomijanie zer)

1000::123:4567:89AB:CDEF

- również zapis w notacji dziesiętnej ::194:204:1:22)

IPv6 – nagłówki rozszerzające

- Nagłówek dodatkowy (*extension header*) – zapewnia uzupełnienie o potrzebne brakujące pola IPv4
- Może zawierać dodatkowe informacje, zakodowane w sposób efektywny
- Istnieje 6 zdefiniowanych, opcjonalnych typów nagłówków dodatkowych
- Opcjonalne, ale gdy w pakiecie > 1 to występują wszystkie, w wymienionej kolejności
- Nagłówki rozszerzające
 - Opcje skok po skoku
 - Routingu
 - Fragmentacja
 - Opcje docelowe
 - Nagłówek uwierzytelniania
 - Zaszyfrowany ładunek (Encrypted security payload)

Nagłówek typu skok po skoku (*hop-by-hop*)

Następny nagłówek	Długość nagłówka w B	Wielkość datagramu	n-bajtową liczbą def. wielkość
Długość ładunku jumbo (> 65 536 B)			

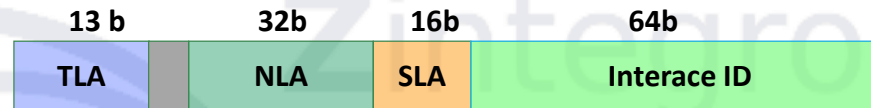
Nagłówek routingu

Następny nagłówek	Dł. nagłówka dodatkowego	Typ routingu	Zostało segmentów
Dane zależne od typu			

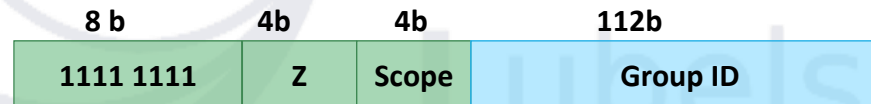
IPv6 – wydajność

- Uproszczona struktura nagłówka – optymalizacja przetwarzania przez routery
 - adres IPv6 = 4 * adres IPv4
 - nagłówek pakietu IPv6 = 2 * nagłówek pakietu IPv4
- Brak fragmentacji pakietów IPv6 → Path MTU lub pakiety <1280B
- Brak sumy kontrolnej nagłówka IPv6 → spójność nagłówek warstw wyższych
- Rozszerzenia dowolnej wielkości w IPv6 → Mniej pakietów kontrolnych
- Większy rozmiar pojedynczego pakietu → JumboFrame do 4GB
- Poprawa bezpieczeństwa → IPSec wymóg (w IPv4-opcja) + integracja zabezpieczeń (szyfrowanie i uwierzytelnianie)

Adres IPv6 – typy



- **Global Unicast** – identyfikator pojedynczego interfejsu (routowalny w Internecie)
 - pierwsze 64 bity – adres sieci (w tym 16b na podsieć) Pozostałe 64 bity – adres hosta
- **Unique Local** – odpowiednik adresu prywatnego IPv4
 - 8 bitów – FD (hex) + 40 bitów – dowolny adres sieci +16 bitów – adres podsieci + 64 bity – adres hosta
- **Anycast** (uniwersalny) – zbiór wielu interfejsów należących do różnych węzłów sieci; pakiet dostarczany do jednego interfejsu, najbliższego wg. metryki
- **Multicast** – identyfikuje grupę (zbiór interfejsów), gdzie pakiet przekazywany jest do każdego z interfejsów (ze zbioru, np.: wszystkie urządzenia FF02::1, routery FF02::2, OSPF: FF02::5 oraz FF02::6)



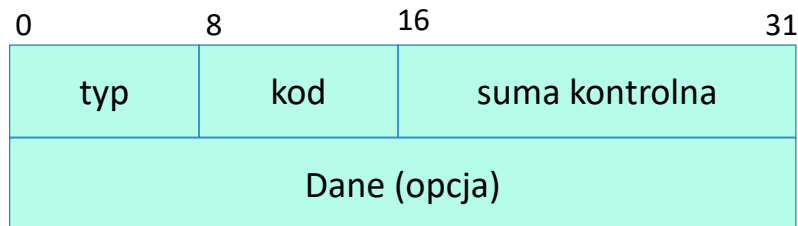
- interfejs sieciowy może mieć 1 lub kilka adresów multikastowych
 - oprócz pola **prefiksu FP** na adres multikastowy składają się:
 - **pole Z** (4b) ze znacznikami określającymi sposób przydziału adresu multikastowego
 - **pole S** (4b) określające zakres adresacji multikastowej (stacja, łącze, sieć)
 - **112b** identyfikator grupy multikastowej
- brak adresu rozgłoszeniowego (broadcast) => multicast + Scope=1

Adres IPv6 – typy

- **Anycast** to w efekcie transmisja typu 1-do-1zN (jeden do jednego z grupy)
- // Analogia do numeru ratunkowego //
 - adresacja anykastowa nie ma odrębnego formatu i wykorzystuje format adresacji unicastowej
 - adresy anykastowe można przydzielać z całej przestrzeni adresowej przeznaczonej dla unicastów
 - rozróżnienie pomiędzy any- a uni-kastami dokonywane jest na podstawie jawnego wskazania podczas konfiguracji interfejsu
- **Stateless address autoconfiguration (SLAAC)**
- Router dostarcza 64 bitowy prefix adresu + druga część adresu EUI-64
- Problem widoczności adresu MAC w internecie (zabezpieczenia: włączenie rozszerzeń prywatności + dynamiczny przydział IP dla urządzeń klienta)
- Specjalne pule adresowe:
 - `::/128` – adres zerowy, wykorzystywany tylko w oprogramowaniu
 - `::1/128` – adres pętli zwrotnej (odpowiednik loopback IPv4)
 - `::/96` – adresy kompatybilne z adresem IPv4 dla hosta korzystającego z IPv6 i IPv4
 - `::ffff:0:0/96` – adresy kompatybilne z adresem IPv4 dla hosta korzystającego wyłącznie z IPv4
 - `ff00::/8` – adresy typu link-local – wykorzystywane wewnątrz sieci lokalnych

Protokół ICMP

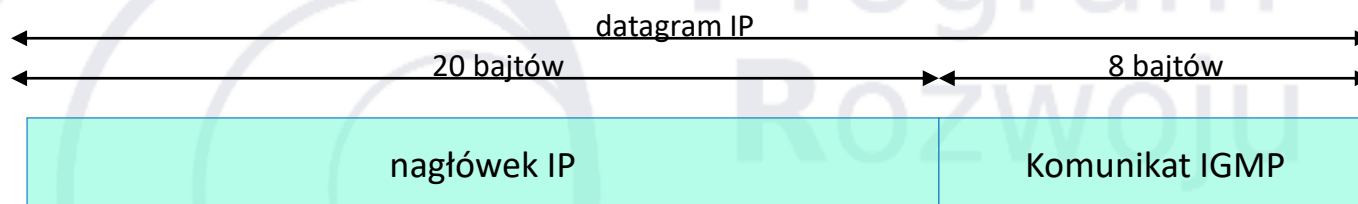
- **Internet Control Message Protocol, ICMP** – protokół komunikatów kontrolnych, opisany w RFC 792, należy do warstwy sieciowej modelu OSI i jest wykorzystywany w diagnostyce sieci oraz trasowaniu; pełni głównie funkcję kontroli transmisji w sieci
- jest wykorzystywany w programach ping oraz traceroute
- komunikaty ICMP przesyła w datagramie IP
- enkapsulacja do postaci pakietów IP -> do ramki warstwy drugiej
- ramka ICMP



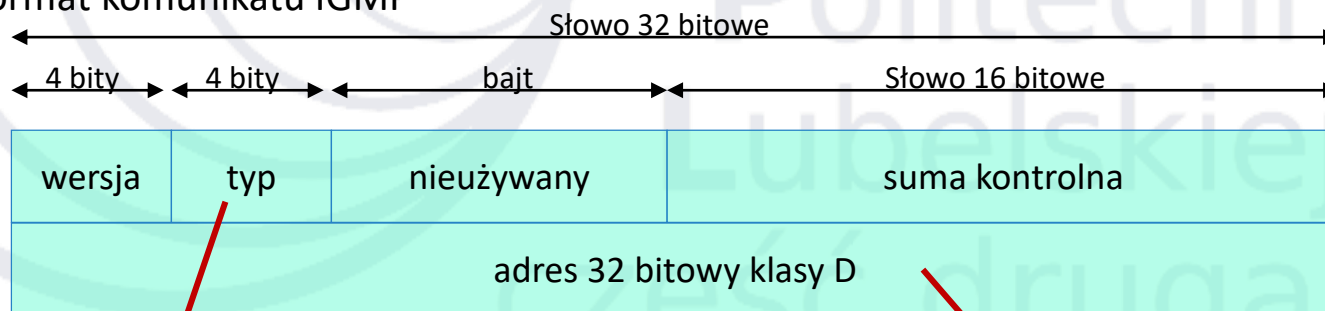
Typ	Znaczenie
0	Echo Reply
1 - 2	Zarezerwowane
3	Destination Unreachable
4	Source Quench (tłumienie nadawcy)
5	Redirect Message (zmień trasowanie)
6	Alternate Host Address
7	Zarezerwowane
8	Echo Request (żądanie echa)
9	Router Advertisement
...	...

Protokół IGMP

- Internet Group Management Protocol, IGMP** – protokół przesyła swoje komunikaty za pośrednictwem oddzielnych datagramów IP (podobnie jak ICMP)



- Format komunikatu IGMP



Wartość 1 gdy dotyczą zapytania przez router multicast
Wartość 2 gdy dotyczy odpowiedzi przesyłanej przez host

Group Address – zawiera adres grupy będący dla hosta potwierdzeniem jego członkostwa w grupie multicast

Podstawy Sieci Komputerowych

Listy sterowania dostępem, translacja adresów

Listy sterowania dostępem: standardowe i rozszerzone.
Statyczna i dynamiczna translacja adresów

dr hab. inż. Konrad Gromaszek

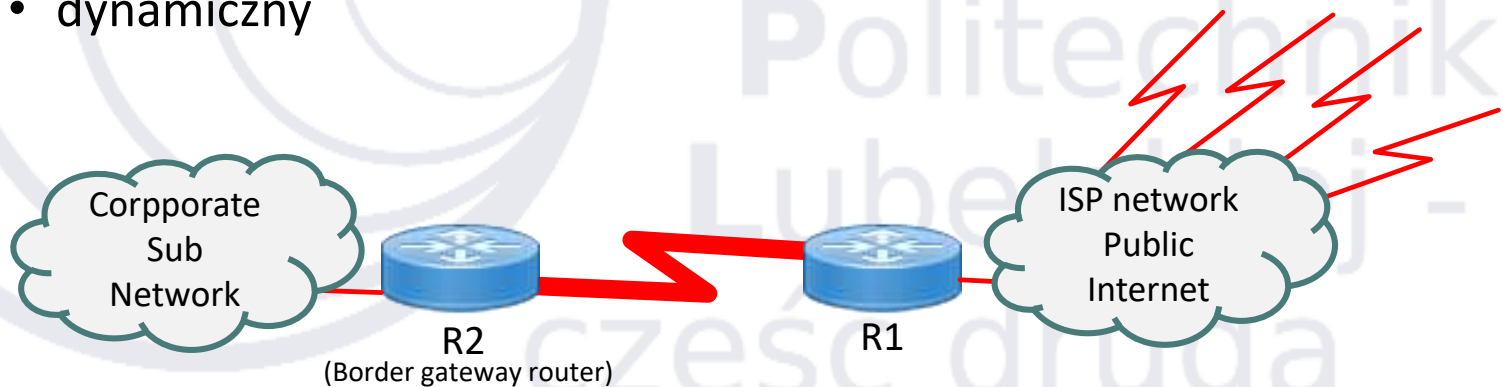
Wprowadzenie

- Mechanizm translacji adresów sieciowych
- NAT
- PAT
- Listy kontroli dostępu

Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Translacja adresów sieciowych

- **Network Address Translation, NAT** – mechanizm translacji adresów sieciowych, opisany w RFC 1631
- Wyróżnia się dwa typy translacji:
 - statyczny
 - dynamiczny



Stub Network = only one exit to outside network

Publiczne i prywatne adresy IP

- Publiczne adresy internetowe są regulowane przez 5 *Regional Internet Registers* (RIRs)

- ARIN
- RIPE
- APNIC
- LACNIC
- AfriNIC



- Pulę prywatnych adresów IPv4 zdefiniowano w RFC 1918:

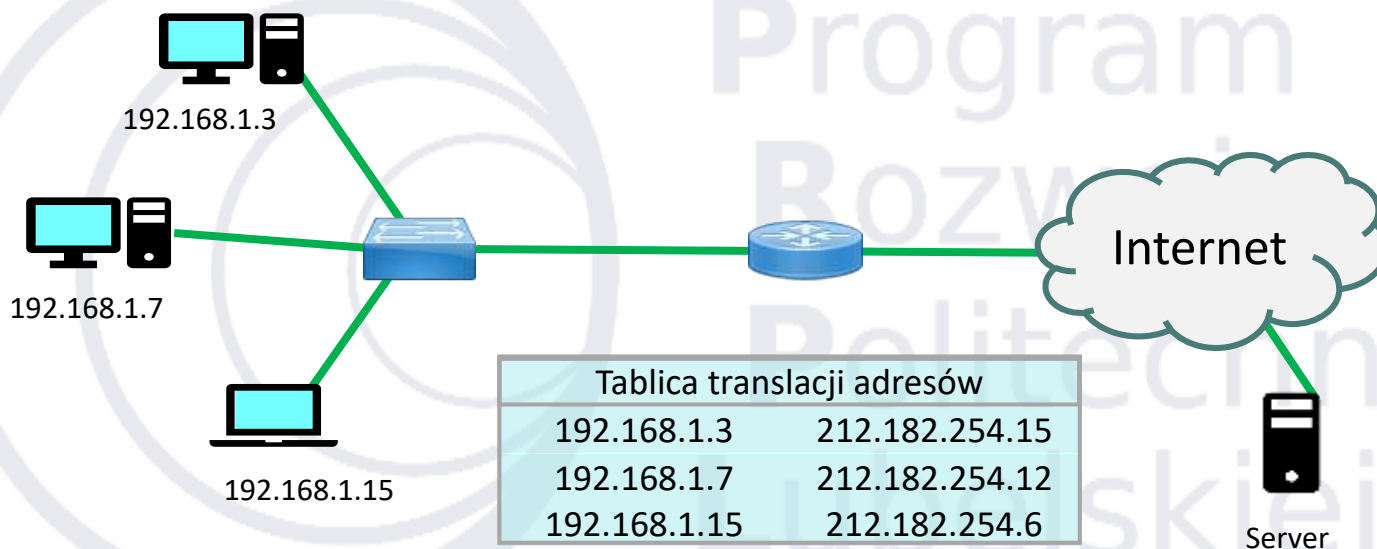
- | | | |
|-----|-------------------------------|----------------|
| • A | 10.0.0.0 – 10.255.255.255 | 10.0.0.0/8 |
| • B | 172.16.0.0 – 172.31.255.255 | 172.16.0.0/12 |
| • C | 192.168.0.0 – 192.168.255.255 | 192.168.0.0/16 |

Statyczna translacja adresów

- **Translacja statyczna** polega na odwzorowaniu adresu sieci wewnętrznej na adres unikatowy w skali światowej w stosunku 1:1, tzn. jednemu adresowi w sieci lokalnej odpowiada jeden adres unikatowy w skali globalnej
- Jej stosowanie jest uzasadnione, gdy posiadamy niewiele hostów w sieci lokalnej i odpowiednią liczbę adresów globalnych

Adres sieci lokalnej	Adres globalny
192.168.1.3	212.182.254.15
192.168.1.7	212.182.254.12
192.168.1.15	212.182.254.6

Statyczna translacja adresów



Translacja adresów

- Nagłówek IP **przed** translacją

Wersja	Dł. nagł.	Typ obsł.	Długość całkowita	
Identyfikacja			Znacznik	Przesunięcie fragm.
TTL		Typ	Suma kontrolna nagłówka	
192.168.1.15				
217.160.0.201				
Opcje IP				Offset
Początek danych				

Translacja adresów

- Nagłówek IP **po** translacji

Wersja	Dł. nagł.	Typ obsł.	Długość całkowita	
Identyfikacja			Znacznik	Przesunięcie fragm.
TTL		Typ	Suma kontrolna nagłówka	
212.182.254.6				
217.160.0.201				
Opcje IP				Offset
Początek danych				

Dynamiczna translacja IPv4

- Dynamiczna translacja adresu sieciowego polega na odwzorowaniu adresów sieci lokalnej na publiczne adresy globalne, w szczególności na pojedynczy publiczny adres globalny
- Wyróżnia się dwa typy dynamicznej translacji adresów sieciowych:
 - translacja adresów sieciowych
(ang. **Network Address Translation, NAT**)
 - translacja adresów portów
(ang. **Port Address Translation, PAT**)

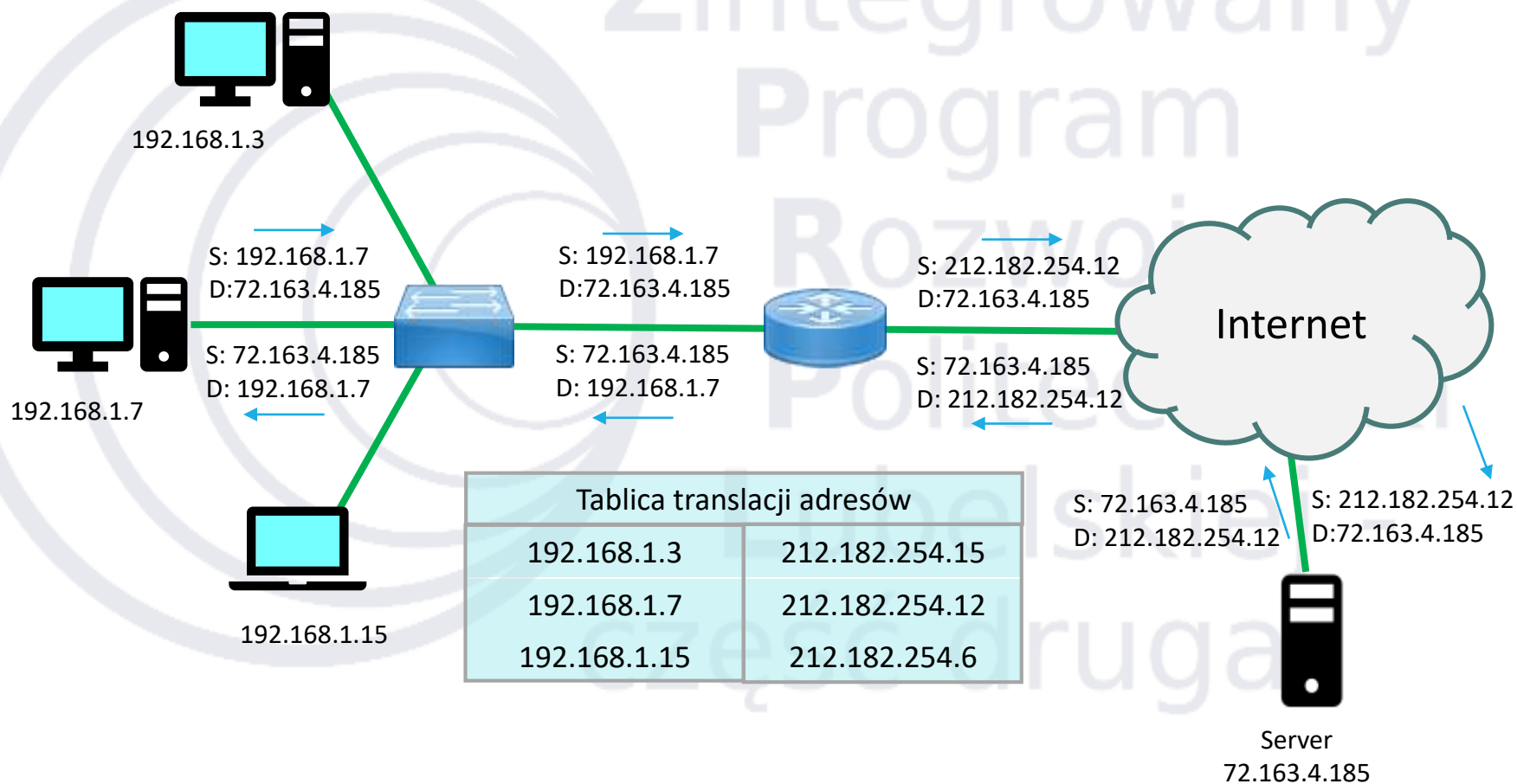
Dynamiczna translacja IPv4

- Alternatywny podział uwzględnia :
 - translacja **źródłowych** adresów sieciowych
(ang. **Source Network Address Translation, SNAT**)
 - translacja **docelowych** adresów sieciowych
(ang. **Destination Network Address Translation, DNAT**)

SNAT

- **SNAT** jest to technika polegająca na zmianie adresu źródłowego w wyniku translacji adresu na podstawie wpisów w tablicy translacji.
 - stosowana jest w sieciach lokalnych, posiadających jeden bądź kilka przydzielonych adresów globalnych
 - wysłany z hosta w sieci lokalnej pakiet w routerze na podstawie wpisu w tablicy translacji podlega działaniom związanych ze zmianą adresu źródłowego hosta na adres zdefiniowany w tablicy translacji
 - może on stanowić jeden z wielu adresów, zdefiniowanych w puli

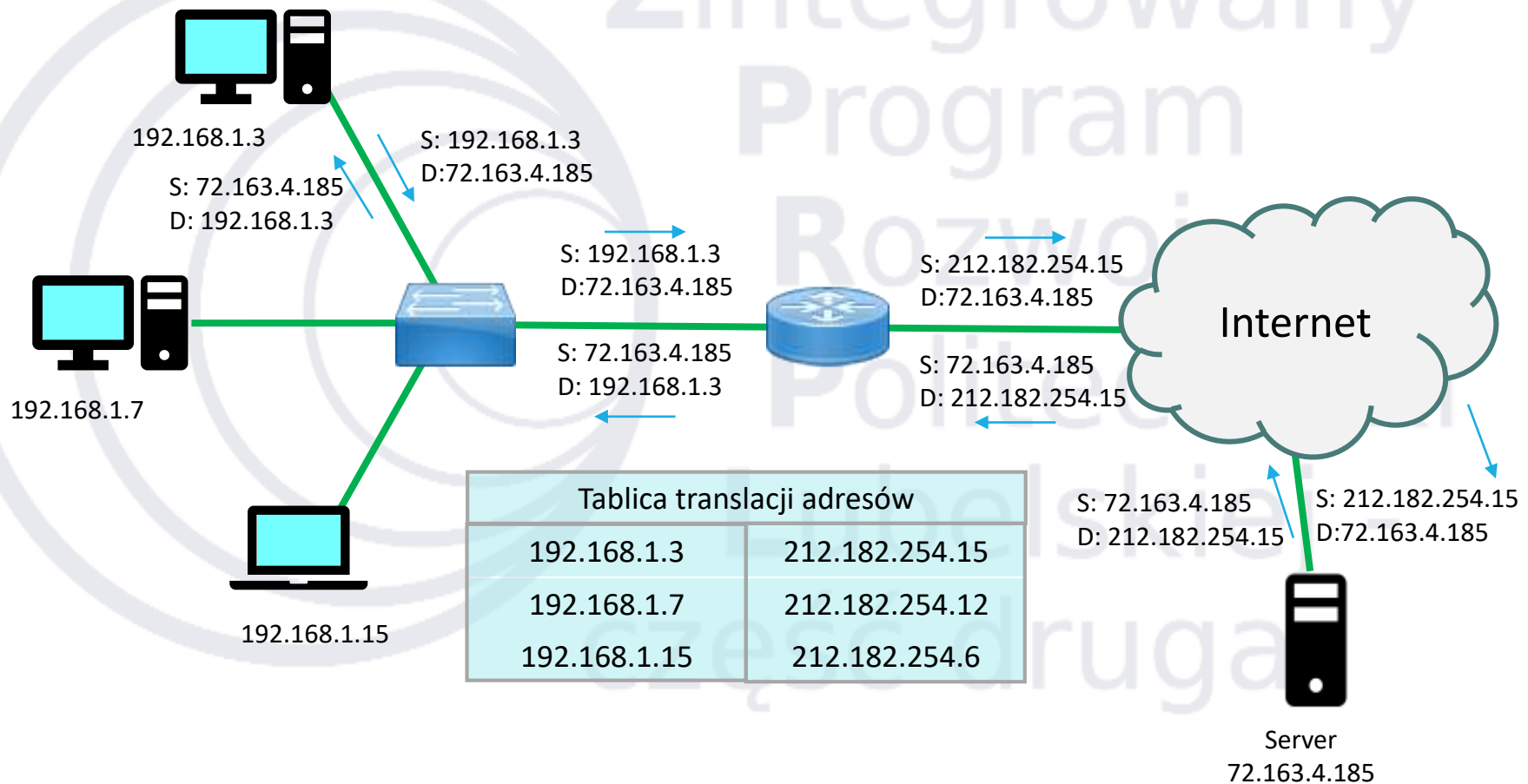
SNAT



DNAT

- **DNAT** jest to technika polegająca na zmianie adresu **docelowego** w wyniku translacji adresu na podstawie wpisów w tablicy translacji
 - stosowana jest celem zapewnienia dostępu do hosta/serwera, znajdującego się wewnątrz sieci lokalnej (np. z hosta w Internecie)
 - pakiet wysłany z hosta znajdującego się poza siecią lokalną dociera do urządzenia obsługującego NAT, gdzie na podstawie tablicy translacji (wł. zawartych w niej wpisów) następuje zmiana adresu docelowego w pakiecie na adres hosta/serwera, znajdującego się w sieci lokalnej

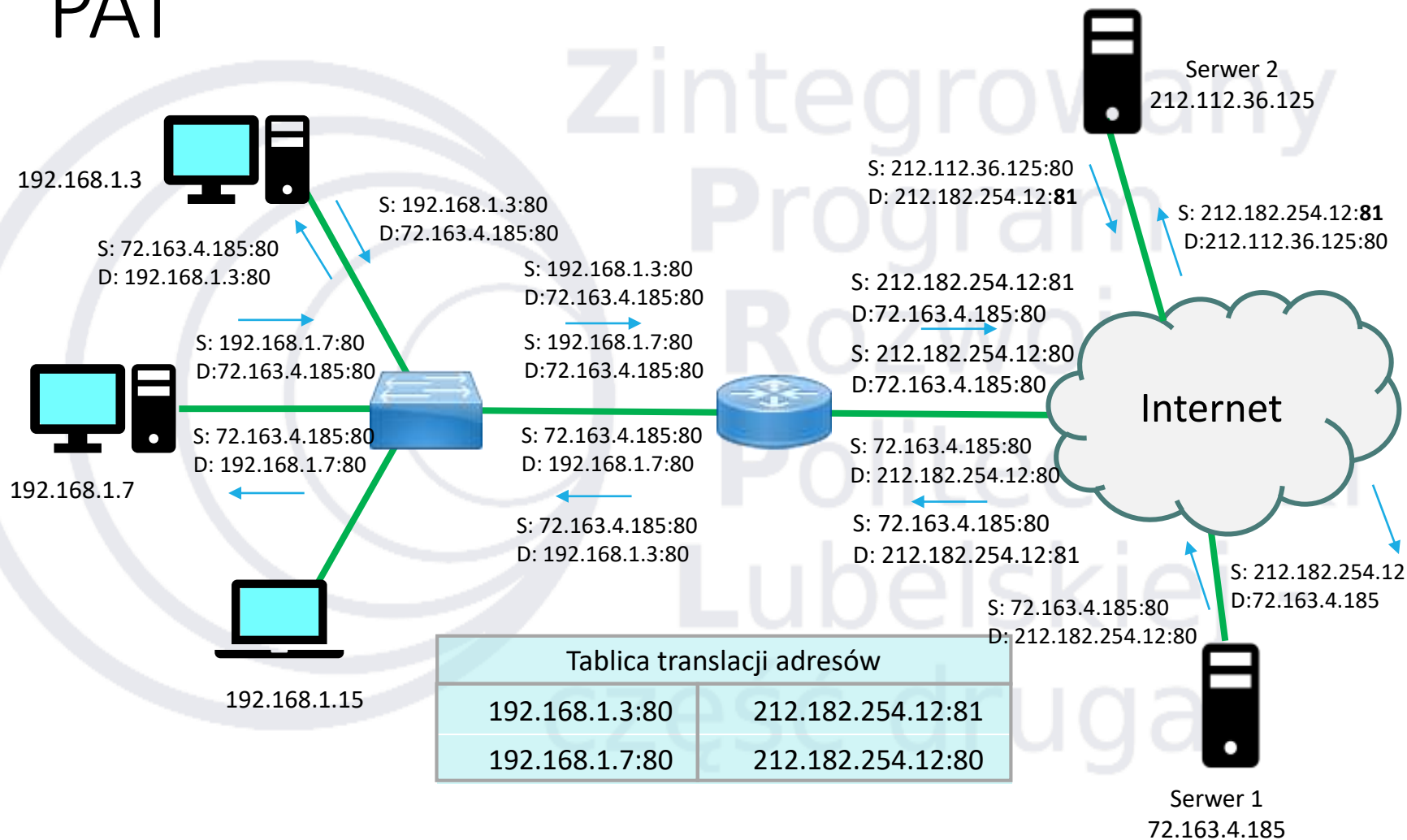
DNAT



Translacja adresów portów

- **Port Address Translation, PAT** jest to mechanizm polegający na odwzorowaniu wielu adresów prywatnych na jeden adres publiczny
 - realizuje zamianę adresu źródłowego na adres (w tym może być to adres routera/bramy)
 - następuje próba zachowania ***pierwotnego numeru portu***, na którym aplikacja próbuje uzyskać połączenie z serwerem/hostem znajdującym się w Internecie
 - w przypadku, gdy dany port jest już wykorzystywany, następuje zmiana numeru portu na kolejny z odpowiedniej grupy: 1-511;512-1023;1024-65535

PAT



Charakterystyka NAT – bezpieczeństwo

- Zastosowanie translacji adresów pozwala na zabezpieczenie przed rozpowszechnianiem informacji o topologii sieci LAN do sieci WAN
 - do WAN nie przedostają się informacje o topologii ani o sposobie adresowania w sieci lokalnej
 - z sieci lokalnej do sieci WAN przedostają się tylko informacje o adresie zewnętrznym (przydzielonym przez ISP)
 - translacja adresów oznacza brak możliwości śledzenia pakietu na całej ścieżce od nadawcy do odbiorcy

Charakterystyka NAT – adresy IPv4

- We wczesnych latach istnienia sieci Internet, kiedy tylko uniwersytety i instytucje rządowe go używały, cztery miliardy adresów IPv4 wydawały się być granicą nie do osiągnięcia
- W chwili obecnej szacuje się, że każdego dnia przybywa kilkanaście tysięcy nowych hostów podłączonych do sieci
- W obecnej sytuacji, translacja adresów pozwala na znaczne zredukowanie zapotrzebowania na adresy publiczne
- W przypadku sieci np. ze 100 hostami, aby wszystkie hosty miały dostęp do sieci Internet powinniśmy każdemu hostowi przypisać adres publiczny
- Jeśli zastosowalibyśmy translację adresów, to moglibyśmy ustawić jeden adres wyjściowy do sieci zewnętrznej dla całej sieci LAN (100 hostów)
- Pozwoliłoby to na zaoszczędzenie aż 99 adresów publicznych, co w obecnej sytuacji, gdy wyczerpują się adresy publiczne pozwala na podłączenie znacznie większej ilości hostów do Internetu

Charakterystyka NAT – dostęp

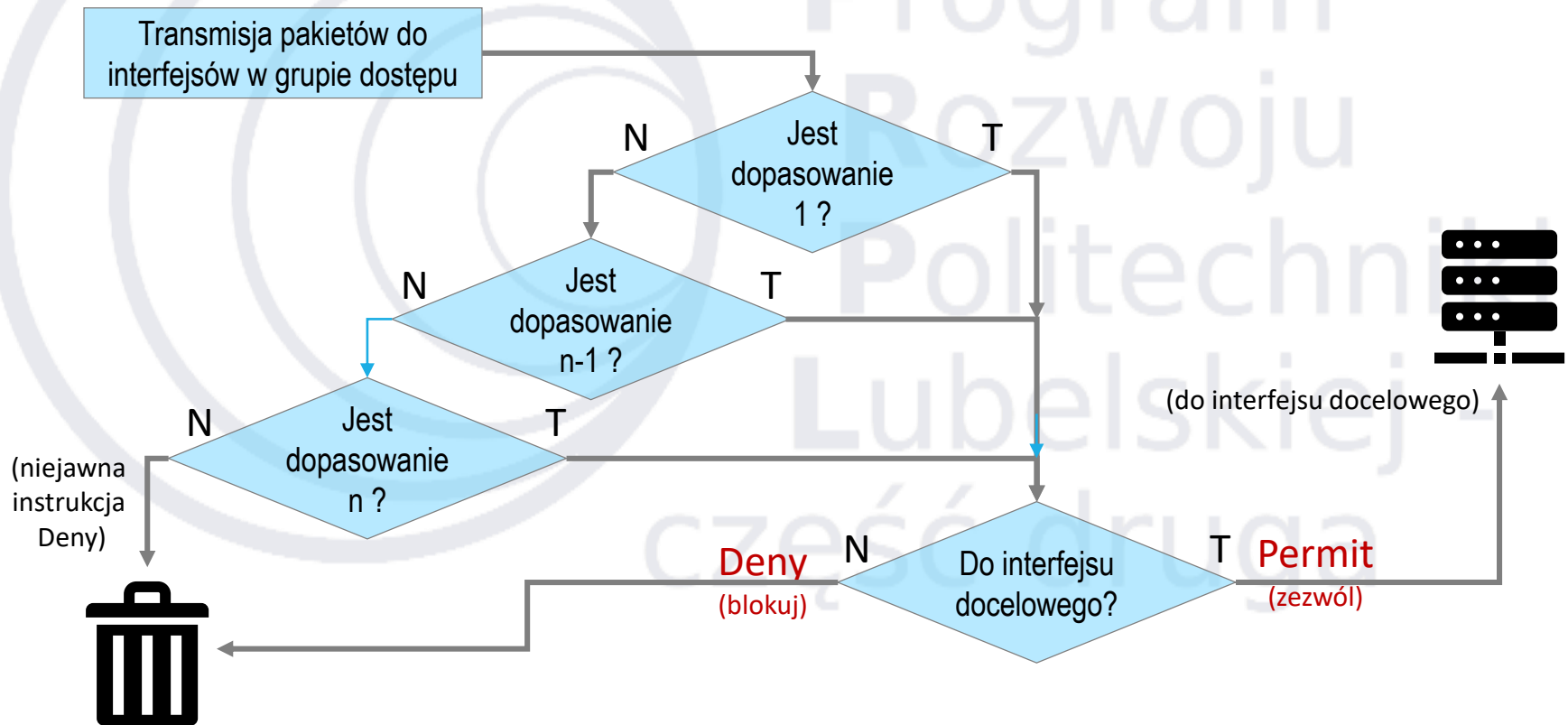
- Stosowanie techniki NAT powoduje brak możliwości umieszczenia serwera w sieci LAN, tak, by był on widoczny z „zewnątrz” (z sieci WAN)
- administrator sieci LAN, musi ustawić na routerze odpowiednią translację DNAT
- w przypadku stosowania DNAT, wpis w DNS prowadzi do adresu zewnętrznego, który następnie zostanie przez bramę przetłumaczony na adres z sieci LAN
- zastosowanie translacji adresów pozwala na obniżenie kosztów
- w przypadku niestosowania translacji adresów i zmiany usługodawcy dostarczającego Internet należy zmienić adresy na wszystkich hostach, które będą mieć dostęp do Internetu
- w przypadku zastosowania translacji adresów cała operacja obejmuje modyfikację konfiguracji routera
- umożliwia skierowanie ruchu generowany przez różne usługi na różne adresy zewnętrzne → optymalizacja obciążenia łączy (tzw. pule równoważące połączenie)
- umożliwia to monitorowanie, która z usług (protokół) (SMTP, POP3, FTP, HTTP) generuje największy ruch
- zastosowanie translacji adresów powoduje także wprowadzanie opóźnień

Listy dostępu ACL

- **Listy kontroli dostępu** (ang. **Access Control List, ACL**) to zestaw instrukcji mających na celu **zezwolenie** (ang. *permit*) na określony **ruch wchodzący** (ang. *inbound*) lub **wychodzący** (ang. *outbound*) lub też **zablokowanie takiego ruchu** (ang. *deny*)
- stanowią ważne zagadnienie z punktu widzenia bezpieczeństwa sieci
- ACL => kontrola ruchu w oparciu o adresy IP oraz numery portów
- dzięki ACL administrator może uniemożliwić dostęp nieuprawnionym klientom, a jednocześnie zapewnić go uprawnionym

Listy dostępu ACL

- Access Control List – zasada działania:



Listy dostępu ACL – tworzenie

- Etapy tworzenia ACL:
 - definiowanie listy dostępu
 - dodanie listy dostępu do interfejsu
- Warunki:
 - router przetwarza utworzone listy sekwencyjnie (od góry do dołu) → kolejność przetwarzania list ma znaczenie
 - dla każdego protokołu konieczna jest konfiguracja osobnej listy dostępu (także konfigurowanej dla ruchu wchodzącego i wychodzącego)
 - pamiętać o instrukcji na końcu listy i zakazującej wszelkiego ruchu

Definicja listy dostępu ACL

- Definicja warunku (dopasowanie) rozszerzonej listy dostępu:

```
ASA(config)# access-list nr_listy extended  
<permit | deny> protokół adres_źródła maska_źródła  
adres_celu maska_celu eq nr_portu
```

```
Router(config)# access-list nr_listy <permit | deny>  
protokół adres_źródła maska_źródła adres_celu  
maska_celu nr_portu
```

- Chcąc określić każdy (dowolny) adres i maskę sieci można zastąpić słowem kluczowym **any** (**any** = **0.0.0.0 0.0.0.0**)
- W celu zwiększenia liczby warunków (dopasowań) w kolejnym poleceniu **access-list** należy podać ten sam numer listy
- Istotna jest kolejność zadeklarowanych warunków, bo zgodnie z nią będą sprawdzane kolejne dopasowania

Dodanie listy dost. do interfejsu

- Dodanie listy dostępu do interfejsu

```
ASA(config)# access-group nr_listy <in | out>  
interface nazwa_interfejsu
```

```
Router(config-if)# ip access-group nr_listy <in | out>
```

- Do weryfikacji można użyć poleceń

```
ASA# show interface  
ASA# show access-list
```

```
Router# show ip interface  
Router# show access-list
```


Rodzaje list ACL

- **Standardowe listy ACL** (ang. *standard ACL*) – kontrola dostępu na podstawie źródłowego adresu IP; obejmują zakres numeracji 1-99 i 1300-1999
- **Rozszerzone listy ACL** (ang. *extended ACL*) – rozszerzają standardowe ACL o możliwość filtrowania pakietów z uwzględnieniem docelowego adresu IP oraz portów (źródłowych i docelowych); posiadają identyfikatory w zakresie 100-199 i 1999-2699
- **Nazwane listy ACL** – istnieje możliwość nadania nazwy ACL poleceniem `ip access-list extended Nazwa_listy` z poziomu konfiguracji globalnej; wówczas wszelkie kolejne instrukcje podawane są w trybie konfiguracji szczegółowej (`config-ext-nacl`)
- **Zwrotne listy ACL** (ang. *reflexive ACL*) umożliwiają filtrowanie pakietów na podstawie rozpoczętej sesji z wykorzystaniem parametru **established**; jeżeli stacja w sieci LAN nawiąże sesję ze stroną np. pollub.pl, czyli wyśle do niej zapytanie, lista ACL umożliwi wyłącznie przesłanie odpowiedzi ze strony pollub.pl, a pozostały ruch niezwiązany z tą sesją będzie zablokowany

Podstawy Sieci Komputerowych

Dynamiczne przydzielanie adresów

Protokoły RARP, BOOTP i DHCP

dr hab. inż. Konrad Gromaszek

Wprowadzenie

- Statyczna konfiguracja adresów sieciowych hostów jest użyteczna przy ich małej ilości
- Potrzeba określenia przy starcie jednostki wszystkich informacji potrzebnych do jej działania w sieci TCP/IP np. adresu IP wynikała z:
 - rosnącej ilości hostów w sieciach
 - automatyzacji procesu przydzielania adresów (obciążonego czynnikiem ludzkim)
 - wsparcia mobilności
- Potrzeba przydzielania adresów sieciowych realizowana jest przez protokoły:
 - **RARP** – *Reverse Address Resolution Protocol* (RFC 903)
 - **BOOTP** – *Bootstrap Protocol* (RFC 951)
 - **DHCP** – *Dynamic Host Configuration Protocol* (RFC 1531 {1993}; RFC 1541; RFC 2131 {2014}; przy czym, DHCPv6 opisano w RFC 3315 {2003}, RFC 3633 oraz RFC 3736)

RARP

- ARP rozwiązuje problem ustalenia, który adres Ethernet odpowiada danemu adresowi IPv4
- Problem odwrotny (jak poznać swój adres IP?) występuje np. podczas uruchamiania bezdyskowej stacji roboczej, która otrzymuje zwykle binarny obraz swojego OS ze zdalnego serwera plików
- Pierwszym rozwiązaniem był **protokół RARP** (RFC 903), pozwalając hostowi rozgłosić adres Ethernet, przechwytywany przez **serwer RARP**; serwer odszukuje w swoich plikach konfiguracyjnych adres Ethernet i odsyła odpowiadający mu adres IP do hosta (inicjującego)
- Wadą RARP jest to, że do połączenia z serwerem RARP używa adresu docelowego złożonego z samych jedynek => ograniczone rozgłaszanie
- Jeżeli host zapyta o adres MAC, którego serwer RARP nie ma swoim pliku odwzorowań – serwer będzie milczał
- Linux – serwer **rarp**, a konkretnie demon **rarpd**, korzysta z pliku **/etc/ethers**, gdzie znajdują się wszystkie odwzorowania adresów MAC na IP dla danego serwera

BOOTP

- W przeciwieństwie do RARP, protokół **BOOTP** (**BOOTstrap Protocol**) (opisany w RFC 951) używa komunikatów **UDP**, które są przekazywane przez routery
- Ponadto, BOOTP podaje bezdyskowej stacji dodatkowe informacje: IP serwera plików z obrazem pamięci, adres IP domyślnego routera i maskę podsieci
- Problem BOOTP to konieczność ręcznej konfiguracji tablic odwzorowujących adresy

BOOTP – procedura_(1/2)

1. Klient formułuje i wysyła zapytanie BOOTP na adres broadcast (port docelowy 67:UDP, port źródłowy 68:UDP). Operacje wykonywane przed wysłaniem:
 - Ustawienie "liczby skoków" na 0 przez klienta; Każdy pośredni router dokonuje inkrementacji; Pakiet jest odrzucany, gdy parametr przekroczy skonfigurowaną na serwerze wartość maksymalną
 - Klient ustawia "liczbę sekund" na 0. Jeśli nie otrzyma odpowiedzi, klient ponawia zapytanie ustawiając pole ponownie wpisując liczbę sekund, jaka upłynęła od czasu wysłania pierwszego pakietu BOOTREQUEST.
 - Klient ustawia "adres IP bramki" na 0. Serwer BOOTP po odczytaniu tak wypełnionego pola wpisuje w nie swój adres.
 - Jeśli klient zna swój adres IP, wypełnia pole "adres IP klienta".
2. Serwer, po odebraniu pakietu, weryfikuje dostępność konfiguracji dla klienta
 - Jeśli tak – wysyła odpowiedni pakiet BOOTREPLY do klienta z wymaganymi informacjami konfiguracyjnymi lub nazwą pliku, który klient pobiera z serwera za pomocą TFTP

BOOTP – procedura (2/2)

3. Jeśli serwer nie posiada konfiguracji dla klienta, sprawdza czy posiada dla niego informację o przekazaniu zapytania do innego serwera BOOTP. Jeśli tak nie jest, pakiet jest odrzucany. W przeciwnym wypadku serwer sprawdza czy:
 - "liczba skoków" przekroczyła skonfigurowane maksimum
 - "liczba sekund" przekroczyła dopuszczalną wartość
- Jeśli którykolwiek z warunków jest spełniony, pakiet zostaje odrzucony, w przeciwnym razie serwer przekazuje zapytanie do innego serwera BOOTP zgodnie z konfiguracją
- Kroki 2, 3 są powtarzane dotąd, aż znajdzie się serwer gotowy udzielić odpowiedzi klientowi lub zostanie spełniony jeden z warunków odrzucenia pakietu
 - UWAGA: Ważne jest, aby serwer BOOTP zawierający informację o przekazaniu zapytania BOOTREQUEST znajdował się po tej samej stronie bramki co klient.

Nagłówek BOOTP

- Struktura nagłówka protokołu BOOTP

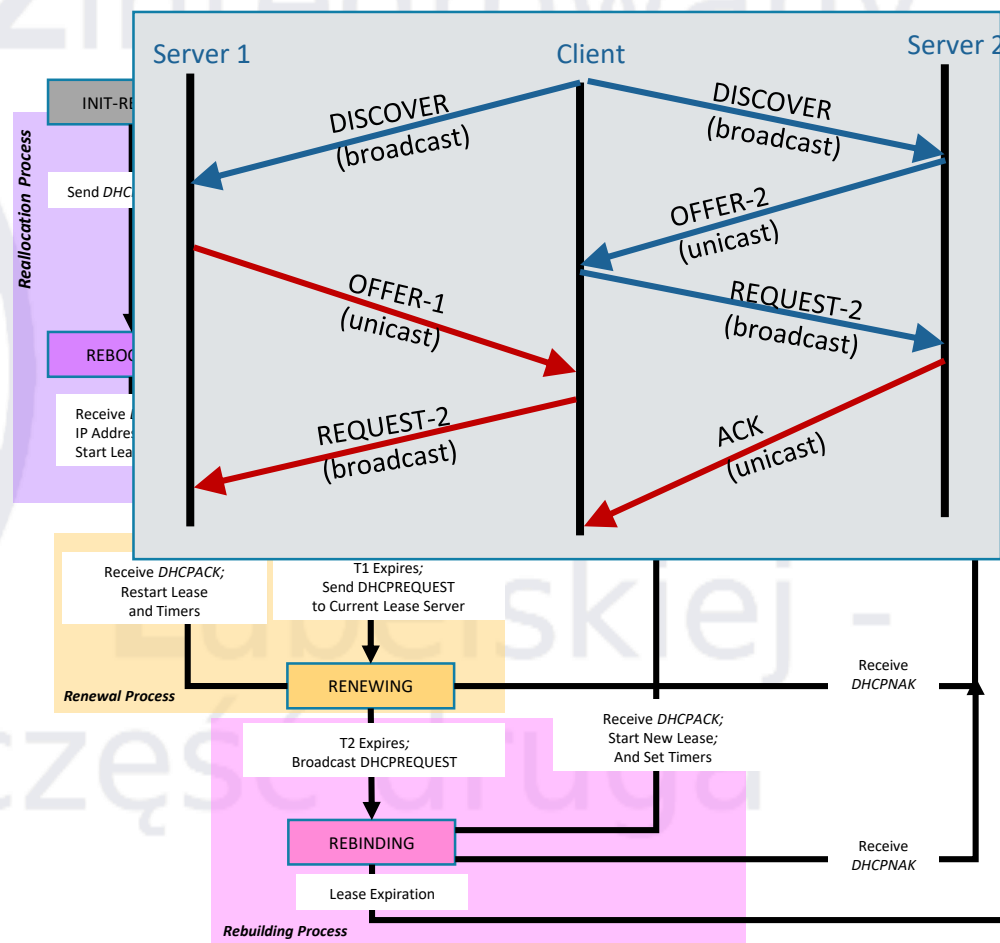
0	8	16	24	31
operacja	typ sprzętu	długość adresu sprzętowego	liczba skoków	
xid (identyfikator transakcji)				
liczba sekund		nieużywane		
adres IP klienta				
przydzielony adres IP klienta				
adres IP serwera				
adres IP bramy				
nazwa klienta (16B)				
nazwa serwera (64B)				
plik startowy (128B)				
opcje producenta (64B)				

DHCP

- DHCP – Dynamic Host Configuration Protocol:
 - RFC 2131 (1993) publikacja rekomendacji, RFC 2132 – rozszerzone parametry konfiguracyjne, RFC 3315, RFC 3633 oraz RFC 3736 – wersja dla IPv6
 - Architektura klient – serwer, oferująca następujące tryby przydzielania adresów:
 - alokacja ręczna – administrator
 - alokacja dynamiczna – serwer
 - dzierżawa – przydział dynamiczny na skończony okres czasu
 - Minimalizacja nakładów przy konfiguracji sieci IP
 - Wiele konfigurowalnych przez DHCP parametrów (oczywiście IP najważniejsze)
 - Protokół warstwy 7 (aplikacji)
 - W warstwie 4 (transportowej) korzysta z UDP.
 - IPv4: Klient wysyła komunikaty do serwera na port 67 i serwer wysyła komunikaty do klienta na port 68 (jak BOOTP)
 - IPv6: porty 546 i 547

DHCP – zasada działania

- Dodatkowe typy komunikatów:
- DHCPDISCOVER – zlokalizowanie serwerów
- DHCPOFFER – przesyłanie parametrów
- DHCPREQUEST – żądanie przydzielenia używanych parametrów
- DHCPACK – potwierdzenie przydziału parametrów
- DHCPNAK – odmowa przydziału parametrów
- DHCPDECLINE – wskazanie że adres sieciowy jest już używany
- DHCPRELEASE – zwolnienie adresu
- DHCPINFORM – żądanie przydziału parametrów (bez adresu IP)



Nagłówek DHCP

- Struktura nagłówka protokołu DHCP

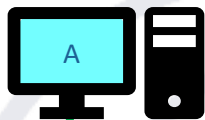
W budowie zbliżony do formatu pakietu protokołu BOOTP
Różnice:

- pole **flagi** w DHCP = 1000000000000000 (w BOOTP = 0)
- pole „**opcje**”, np.: dodatkowe dane konfiguracyjne; okres dzierżawy; maska podsieci lokalnej; adres IP serwera czasu; adres IP serwera DNS; rozmiar pliku konfiguracyjnego

0	8	16	24	31
operacja	typ sprzętu	długość adresu sprzętowego	liczba skoków	
xid (identyfikator transakcji)				
liczba sekund		flagi		
adres IP klienta				
przydzielony adres IP klienta				
adres IP serwera				
adres IP bramy (routera)				
adres sprzętowy klienta (16B)				
nazwa serwera (64B)				
plik startowy (128B)				
opcje producenta (zmiennej długości)				

DHCP – zasada działania^(1/2)

Klient
IP: ???



Serwer
IP: 192.168.1.254/24



MAC: Adres sprzętowy MAC
CIADDR: Adres IP klienta
GIADDR: Adres IP bramy
CHADDR: Adres sprzętowy klienta

SRC MAC: MAC A	SRC MAC: ?	UDP	CIADDR: ?	GIADDR: ?
DST MAC: FF:FF:FF:FF:FF:FF	IP DST: 255.255.255.255	67	Mask: ?	CHADDR: MAC A

Klient
IP: ???



Serwer
IP: 192.168.1.254/24



SRC MAC: MAC Srv	SRC MAC: 192.168.1.254	UDP	CIADDR: 192.168.1.5	GIADDR: ?
DST MAC: MAC A	IP DST: 192.168.1.5	67	Mask: 255.255.255.0	CHADDR: MAC A

DHCP – przyznawanie adresów

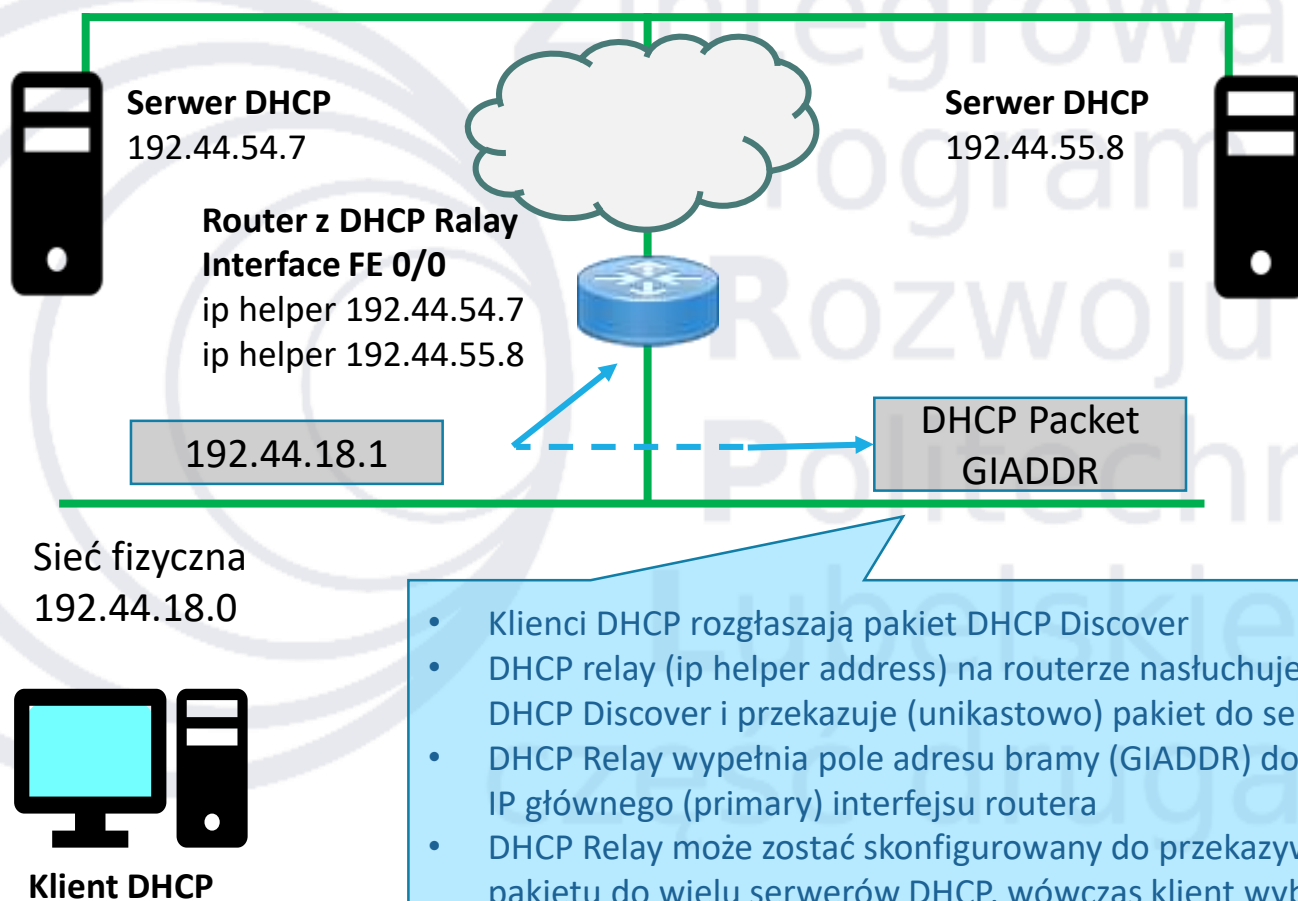
- Serwery DHCP przydzielają adresy ze zdefiniowanych pul adresów.
 - na serwerach DHCP mogą być dostępne także inne informacje, takie jak adresy serwerów DNS, adresy serwerów WINS i nazwy domen
 - serwery DHCP mogą także mieć zdefiniowane adresy MAC i automatycznie przypisywać dla tych klientów zawsze te same adresy IP
- DHCP obsługuje trzy metody przyznawania adresów:
 - ręczne wyznaczanie adresu (statyczne)
 - automatyczne przyznawanie stałego adresu dla jednostki włączającej się po raz pierwszy do sieci (dynamiczne bez ograniczeń)
 - automatyczne przyznawanie adresu dla na określony czas (dynamiczne na czas)
- Jednostki są identyfikowane przez serwer po identyfikatorze, którym przeważnie jest ich adres sprzętowy.
- Sposób obsługi jednostki zależy od konfiguracji serwera.

DHCP – przyznawanie adresów

- Dynamiczne przyznawanie adresów, a więc możliwość obsługi dowolnego węzła, daje możliwość budowania samokonfigurujących się sieci
- Rola administratora przy konfiguracji serwera DHCP:
 - wyznaczanie puli adresów, z której może korzystać serwer DHCP
 - określenie reguł, którymi posługuje się serwer przy przyznawaniu adresów
- Czas, na który serwer przyznaje adres zależy od życzenia klienta oraz konfiguracji serwera
 - dla szybko zmieniających się warunków sieci – krótki, dla innych – długi
- Przykładowe polecenia konfiguracyjne

```
Router(config)#ip dhcp pool Mypool-name
Router(dhcp-config)#network 172.16.10.0 255.255.255.0
Router(dhcp-config)#default-router 172.16.10.254
Router(dhcp-config)#dns-server 172.16.1.2
Router(config)#ip dhcp excluded-address 172.16.1.1 172.16.1.10
Router(config)#ip dhcp excluded-address 172.16.1.254
Router#show ip dhcp binding // sprawdzenie konfiguracji
```

DHCP Relay (przekazanie DHCP)



Podstawy Sieci Komputerowych

Omówienie wybranych protokołów warstwy transportowej

Protokoły TCP i UDP

dr hab. inż. Konrad Gromaszek

Plan wykładu

- Usługa transportowa – charakterystyka
- Prymitywy usług transportowych
- Połączenia w warstwie transportowej: port i gniazdo
- Protokół TCP
- Nawiązywanie, zamykanie i zarządzanie połączeniem
- Protokół UDP
- Protokoły strumieniowania DCCP, SCPT, RSVP

Warstwa transportowa

- **Cel:** dostarczanie wydajnych, niezawodnych i ekonomicznych usług na potrzeby użytkowników, działające najczęściej na poziomie warstwy aplikacji modelu odniesienia
- **Jednostka transportowa** (*transport entity*) – całokształt sprzętu i oprogramowania służącego do realizacji powyższego celu. Może być zlokalizowana w jądrze systemu operacyjnego, w wydzielonych procesach użytkowników, w bibliotekach wbudowanych w aplikacje sieciowe, karcie sieciowej itp.
 - 2 rodzaje usług transportowych, w 3 etapach: nawiązanie połączenia, transfer danych, rozłączenie
 - Kod związany z usługami transportowymi realizowany jest całkowicie w komputerach użytkowników
 - Warstwa transportowa przyczynia się do generalnej poprawy jakości transmisji i ułatwia proces programowania
 - O jej istnieniu przesądzają: duże zróżnicowanie w zakresie konstrukcji sieci oraz nieuchronna zawodność połączeń sieciowych

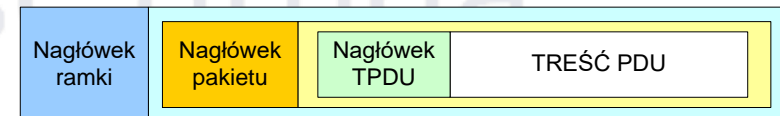
Komunikacja w w.transportowej

- **Połączeniowa (connection-oriented)**
 - etap połączenia przed właściwym przesłaniem danych
- **Bezpołączeniowa (connectionless)**
 - przesyłanie danych bez sprawdzania czy dotarły do adresata
- **Niezawodna (reliable)**
 - zapewnienie kontroli procesu przesyłania,
 - ponawianie transmisji w wypadku niedostarczenia segmentu
- **Zawodna (unreliable)**
 - brak kontroli dostarczenia pakietów
 - brak retransmisji pakietów (ew. warstwy wyższe)
- **Stanowa (stateful)**
 - sesja pomiędzy serwerem i klientem (monitorowana przez serwer)
- **Bezstanowa (stateless)**
 - brak monitorowania stanu klienta przez serwer
 - mniejsze obciążenie, brak informacji o poprzednich odpowiedziach

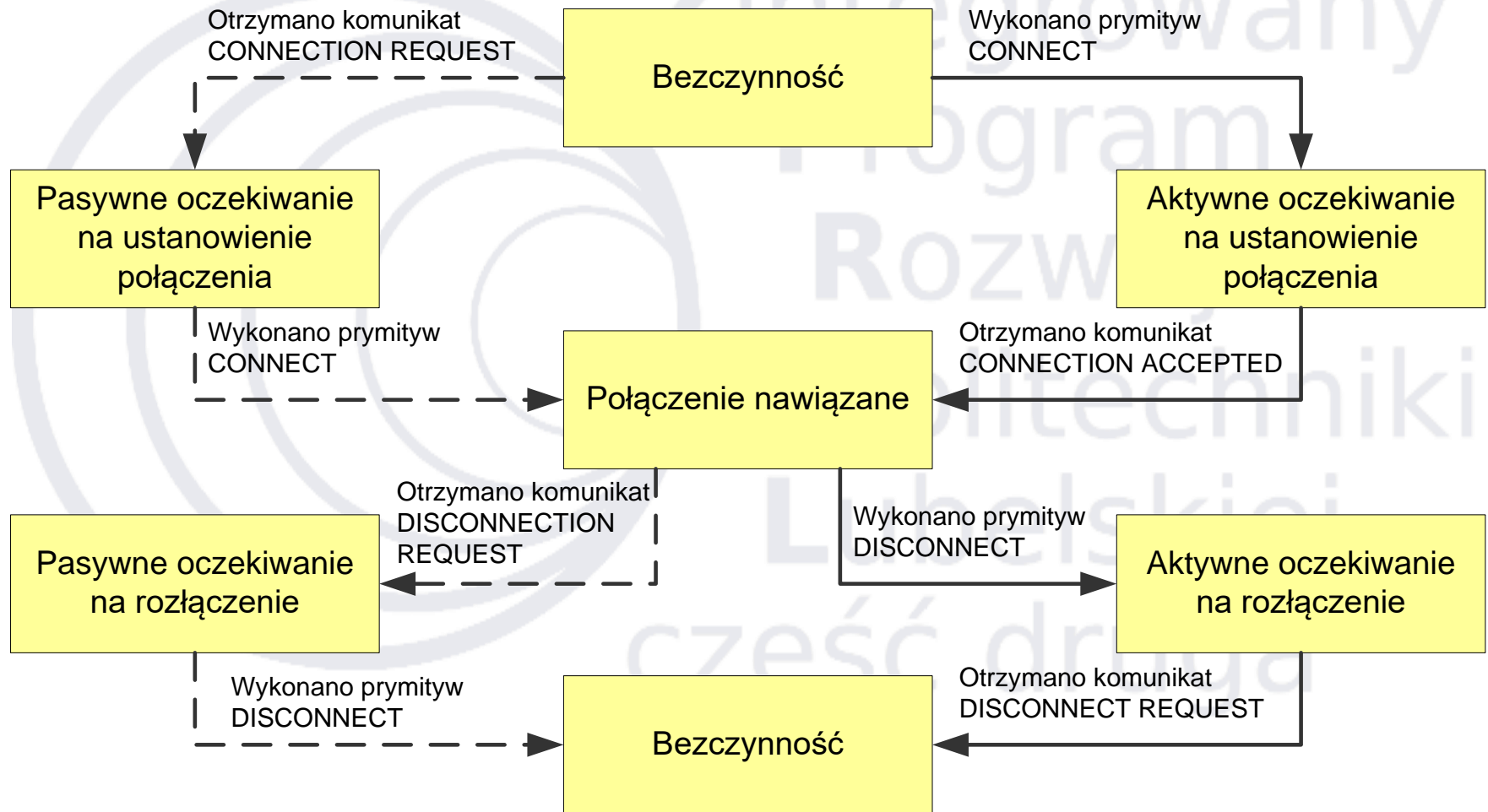
Prymitywy usług transportowych

- w. transportowa udostępnia warstwom wyższym swe usługi za pośrednictwem odpowiednich interfejsów; każda usługa transportowa ma swój własny charakterystyczny interfejs
- usługa transportowa (połączeniowa) jest z założenia niezawodna, choć zrealizowana (przez jednostkę transportową) na bazie zawodnych usług sieciowych
- pipe jako przykład (zorientowanej na połączenie) usługi sieciowej, kompensującej niedoskonałości sieci
- przeznaczenie w. transportowej: aplikacje, stąd konieczność przejrzystych i prostych w użyciu interfejsów
 - interfejs oparty na prymitywach jest skrajnie prosty ale i pogładowy
 - komunikat – wymieniany przez jednostki transportowe klienta i serwera to tzw. Jednostka danych protokołu transportowego TPDU (*Transport Protocol Data Unit*)

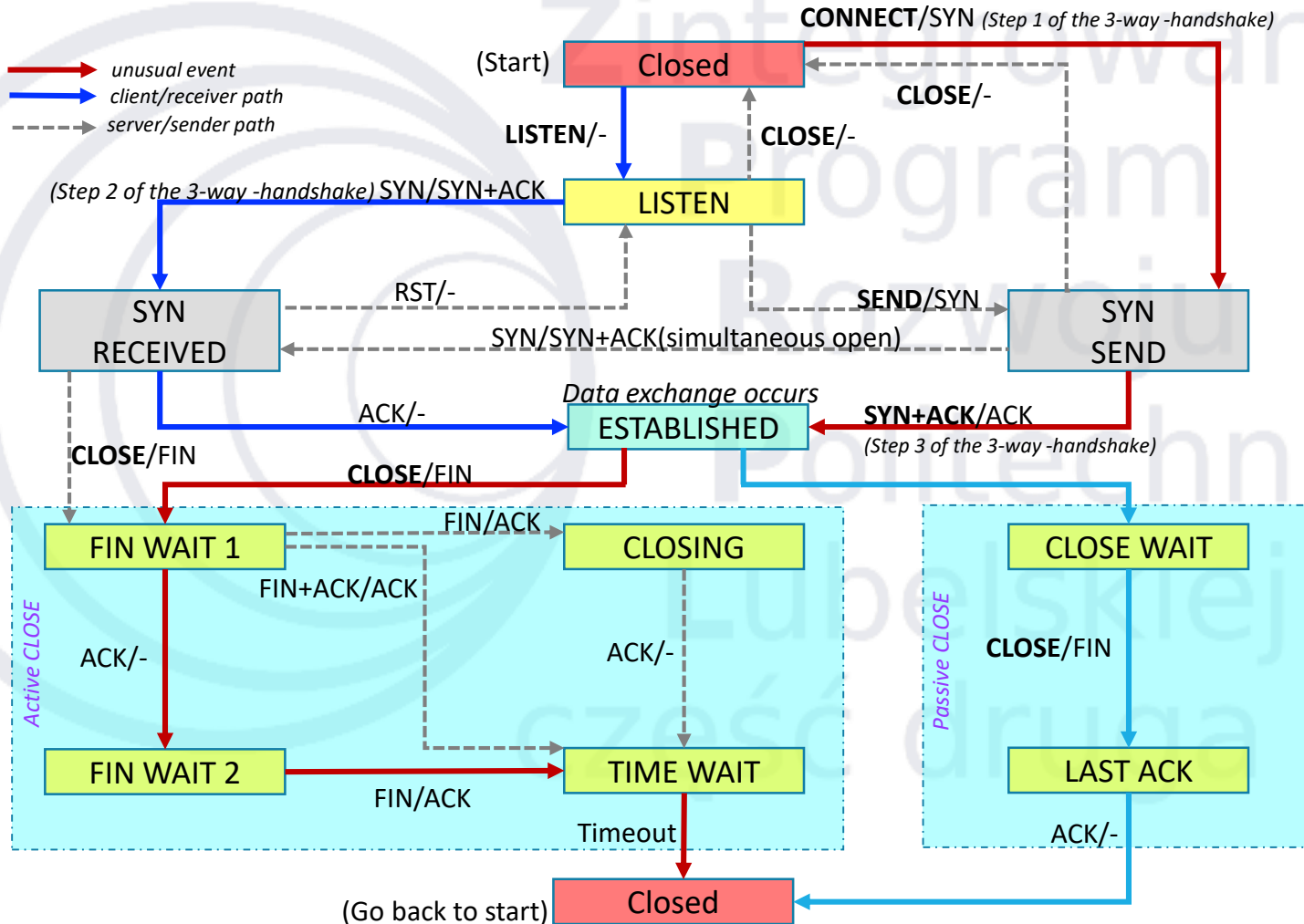
Prymityw	Przesyłany pakiet	Przeznaczenie
LISTEN	(brak)	Zablokowanie procesu do czasu nadejścia żądania połączenia
CONNECT	CONNECTION REQUEST	Aktywna próba nawiązania połączenia
SEND	DATA	Wysyłanie informacji
RECEIVE	(brak)	Zablokowanie procesu do momentu nadejścia pakietu DATA
DISCONNECT	DISCONNECTION REQUEST	Żądanie rozłączenia



Prymitywy usług transportowych



Połączeniowa u. transportowa



Adres portu i gniazdo

- Użytkownik połączenia TCP/UDP jest identyfikowany za pośrednictwem numeru, zwanego **adresem portu** (ang. **port address**)
- Adres portu jest łączony z adresem internetowym IP hostu, tworząc **gniazdo** (ang. **socket**)
- Para gniazd identyfikuje oba końce każdego połączenia TCP/UDP

Socket nadający

=

Adres IP Nadawcy

+

Adres Portu
Źródłowego

Socket odbierający

=

Adres IP Odbiorcy

+

Adres Portu
Docelowego

Well-known ports

- Konkretnie usługi sieciowe są przypisane na stałe do pewnych numerów portów (przydzielane przez IANA)
 - dla przykładu w systemie Linux plik `/etc/services` definiuje przypisanie poszczególnych usług do portów (również w `/etc/protocols`)
 - analogicznie w Windows: `%WINDIR%\system32\drivers\etc`

ftp-data	20/tcp	
ftp	21/tcp	
telnet	23/tcp	
Smtp	25/tcp	mail
time	37/tcp	timserver
time	37/udp	timserver
name	42/udp	nameserve
whois	43/tcp	nickname
domain	53/tcp	
bootps	67/udp	# bootp server
bootpc	68/udp	# bootp cl ient
tftp	69/udp	
http	80/tcp	

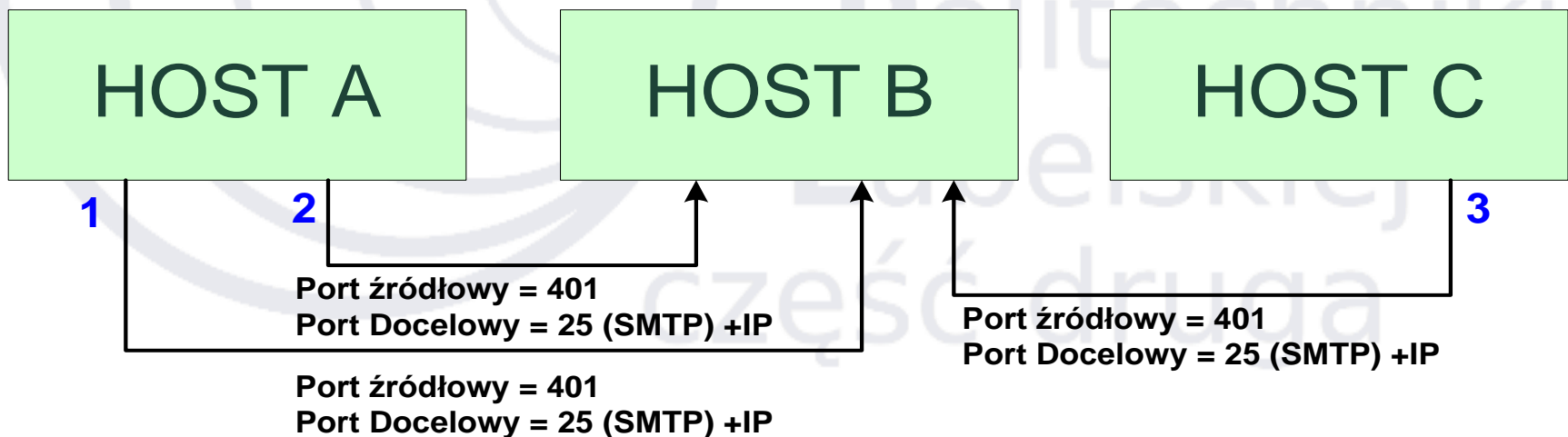
UWAGA: Możliwe jest teoretycznie przypisanie dla TCP numeru portu do jednej usługi, a dla UDP przypisanie tego samego numeru do zupełnie innej usługi, jednak w celu uniknięcia nieporozumień, nigdy się tego nie robi

Lista otwartych portów: `netstat`

Rola gniazd przy połączeniach

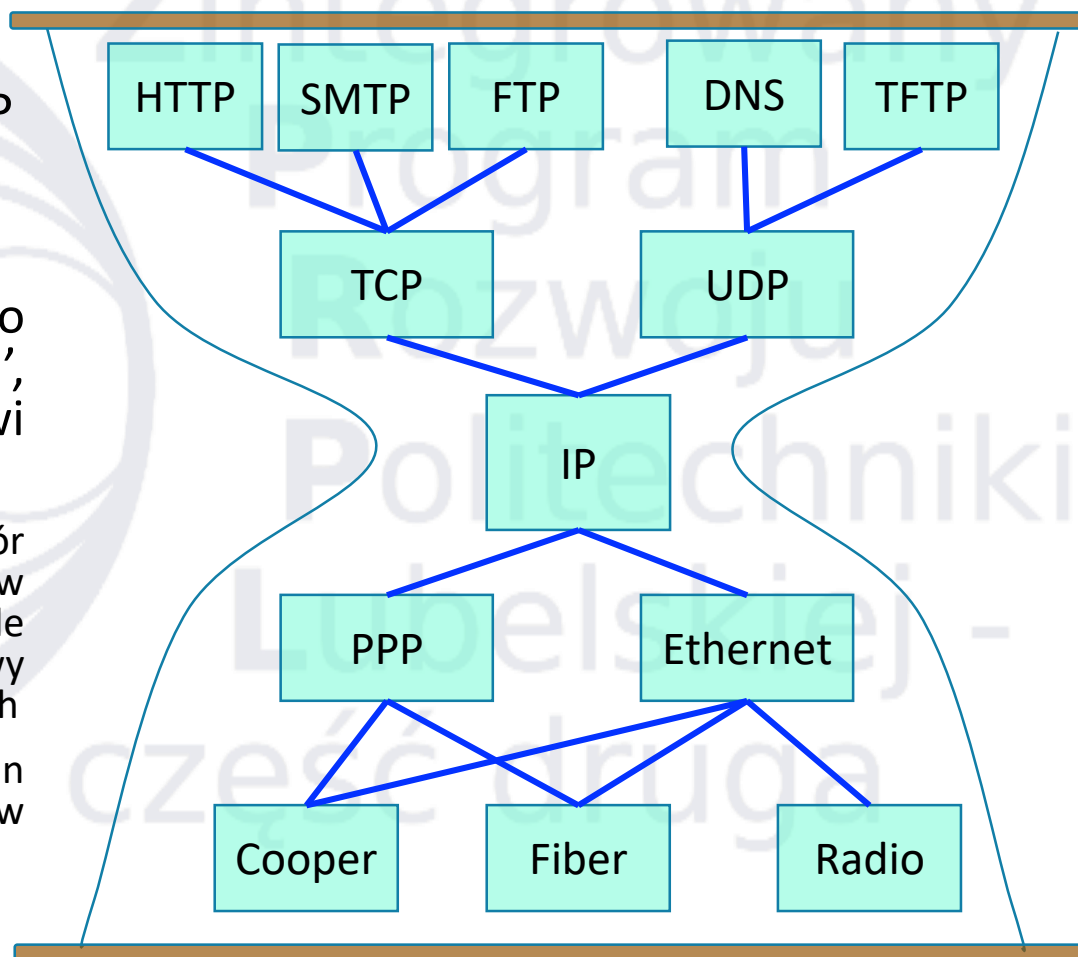
- UWAGA: identyczne adresy portów źródłowych mogą być rozróżniane poprzez sprawdzenie dołączonych do nich adresów IP => wielu użytkowników może korzystać usług i sieciowej oferowanej za pośrednictwem tego samego portu TCP (np. SMTP)

Rozróżnienie połączeń za pośrednictwem adresów IP



Protokoły stosu TCP/IP

- Protokoły stosu TCP/IP
 - TCP
 - UDP
- Określany żargonowo jako 'hourglass model', w którym IP stanowi talię stosu
 - bogaty zbiór protokołów wyższych warstw oraz wiele protokołów warstwy fizycznej i łączy danych
 - wyłącznie jeden protokół występuje w warstwie sieciowej



Protokół TCP

- Podstawowe usługi TCP (*Transmission Control Protocol*):
 - kontrola poprawności transmisji
 - kontrola przepływu danych
 - sekwencjonowanie
- Główne usługi TCP:
 - *Connection-oriented data management* – zarządzanie danymi przesyłanymi za pośrednictwem zestawionych uprzednio połączeń
 - *Reliable data transfer* – gwarantowanie poprawności przesyłanych danych
 - *Stream-oriented data transfer* – przesyłanie danych w formie strumieni
 - *Resequencing* – mechanizm ponownego łączenia segmentów w strumień
 - *Flow control - sliding window* – kontrola przepływu danych z wykorzystaniem mechanizmu przesuwającego okna
 - *Multiplexing* – jednoczesna obsługa wielu sesji transmisyjnych
 - *Full duplex transmission* – pełna transmisja dwukanałowa
 - *Graceful close* – łagodne zamykanie połączeń logicznych
 - *Precedence and security* – obsługa poziomów bezpieczeństwa oraz priorytetów danych

Segment TCP (1/3)

port źródłowy (source port)		port docelowy (destination port)	
numer sekwencyjny (sequence number)			
numer potwierdzenia (acknowledgement number)			
Przes. (data offset)	rreserved	(flags)	okno przesuwne (window)
Suma kontrolna (checksum)			wsk. priorytetu (urgent pointer)
opcje (options (+ padding))			
dane (data (variable))			

Numer portu źródłowego (*source port*), przeznaczenia (*destination port*) – identyfikują aplikacje wysyłającą odbierającą dane, te dwie wielkości wraz adresami IP źródła i przeznaczenia umieszczonymi w nagłówku IP, jednoznacznie identyfikują każde połączenie

Numer sekwencyjny (*sequence number*) – identyfikuje bajt w strumieniu danych, przesyłanych między nadawcą a odbiorcą. Po osiągnięciu $2^{32}-1$ rozpoczyna się znowu od zera

Numer potwierdzenia – 2-bitowy numer będący potwierdzeniem otrzymania pakietu przez odbiorcę, co pozwala na synchronizację nadawanie-potwierdzenie. Pole to jest ważne przy ustawieniu bitu ACK

Długość nagłówka (*data offset*) – pole podaje długość nagłówka w postaci słów 32 bitowych, typowy rozmiar bez opcji wynosi 20 bajtów. Niezbędne przy określaniu miejsca rozpoczęcia danych

Segment TCP (2/3)

port źródłowy (source port)		port docelowy (destination port)	
numer sekwencyjny (sequence number)			
numer potwierdzenia (acknowledgement number)			
Przes. (data offset)	rreserved	(flags)	okno przesuwne (window)
Suma kontrolna (checksum)		wsk. priorytetu (urgent pointer)	
opcje (options (+ padding))			
dane (data (variable))			

Numer Bity znaczników (*flags*):

- NS – (ang. *Nonce Sum*) jednobitowa suma wartości flag ECN (*ECN Echo, Congestion Window Reduced, Nonce Sum*) weryfikująca ich integralność
- CWR – potwierdza odebranie powiadomienia przez nadawcę, umożliwia odbiorcy zaprzestanie wysyłania echa
- ECE – (ang. *ECN-Echo*) ustawiana przez odbiorcę po otrzymaniu pakietu z ustawioną flagą CE
- URG – informuje o istotności pola "Priorytet"
- ACK – informuje o istotności pola "Numer potwierdzenia"
- PSH – wymusza przesłanie pakietu
- RST – resetuje połączenie (wymaga ponowienia sekwencji)
- SYN – synchronizuje kolejne numery sekwencyjne
- FIN – oznacza zakończenie przekazu danych

Rozmiar okna (*window*) – liczba bajtów, liczba bajtów poczynając od tego, który określony został przez pole numeru potwierdzenia, które odbiorca będzie w stanie zaakceptować

Segment TCP (3/3)

port źródłowy (source port)		port docelowy (destination port)	
numer sekwencyjny (sequence number)			
numer potwierdzenia (acknowledgement number)			
Przes. (data offset)	rreserved	(flags)	okno przesuwne (window)
Suma kontrolna (checksum)		wsk. priorytetu (urgent pointer)	
opcje (options (+ padding))			
dane (data (variable))			

Wskaźnik priorytetu (*urgent pointer*) – brane pod uwagę przy ustawieniu bitu URG, jest on dodatnim przesunięciem, które musi być dodane do pola numeru sekwencyjnego segmentu, aby uzyskać numer sekwencyjny ostatniego bajtu ważnych danych

Opcje (*options*) – może określać np. maksymalną długość segmentu MSS (ustalana przy nawiązaniu połączenia, maksymalny rozmiar segmentu jaki nadawca chce otrzymać), często określa również współczynnik rozmiaru okna (zwykle w bajtach, przy ustawieniu skala okna jest ustawiona na F wówczas rozmiar okna jest mnożony przez 2^F , przy czym maksymalnie $F=14$) oraz znaczniki czasu wykorzystywane przy pomiarze czasu dostarczania pakietu

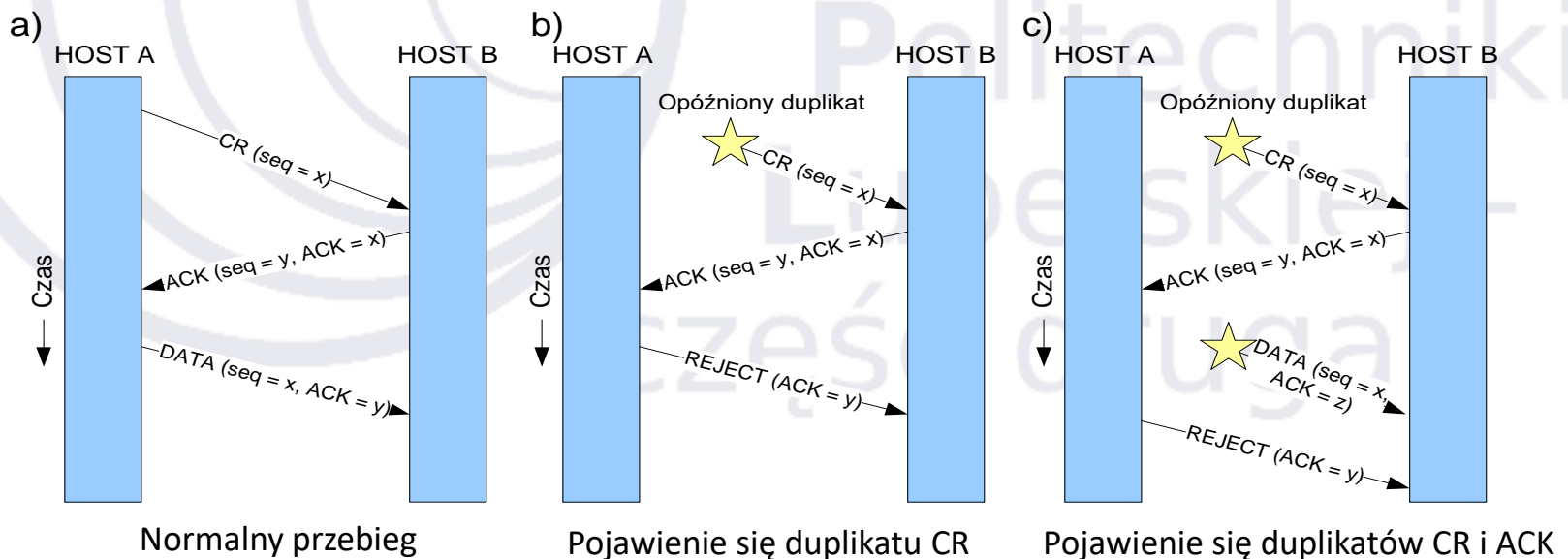
Suma kontrolna (*checksum*) – 16-bitowa liczba liczona dla danych jak i nagłówek, weryfikowana po stronie odbiorczej

Ustanawianie połączenia

- Przestaje być oczywisty przy uwzględnieniu komplikacji związanej ze specyfiką sieci (utrata, opóźnienia ...)
- Niepożądane kopie pakietów krążą po sieci, stając się niegroźne o ile przestana być honorowane przez host docelowy
- Rozwiązania oparte na tej idei:
 - przypisanie każdemu połączeniu innych adresów transportowych (wyklucza zastosowanie serwera procesów)
 - opatrzenie każdego połączenia unikalnym identyfikatorem (konieczność utrzymywania tablicy zdezaktualizowanych pakietów)
 - limitowanie czasu pakietu który może być realizowany na jeden lub kilka sposobów:
 - rygorystyczny projekt sieci
 - wbudowanie licznika przeskoków w każdy pakiet
 - znakowanie czasowe każdego pakietu

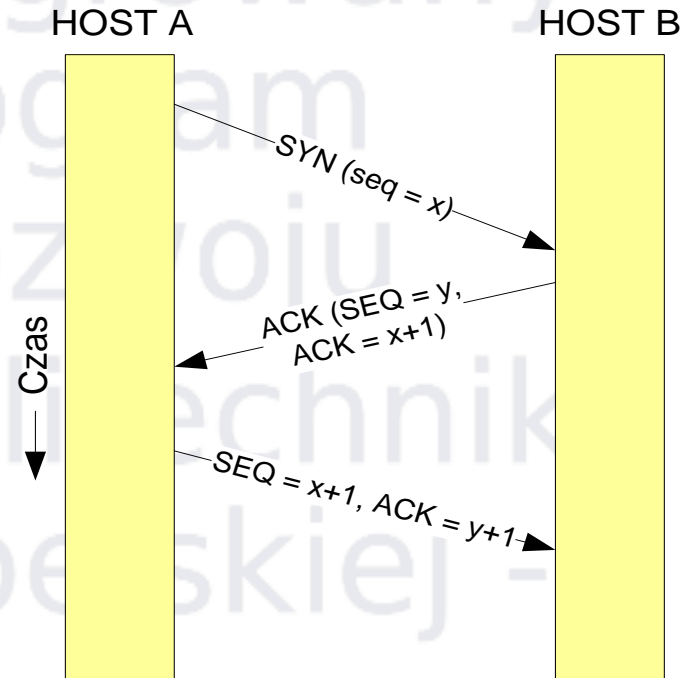
Ustanawianie połączenia

- Metoda negocjowania trójstopniowego (three –way handshake) – nie wymaga aby obydwie strony połączenia używały tego samego początkowego numeru sekwencyjnego
- Przygotowany na dowolną kombinację opóźnionych pakietów krążących po sieci



Nawiązywanie połączenia TCP

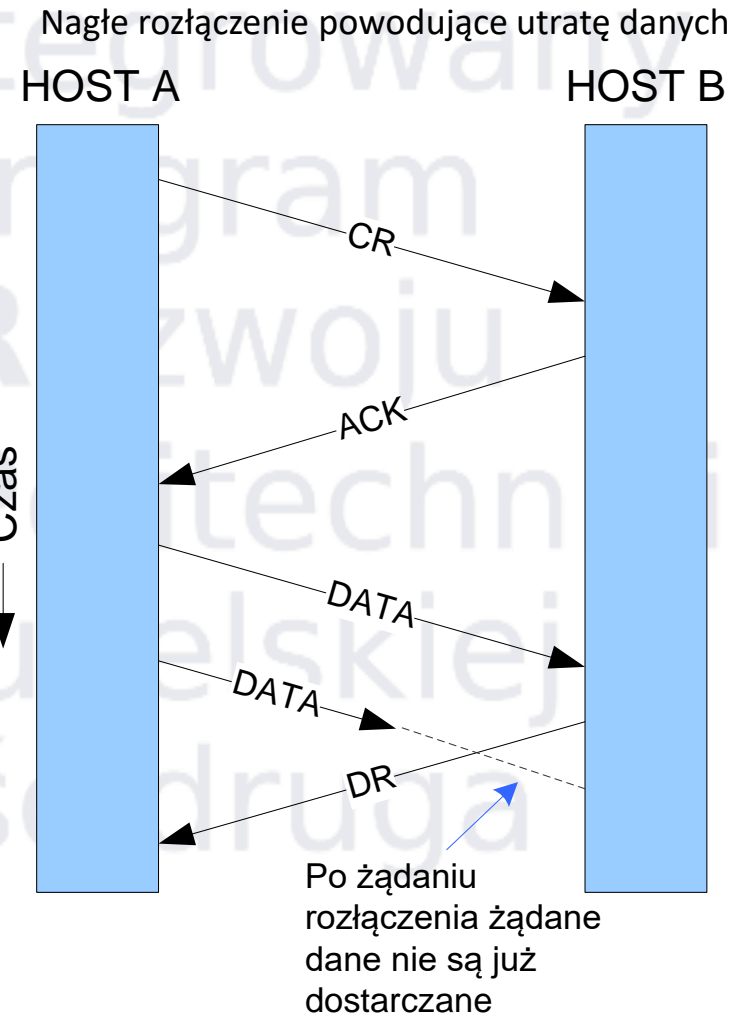
- Proces nawiązania połączenia składa się z 3 kroków (metoda „*three way handshake*”):
- Strona, która wysyła zapytania (zwykle zwana klientem) nadaje segment SYN, określający numer portu serwera, z którym klient chce się połączyć, a także początkowy numer sekwencyjny klienta
- Serwer odpowiada, wysyłając własny segment SYN zawierający początkowy numer sekwencyjny serwera i potwierdza odebranie segmentu SYN klienta, wysyłając (ACK) z nadesłanym przez klienta ISN plus jeden
- Klient potwierdza nadesłany przez serwer segment SYN – wysyłając ACK z INS serwera powiększony o jeden



Strona, która wysyła pierwszy SYN wykonuje => **aktywne otwarcie**; druga strona, która odbiera SYN i wysyła w odpowiedzi segment SYN, wykonuje tak zwane **pasywne otwarcie**

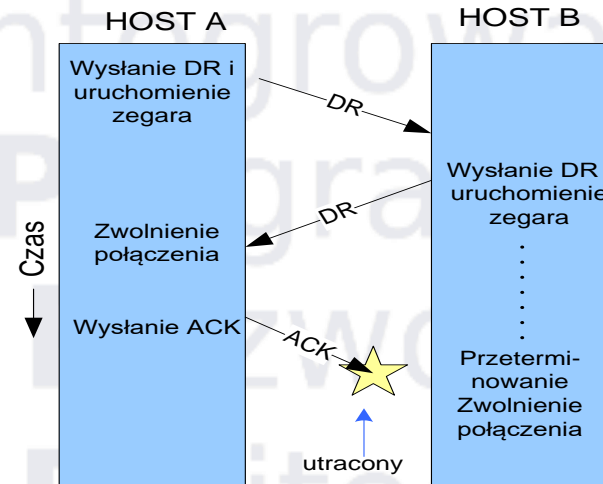
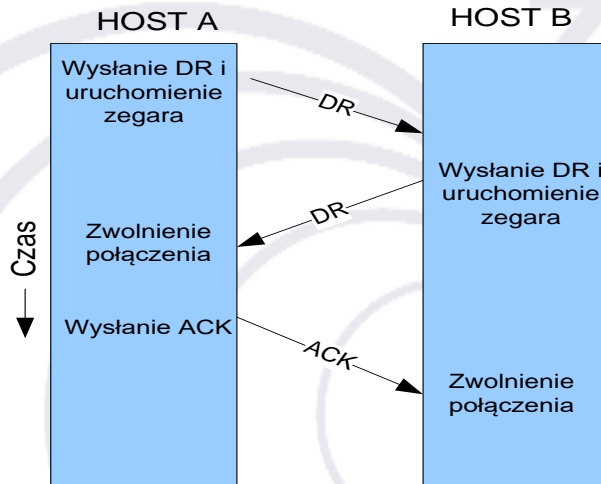
Zwalnianie połączenia

- Istnieją dwa warianty rozłączania: **symetryczny** i **asymetryczny**
- Asymetryczny** – rozłączenie następuje, gdy którakolwiek ze stron inicjuje zakończenie transmisji
- Symetryczny** – połączenie traktowane jest jako superpozycja dwóch połączeń jednokierunkowych, z których każde rozłączane jest oddzielnie
- Analogia: problem dwóch armii



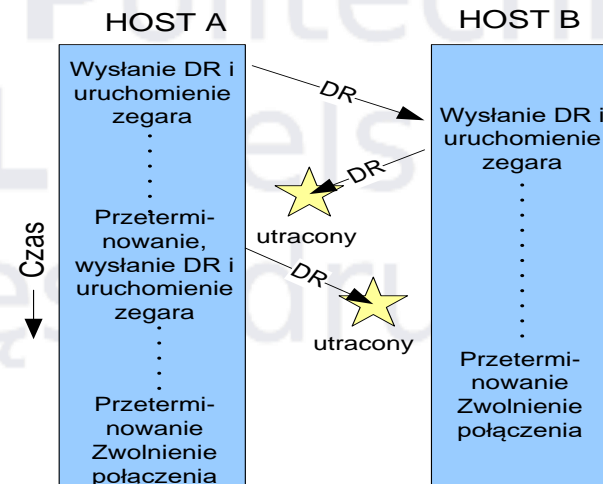
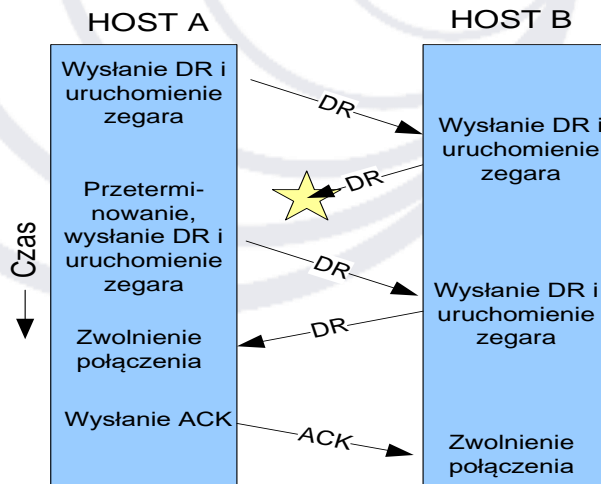
Zwalnianie połączenia – warianty

Normalny przebieg



utracone
końcowe
potwierdzenie

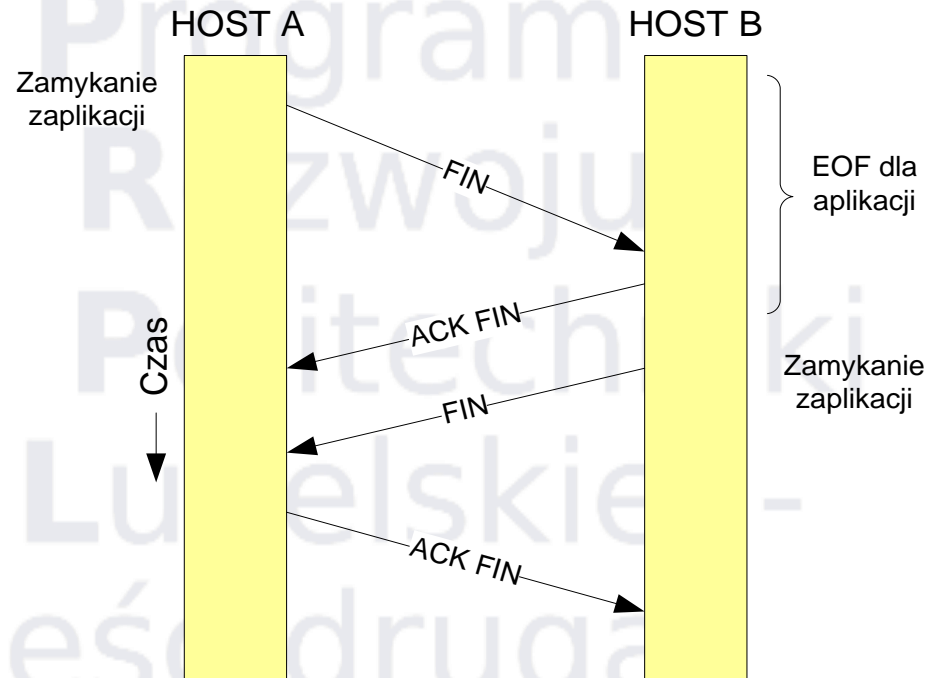
Utracona
odpowieź
na żądanie
rozłączenia



Utracona
odpowieź
na żądanie
rozłączenia,
jak i kolejne

Zamykanie połączenia TCP

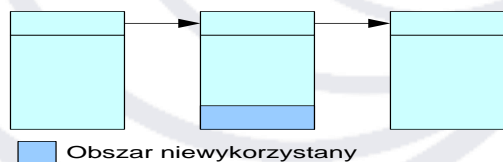
- Ze względu na to, że połączenie TCP jest połączeniem pełnodupleksowym (*full-duplex*), to każdy z kierunków musi zostać zamknięty niezależnie
- Odebranie FIN oznacza jedynie, że w tym kierunku połączenia nie będą płynęły już dane, ale TCP może nadal wysyłać dane po odebraniu FIN (połączenie półzamknięte)
- W celu pełnego zamknięcia połączenia druga strona musi wykonać podobną sekwencję operacji (FIN, oraz potwierdzenie ACK FIN)



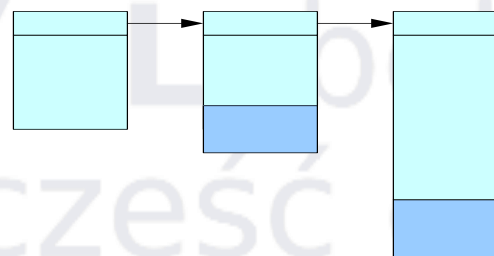
Sterowanie przepływem i buforowanie

- W środowisku zawodnych usług sieciowych host wysyłający datagramy zmuszony jest do buforowania każdej z nich, aż do otrzymania potwierdzenia
- Rozsądnym pomysłem w zakresie zarządzania buforowaniem jest stosowanie okna przesuwającego o rozmiarze zmieniającym się dynamicznie
- Ograniczenia: rozmiar dostępnych buforów, przepustowość sieci

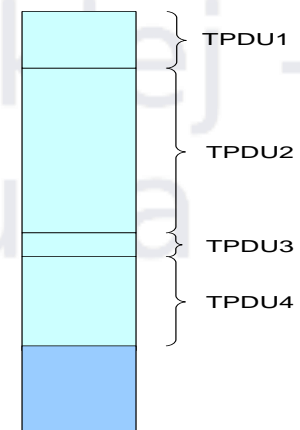
Łańcuch buforów jednakowego rozmiaru



Łańcuch buforów o zróżnicowanych rozmiarach

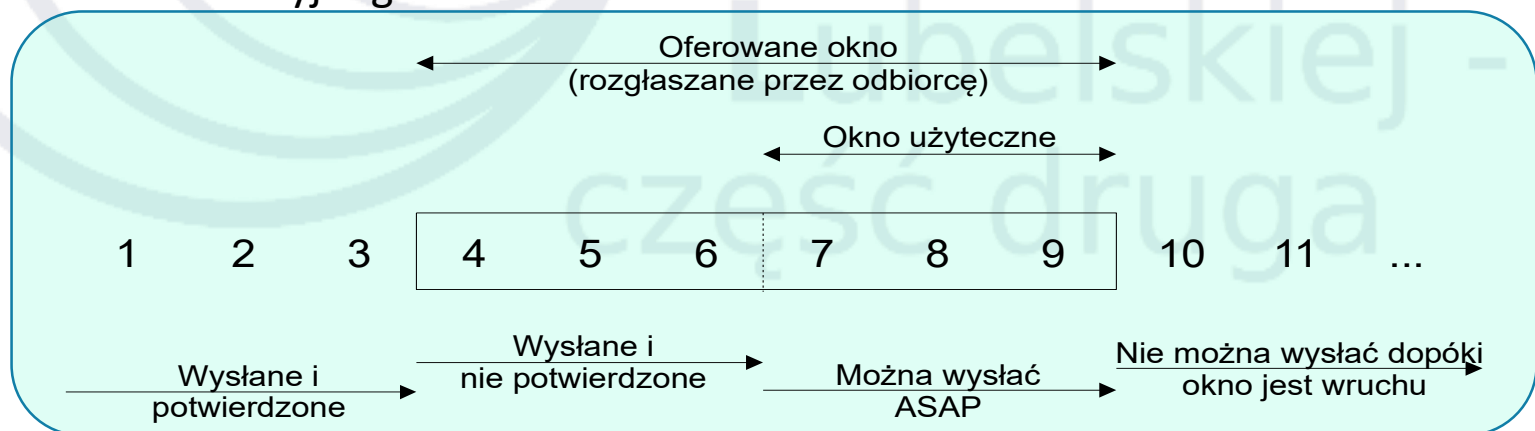


Bufor cykliczny



Okno przesuwne

- (Belsnes 1975) proponuje schemat **przesuwnego okna** (***sliding window***) dostosowywanego dynamicznie do możliwości transmisyjnych sieci:
 - Okresowe monitorowanie sieci przez host
 - Kompromis między przybywaniem potwierdzeń a częstotliwością dostosowywania rozmiaru okna
 - **Okno oferowane** (***offered window***) i **okno użyteczne** (***usable window***)
 - Rozmiar okna jest uzależniony od potwierdzonego numeru sekwencyjnego



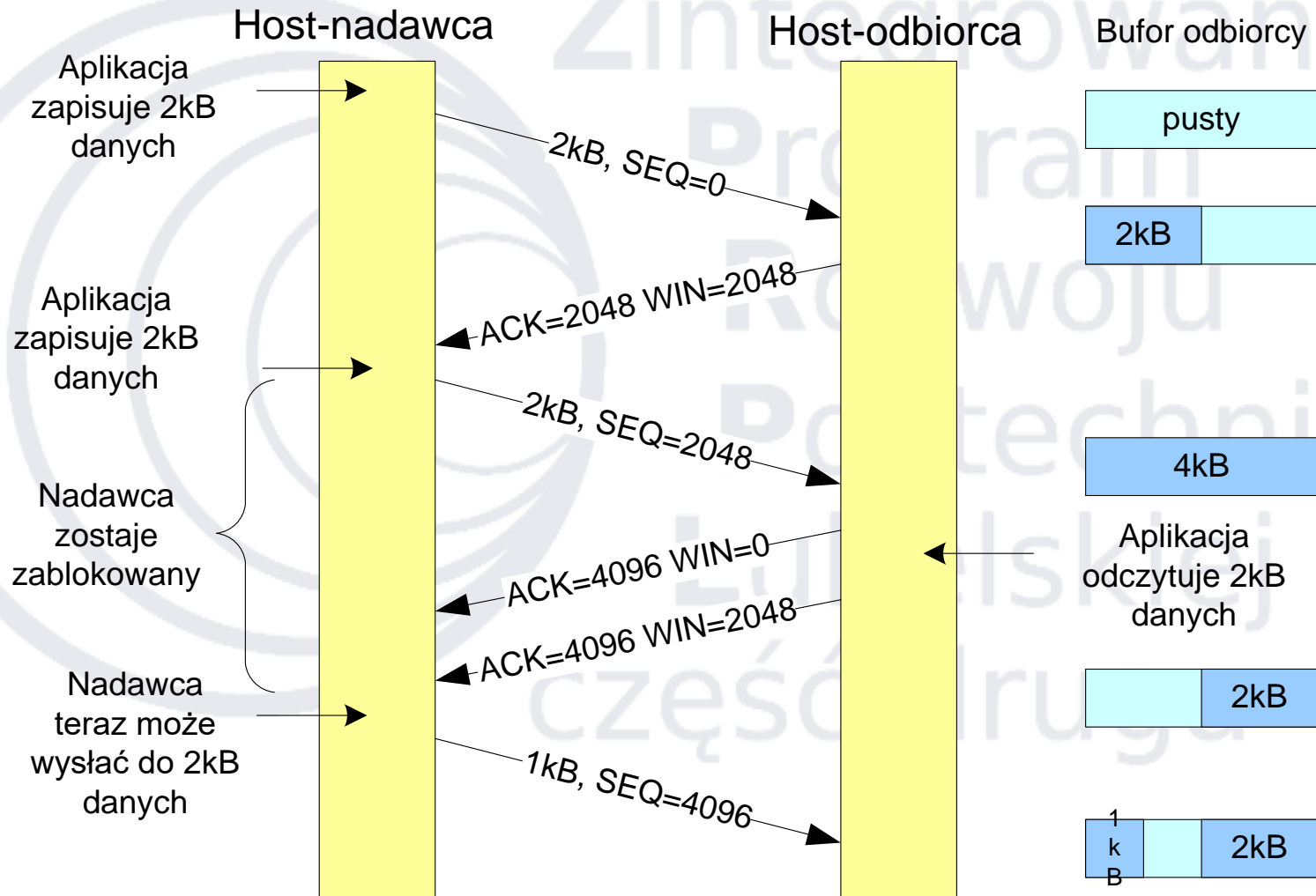
Szerokość okna vs. wydajność

- Mechanizm ślizgającego okna pozwala na określenie maksymalnej przepustowości w połączeniu TCP, która zależy od *szerokości okna* (W), *czasu propagacji* (D) i *prędkości transmisji* (R).
- Współczynnik znormalizowanej przepustowości (S).

$$S = \begin{cases} 1 & W > RD / 4 \\ \frac{4W}{RD} & W < RD / 4 \end{cases}$$

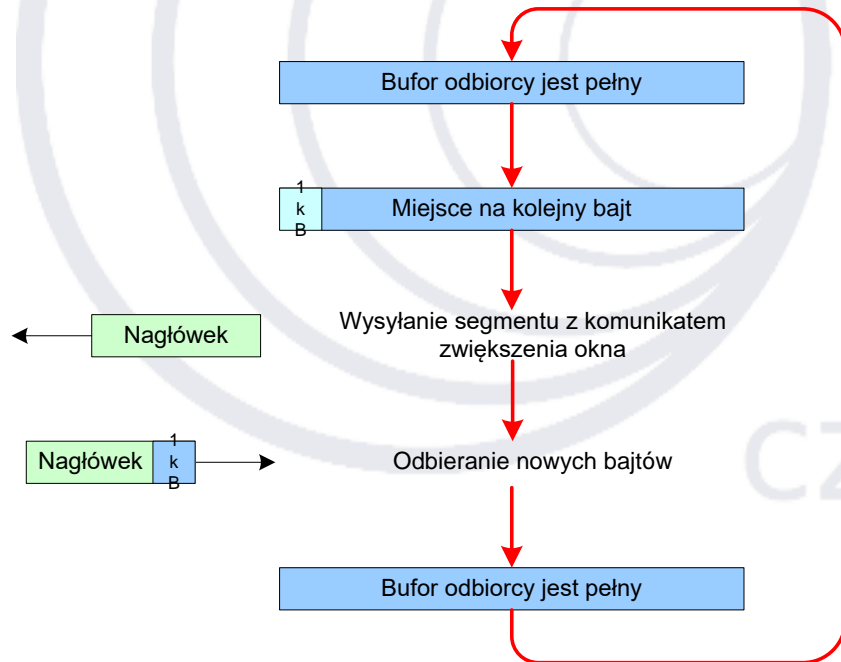
- D – jest to czas propagacji pomiędzy źródłem a odbiorcą w połączeniu TCP (czas przesłania i czas potwierdzenia wynosi $2D$)
- W – rozmiar okna (w oktetach)
- R – szybkość transmisji (bps)

Przykład zarządzania transmisją



Problem *Silly Window Syndrome*

- **Syndrom głupiego okna** (ang. *silly window syndrome*; Clark /1982/) – występował, gdy host nadawca może wysyłać dane w dużych porcjach, lecz działająca na hoście-odbiorcy aplikacja odbiera je w małych porcjach, np. po jednym bajcie



Rozwiązanie: propozycja powstrzymania hosta-odbiorcy od każdorazowego ogłaszania rozmiaru okna → max. wielkość segmentu lub $\frac{1}{2}$ wielkości bufora

Implementacje TCP

- Standard TCP dostarcza ściśle określonej specyfikacji i opisu protokołu używanego pomiędzy
- Opis protokołu zakłada i dopuszcza kilka możliwych opcji implementacyjnych, do których należą :
 - nadawanie (***send policy***)
 - dostarczanie (***deliver policy***)
 - przyjmowanie (***accept policy***): in order, in window
 - retransmisja (***retransmit policy***): first only, batch, individual
 - potwierdzenie (***acknowledge policy***) : immediate, cumulative

User Datagram Protocol (UDP)

- UDP – organizuje wymianę datagramów IP pomiędzy aplikacjami bez nawiązywania połączenia między nimi.
- Jednostkami UDP są segmenty składające się z 8B nagłówka, po którym następuje ładunek danych
- Nie pozwala na: kontrolę przepływu, wykrywanie i niwelowanie błędów transmisji, retransmisje błędnych segmentów,
- Umożliwia: stworzenie dla protokołu IP interfejsu umożliwiającego multipleksowanie i demultipleksowanie pakietów IP należących do różnych procesów, pragnących zachować kontrolę nad przepływem pakietów, poprawnością transmisji oraz zależnościami czasowymi
- Obszar zastosowań: konstrukcje klient-server (np. DNS):
 - aplikacje komunikacji multimedialnej
 - strumieniowe przesyłanie dźwięku i obrazu
 - DNS – rozwiązywanie nazw symbolicznych
 - TFTP – transfer plików

Komunikat UDP

port źródłowy (source port)	port docelowy (destination port)
długość (length)	suma kontrolna (checksum)

Pola **port nadawcy** (ang. *source port*) i **port odbiorcy** (ang. *destination port*) zawierają 16-bitowe numery portów UDP używane do odnajdywania procesów oczekujących na dany datagram (pakiet). Pole port nadawcy jest opcjonalne

Suma kontrolna (*checksum*) – 16-bitowa liczba liczona dla danych jak i nagłówek, weryfikowana po stronie odbiorczej. Pole to jest opcjonalne, ale stanowi jedyną gwarancję, że dane nie zostały uszkodzone (bo IP nie wylicza sum kontrolnych dla danych)

Pole **długość** (ang. *length*) zawiera wartość odpowiadającą liczbie bajtów pakiet UDP wliczając nagłówki i dane; minimalna wartość tego pola wynosi 8 i jest długością samego nagłówka

Protokoły strumieniowania

- **UDP** – stanowi protokół bazowy do strumieniowania w L4
- ***Real-time Transport Protocol, RTP*** – służy transmisjom w czasie rzeczywistym; uwzględnia nr sekwencyjny, typ danych i *timestamp*; *nie gwarantuje QoS*
- ***Real-time Streaming Protocol, RTSP*** – protokół w. aplikacji, który dostarcza danych w czasie rzeczywistym, jak i generuje oraz zarządza strumieniami ciągłych danych (AV)
- ***Real-time Transport Control Protocol, RTCP*** – protokół sterujący, wspierający dla RTP; do jego zadań należy:
 - dostarcza zwrotną informację o poprawności odebranych danych
 - przenosi stały identyfikator transportowy źródła protokołu RTP
 - dopasowuje częstotliwość wysyłanych pakietów kontrolnych do liczby użytkowników sesji,
 - przenosi zminimalizowaną informację kontrolną sesji (opcja)

Protokoły spoza stosu TCP/IP

- **DCCP (*Datagram Congestion Control Protocol*)** RFC4340 – oferuje dostęp do mechanizmów kontroli przeciążeń
 - transmisja zawodna, bez kontroli kolejności
 - do zastosowań z czasowymi ograniczeniami transmisji danych (streaming, VoIP)
- **SCTP (*Stream Control Transmission Protocol*)** RFC2960
 - transmisja niezawodna, z zagwarantowaną kolejnością i brakiem przeciążeń
 - message-oriented (jak w UDP)
 - dedykowany dla VoIP
 - multihoming
 - możliwość transmisji przy użyciu wielu łączy
 - zakończenia połączeń mogą zawierać wiele adresów IP
- **RSVP (*Resource Reservation Protocol*)** RFC2205 – zapewnia konfigurację zasobów w systemach IS (*Integrated Services*), zorientowanych na QoS
 - wspiera zarówno IPv4, jak IPv6

Datagram Congestion Control Protocol

- **DCCP (*Datagram Congestion Control Protocol*)**

- protokół kontroli przeciążeń datagramów (RFC 4340, 4336–2006 rok)
- Uniwersalny protokół transportowy przeznaczony do transmisji danych w trybie rzeczywistym
- Transmisja niezawodna
- Brak gwarancji kolejności dostarczenia datagramów
- Implementuje mechanizmy ECN (Explicit Congestion Notification)
 - powiadamianie o zatorach bez gubienia pakietów
 - realizowany przez urządzenia wspierające (końcowe i pośredniczące)
 - sygnalizowanie nadchodzącego przeciążenia przez routery
- Dodanie znacznika do nagłówka IP
- Przekazanie do odbiorcy, odesłane do nadawcy, który ogranicza transmisję
- w odróżnieniu do TCP, który sygnalizuje przeciążenie przez odrzucanie pakietów
 - liczba serwerów nieobsługujących mechanizmu ECN < 1% (2015r.)
- stosowany przy z czasowych ograniczeniach transmisji (strumieniowanie, gry wieloosobowe, VoIP) – preferowane otrzymywane nowych danych nad dosyłaniem starych

Stream Control Transmission Protocol ^(1/3)

- **SCTP** (*Stream Control Transmission Protocol*) – opracowany w RFC 2960, jako alternatywa dla TCP i UDP

Podobieństwa do TCP	Podobieństwa do UDP
<ul style="list-style-type: none"> • niezawodna transmisja • gwarancje kolejności i brak przeciążeń 	<ul style="list-style-type: none"> • message-oriented (SCTP służy do przesyłania pakietów z ukształtowanymi wiadomościami)

- Dedykowany dla VoIP
- Wieloadresowość (ang. *multihoming*)
➔ transmisja przy użyciu wielu łącz
- Posiada prostszą strukturę pakietu niż TCP, obejmującą wspólny 12B nagłówek oraz różnego typu **data chunks**, tworzące pozostałą część pakietu

port źródłowy (source port)		port docelowy (dest. port)
etykieta weryfikacyjna (verification tag)		
suma kontrolna (checksum)		
chunk 1 type	chunk 1 flags	chunk 1 length
chunk 1 data		
...		
chunk 1 type	chunk 1 flags	chunk 1 length
chunk 1 data		

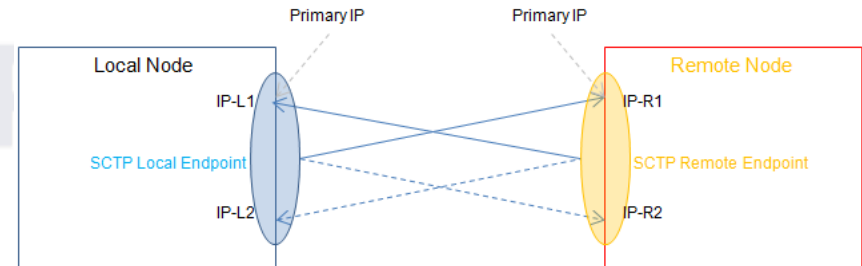
Stream Control Transmission Protocol ^(2/3)

- **SCTP** oferuje redundantne ścieżki celem zwiększenia niezawodności
- Każdy punkt końcowy SCTP musi sprawdzić osiągalność adresów pierwotnego i nadmiarowego dla zdalnego punktu końcowego za pomocą tzw. **pulsu** (ang. **heartbeat**). Punkt końcowy SCTP musi potwierdzać odbierane takich pakietów ze zdalnego punktu końcowego
- Gdy SCTP wysyła wiadomość na zdalny adres, o interfejsie źródłowym decyduje tylko tabela routingu hosta (a nie SCTP)
- Rodzaje wieloadresowości w SCTP:
 - **Asymmetric multi homing** (asymetryczna) – jeden z dwóch punktów końcowych nie wspiera multi homing'u
 - **Local multi homing – Remote single homing** – dla nieosiągalnego zdalnego adresu podstawowego, SCTP kończy się niepowodzeniem, nawet jeśli możliwa jest alternatywna ścieżka
 - **Local single homing – Remote multi homing**

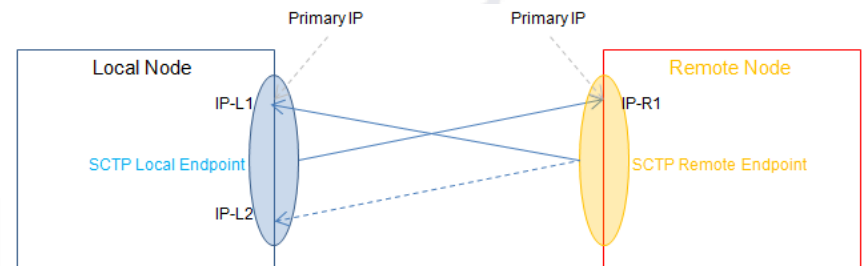
Stream Control Transmission Protocol (3/3)

- Rodzaje wieloadresowości w SCTP:

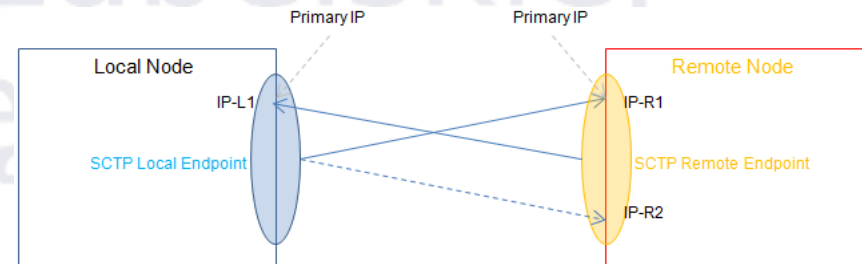
- Asymmetric multi homing** (asymetryczna) – jeden z dwóch punktów końcowych nie wspiera multi homing'u



- Local multi homing - Remote single homing** – dla nieosiągalnego zdalnego adresu podstawowego, SCTP kończy się niepowodzeniem, nawet jeśli możliwa jest alternatywna ścieżka



- Local single homing - Remote multi homing**



Źródło rysunków: Arkrishna - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=38196756>

Resource Reservation Protocol_(1/2)

- **RSVP (*Resource Reservation Protocol*)** RFC 2205, RFC 2210
- Nie jest protokołem transmisji danych ani routingu
- Konfiguracja zasobów w systemach zintegrowanych usług IS
 - Umożliwia rezerwację zasobów sieci dla poszczególnych strumieni danych
 - Wymaga implementacji na każdym routerze IP, zapewniając:
 - przyjmowanie żądań rezerwacji
 - kojarzenie rezerwacji ze strumieniem danych
 - Usługi w architekturze IntServ
 - **BestEffort** – usługa standardowa
 - **Guaranteed Service** – gwarancje parametrów związanych z opóźnieniami
 - **Controlled-load Service** – bezstratny przekaz danych, jakość lepsza niż Best Effort

Integrated
Services

Resource Reservation Protocol_(2/2)

- **RSVP (*Resource Reservation Protocol*)** RFC 2205, RFC 2210
 - umożliwia realizację żądania przez daną aplikację rezerwacji zasobów w sieci
 - niezależny od protokołów trasowania
 - obsługuje transmisje *unicast* oraz *multicast*
 - umożliwia aplikacji inicjującej przesyłanie danych
 - zarezerwowanie przepustowości połączenia
 - zarządzanie zarezerwowanymi na węzłach sieciowych zasobami
 - zwolnienie zasobów po zakończeniu transmisji
 - wymaga okresowego odnawiania dokonanych na każdym węźle rezerwacji, co umożliwia dostosowanie do zmieniającego się ruchu w sieci
 - oferuje dużą skalowalność

Podstawy Sieci Komputerowych

Charakterystyka wybranych protokołów usług sieciowych

System nazw DNS. Protokoły pocztowe

dr hab. inż. Konrad Gromaszek

Plan wykładu

- Wprowadzenie
- System nazw DNS
 - Domeny i rekordy zasobów
 - Protokół komunikacyjny i komunikat DNS
 - Domena odwrotna
- Protokoły poczty elektronicznej
 - SMTP
 - POP3
 - IMAP

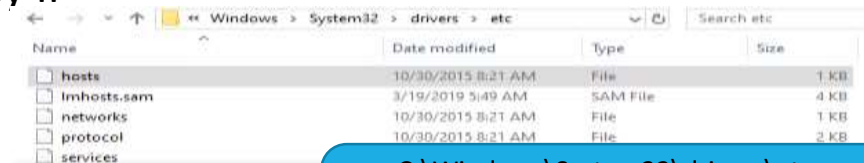
Wprowadzenie

- Aplikacja może odwoływać się do (zasobów) hosta/ów za pomocą adresów IP
- Adresy IP są niewygodne do zapamiętania i nie zawsze są przypisane do określonych zasobów na stałe
- W przeciwieństwie do ludzi, hosty interpretują adresy sieciowe (gł. IP), stąd posługiwanie się nazwami mnemonicznymi wymaga obecności mechanizmu dokonującego ich konwersji na adresy IP
- host.txt i ARPANET

Linux – /etc/hosts.conf

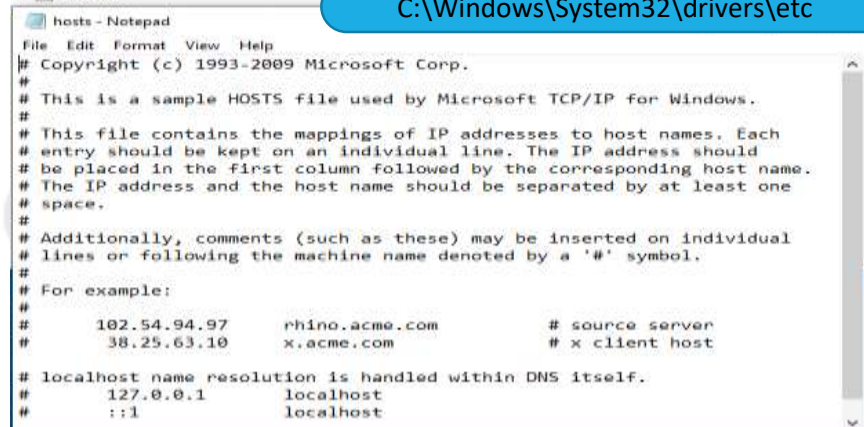
```
127.0.0.1 localhost
127.0.1.1 clientname
192.168.1.x ubuntuobx.hobby-site.org

# The following lines for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```



Name	Date modified	Type	Size
hosts	10/30/2015 8:21 AM	File	1 KB
lmhosts.sam	3/19/2019 5:49 AM	SAM File	4 KB
networks	10/30/2015 8:21 AM	File	1 KB
protocol	10/30/2015 8:21 AM	File	2 KB
services			

C:\Windows\System32\drivers\etc



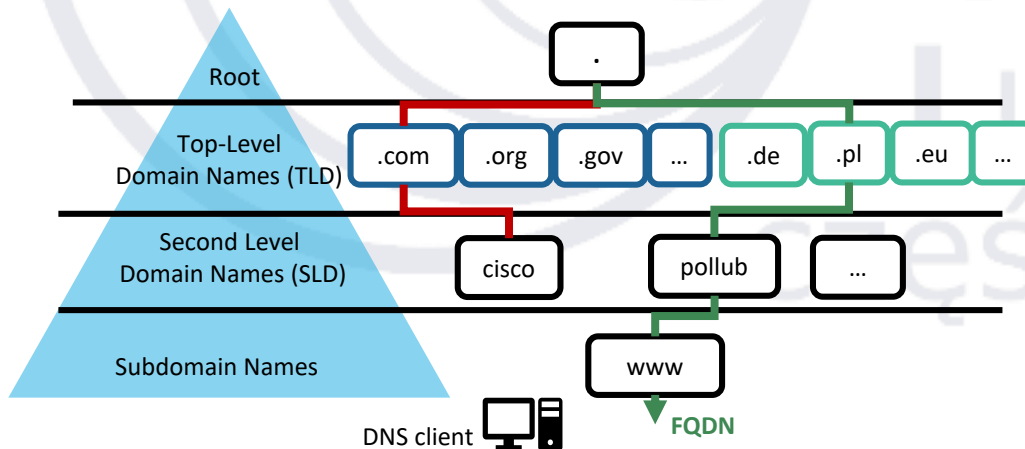
```
hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10       x.acme.com           # x client host
#
# localhost name resolution is handled within DNS itself.
#
#       127.0.0.1         localhost
#       ::1               localhost
```


System nazw domenowych

- **Domain Name System, DNS** bądź też Domain Name Servers, opisany w RFC 1034 i 1035
- Istotą DNS jest nadawanie nazwom mnemotechnicznym struktury hierarchicznej opartej na koncepcji domen i zrealizowanie zarządzania nimi w postaci rozproszonego systemu baz danych.
- Nazwy domen zaczynają się od najbardziej szczegółowej czyli hosta i przesuwają się do najbardziej ogólnej, czyli nazwy root
- Nazwa domenowa zaczyna się od *hosta* i przechodzi całą drogę do *korzenia* jest zwana *wpełni kwalifikowana nazwą domeny* **FQDN** (ang. Fully Qualified Domain Name)
- Przyczyny rozwoju
 - lata 70/80 – znaczny wzrost liczby hostów w ARPANET
 - duży rozmiar pliku /etc/hosts.conf
 - częste zmiany (nazwa-adres) wymagały transferu pliku do wszystkich hostów
 - transfer poczty wymagał specyfikacji hostów pośredniczących, user@host

DNS – Struktura systemu

- Ogólnosiwiatowa sieć serwerów (przechowujących dane na temat adresów domen)
- Struktura drzewiasta
 - serwery „root” (**root servers**) – 13! → <ftp://ftp.rs.internic.net/domain/named.root>
 - serwery główne (**top-level domain servers**) – domeny krajowe, funkcyjne
 - serwery niższego rzędu (**secondary-level domain servers**) (przechowują dane wybranych domen)



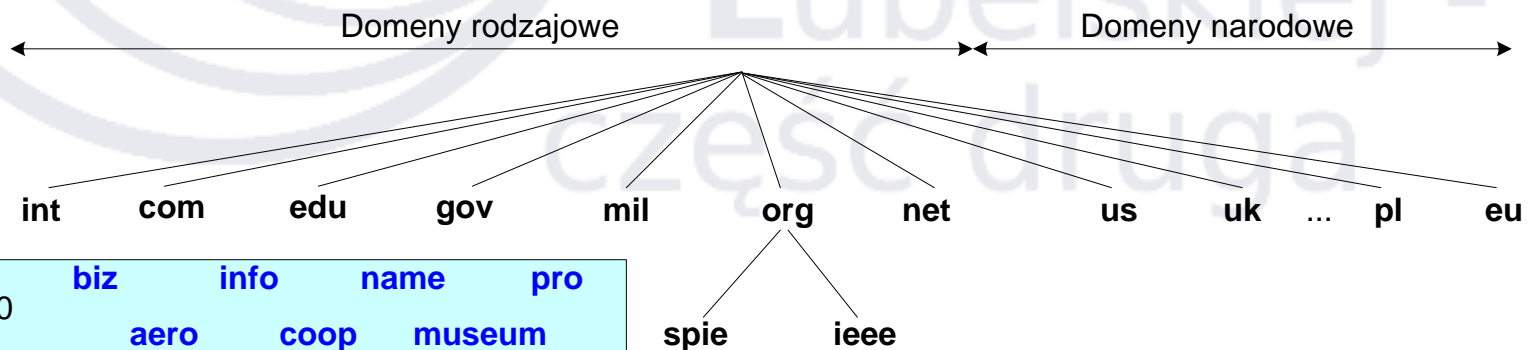
```
; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>"
; configuration file of BIND domain name servers).
;
; This file is made available by InterNIC
; under anonymous FTP as
;   file           /domain/named.cache
;   on server      FTP.INTERNIC.NET
;   -OR-          RS.INTERNIC.NET
;
; last update:     February 20, 2020
; related version of root zone:  2020022000
;
; FORMERLY NS.INTERNIC.NET
;
;
; 3600000      NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000      A      198.41.0.4
A.ROOT-SERVERS.NET. 3600000      AAAA   2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
```

DNS – Struktura systemu

- Serwer nazw ma najczęściej pełne informacje o części całej przestrzeni nazw, określaną **strefą**
- Dane te pobierane są z pliku lokalnego lub z innego serwera nazw, co jest określane mianem, że serwer ma **autoryzację dla strefy** lub jest **wiarygodnym dla całej strefy**.
- Ze względu efektywność zarządzaniu bazą wpisów wszystkich hostów np. w domenie *.pl* stosuje się **delegację strefy**; przechowywanie i uzupełnianie informacji o hostach np. *pol.lublin.pl* przekazane jest serwerowi nazw należącemu do firmy, a serwer zarządzający domeną *lublin.pl* zawiera jedynie wpis o adresie IP serwera nazw domeny *pol.lublin.pl*

Przestrzeń nazw DNS

- Konceptyjny podział Internetu na domeny najwyższego poziomu (TLD)
- Każda domena dzieli się na poddomeny (subdomains), podlegające analogicznemu podziałowi
- Nazwa absolutna zawsze kończy się kropką
- Wielkość liter nie ma znaczenia
- Nazwa każdego członu może liczyć co najwyżej 63 znaki, a domena -255
- Każda domena może być zarejestrowana jako **rodzajowa** lub **narodowa**
- Granice między domenami określa organizacyjna struktura sieci (nie fizyczna)



DNS – Serwery ROOT

- ftp://ftp.rs.internic.net/domain/named.root
- https://www.iana.org/domains/root/servers
- Fizycznie każdy root serwer ma kilkadziesiąt kopii rozmieszczonych po całym świecie
- 1094 instancje root serwerów obsługiwanych przez 12 niezależnych operatorów (marzec 2020)

List of Root Servers

HOSTNAME	IP ADDRESSES	MANAGER
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project



Źródło: <https://www.iana.org/domains/root/servers>

Źródło: <https://coednssecurity.in/images/rootloc.jpg>

DNS – Domeny TLD

- **Top Level Domains (TLD)** – *domeny najwyższego poziomu* są tworzone i zarządzane przez **IANA** (*Internet Assigned Numbers Authority*) oraz **ICANN** (*Internet Corporation for Assigned Names and Numbers*)

- typy domen TLD

- **Funkcjonalne** (gTLD – generic TLD)

- **Niesponsorowane**

.com	.org	.net	.int	.edu	.gov	.mil
komercyjne	organizacje	internetowe	organizacje międzynarodowe	uczelnie wyższe w USA	organizacje rządowe w USA	organizacje wojskowe USA

- **Sponsorowane**

- .aero – transport lotniczy
 - .mobi – telefonia komórkowa

- **Infrastrukturalne**

- .arpa – infrastruktura sieciowa internetu (Reverse DNS)
 - .root – niektóre główne serwery DNS

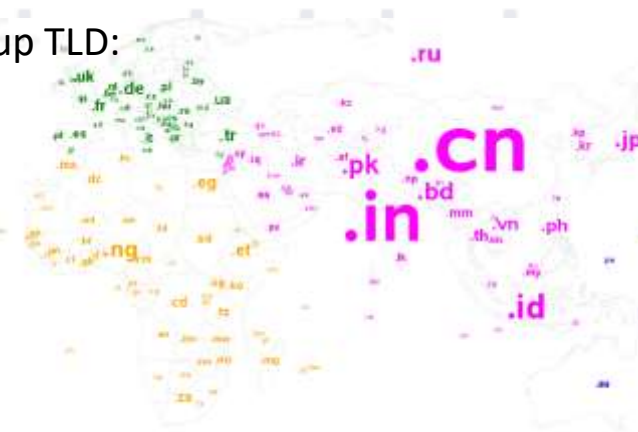
- Usługowe (.tel, .post)

- Inne (.kids, .eco)

- Krajowe (ccTLD – country code TLD)

DNS – Domeny ccTLD

- **country code Top Level Domains (ccTLD)** – krajowe domeny najwyższego poziomu
 - zawsze dwuliterowe
 - odpowiadają kodom krajów ze standardu ISO 3166-1
 - przyporządkowane także do odrębnych obszarów geograficznych (np. Antyle Holenderskie [.an])
- W 2015r. IANA wprowadziła następujący podział grup TLD:
 - Infrastructure top-level domain (ARPA)
 - Generic top-level domains (gTLD)
 - Restricted generic top-level domains (grTLD)
 - Sponsored top-level domains (sTLD)
 - Country code top-level domains (ccTLD)
 - Test top-level domains (tTLD)



Źródło: http://robslink.com/SAS/democd93/country_code_tld.htm

Autor: Allison Robert, "A map of country code top-level domains (ccTLD)", SAS Learning Post June 22, 2017, <https://blogs.sas.com/content/sastraining/2017/06/22/map-of-ccTld/>

DNS – domeny drugiego poziomu

- **subdomains** – domeny drugiego poziomu funkcjonują jako poddomeny TLD
- Rodzaje domen drugiego poziomu na przykładzie TLD **.pl**
 - Regionalne
 - lublin.pl
 - lubelskie.pl
 - Funkcjonalne
 - com.pl – biznesowe
 - gov.pl – rządowe
 - org.pl – organizacje pozarządowe
 - Należące do firm lub osób prywatnych
 - zus.pl
 - x-kom.pl
 - filmweb.pl

DNS – Nazwy domen

- Poszczególne nazwy węzłów (ang. *nodes*) stanowią etykiety tekstowe o długości od 1 do 63 znaków, oddzielane kropką „.”
- Dozwolone znaki
 - Standardowe:
 - litery
 - cyfry
 - znak „-”
 - Znaki narodowe **Internationalized Domain Name (IDN)**
 - zawierają zestaw znaków spoza kodu ASCII, (np. ą, ć, ę, ł, ń, ó, ś, ź, ż w j. polskim)
 - przekształcane do 7b znaków – Punycode (RFC 3490)
 - technicznie domena posiada prefiks „xn-” przed nazwą domeny (np.: www.**żółtałódź.pl** = **www.xn--tad-fnac58bc46bka.pl**)
 - obsługiwane przez wszystkie przeglądarki i programy pocztowe

DNS – Administracja systemem

- Za administrację systemem DNS odpowiadają dwie instytucje

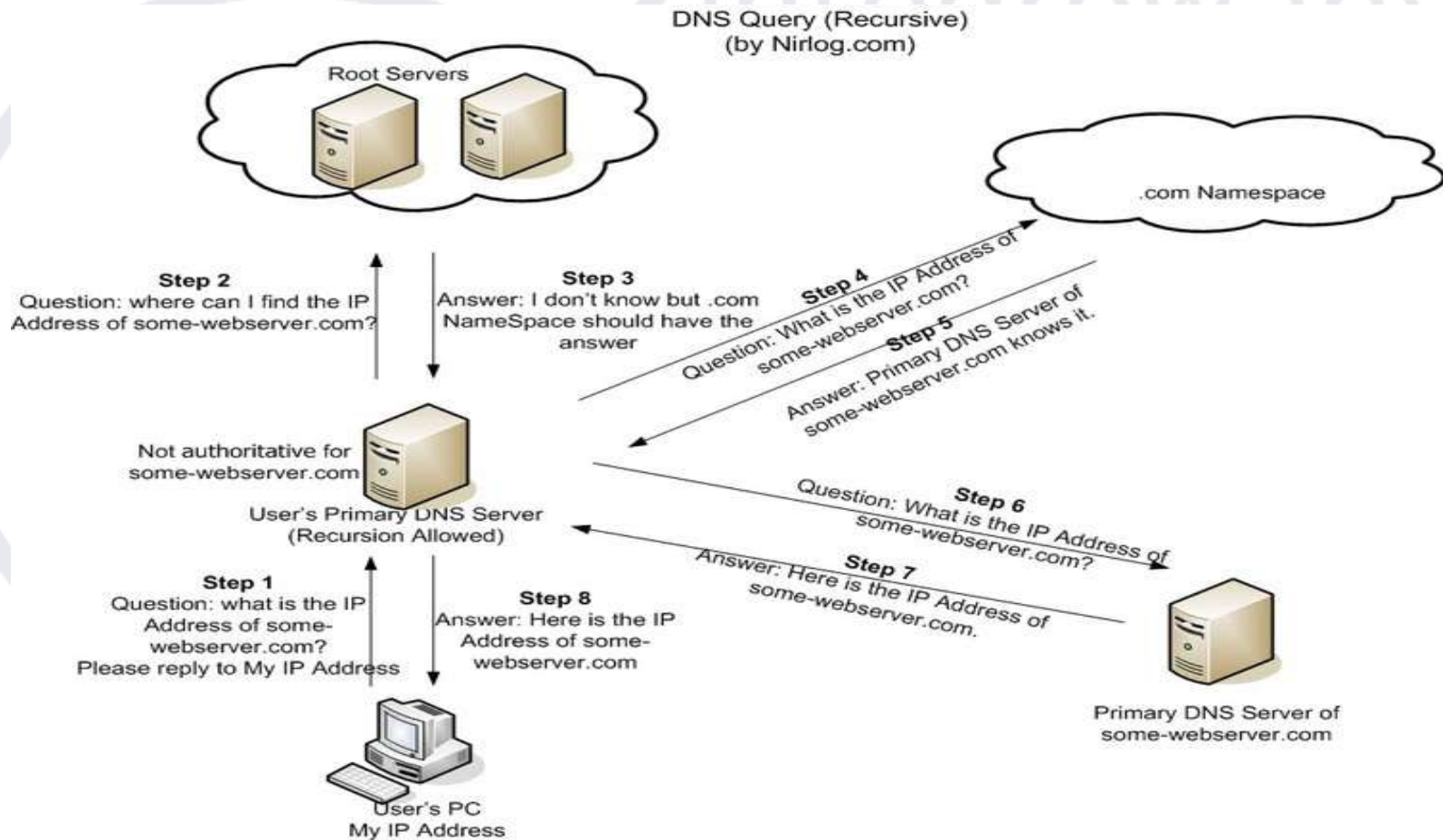
Internet Assigned Numbers Authority (IANA)	Internet Corporation for Assigned Names and Numbers (ICANN)
<ul style="list-style-type: none">• zarządzanie domenami TLD• kordynacja nad działaniem mechanizmu DNS	<ul style="list-style-type: none">• przyznawanie nazw domen internetowych• ustalanie struktury domen• administracja rozdziałem adresów IP i protokołami Internetu

- Rozdzielają domeny najwyższego poziomu TLD pośród krajów i organizacji wraz z delegacją praw do zarządzania
- Polska – rząd powierzył nadzór nad domeną **.pl** *Naukowej i Akademickiej Sieci Komputerowej (NASK)* (w tym również com.pl, gov.pl, biz.pl, org.pl, net.pl, ...)
 - Powstał wiosną 1991 roku przy Uniwersytecie Warszawskim (17 sierpnia 1991 – pierwsza łączność IP z uniwersytetem w Kopenhadze)
- Rejestr domen internetowych .pl
 - Mechanizm automatyczny w oparciu o protokół EPP (*Extensible Provisioning Protocol*)
 - Model registry-registrar (NASK-partnerzy) (Home.pl, nazwa.pl, www.netart-registrar.com)

Serwery DNS

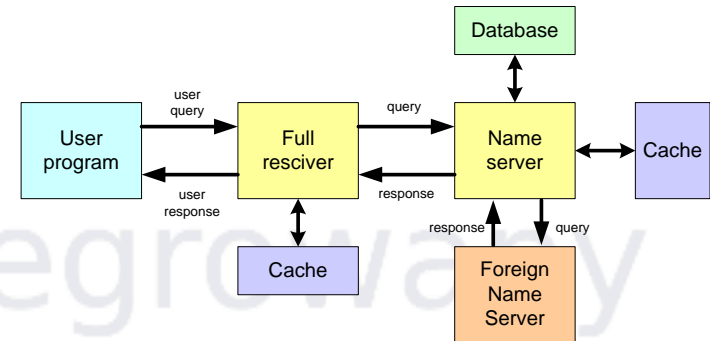
- **ROOT SERVER** – zna wszystkie *top level domains* w sieci Internet. Informacje o hostach jest zbierana z tych domen poprzez przeprowadzenie zapytania dla komputera z innej strefy (*name server query*) ROOT SERVER może stwierdzić miarodajnie o istnieniu danego host'a w tej poddomenie.
- **DNS MASTER SERVER** – "miarodajny" dla całego obszaru bieżącej domeny, prowadzi bazy danych dla całej strefy. Istnieją dwa rodzaje MS: **PRIMARY MASTER SERVER** oraz **SECONDARY MASTER SERVER**, Może się zdarzyć, że serwer jest zarazem MASTER SERVER dla kilku domen – dla jednych PRIMARY MASTER SERVER, dla innych SECONDARY MASTER SERVER.
- **CACHING SERVER** – wszystkie serwery (PRIMARY jak i SECONDARY) prowadzą cache'owanie informacji, które otrzymują Korzystają z nich aż do zdezaktualizowania danych. Wygasanie określone jest w polu TTL , które jest zawsze dołączane do danych dostarczanych serwerowi. CS nie mają pełnomocnictw dla żadnej strefy, w związku z tym nie zarządzają żadnymi bazami danych. Mogą natomiast odpowiadać poprzez wysyłanie zapytań (*queries*) do innych serwerów posiadających takie pełnomocnictwa.

Funkcjonowanie DNS



Źródło: MSOffice Obrazy on-line, autor: Nieznany, licencja: CC BY-SA

Zapytania DNS



- Realizowane pomiędzy klientem (resolverem) a serwerem DNS
- Rekurencyjne
 - zmuszenie odpytywanego serwera do znalezienia wiarygodnej informacji / zwrot błędu
 - odpytywany serwer nie znając zapytania, odpytuje inne serwery DNS
 - umożliwia zapamiętanie odwzorowania w pamięci serwera (DNS caching)
 - realizowane wyłącznie przez:
 - serwery lokalne dla lokalnych hostów (resolverów) (realizowane później iteracyjnie)
 - serwery przekazujące (forwarding)
- Iteracyjne
 - odpytywany serwer odpowiada najlepszą znaną mu odpowiedzią (np. adresem serwerów autorytatywnych dla danej domeny)
 - odpytywany serwer nie łączy się z innymi serwerami

Odpowiedzi DNS

- Autorytatywne
 - dotyczą domen w strefie dla której dany serwer ma autoryzację
 - ustalane na podstawie bazy danych serwera
 - ma ustawiony bit uwierzytelniania (Authoritative Answer, AA)
- Nieautorytatywne
 - dane pochodzą spoza strefy zarządzanej przez dany serwer
 - odpowiedzi takie są buforowane na serwerze przez określony czas, a następnie usuwane
- Programy użytkowe – diagnostyka DNS
 - **nslookup** – pyta jednocześnie tylko jednego serwera DNS, wykorzystuje zdefiniowane w resolverze listy przeszukiwania (ale nie korzysta z /etc/hosts), nie sprawdza numeru seryjnego SOA
 - **dig** – występuje głównie w systemach z rodziny Linux, jest poleceniem bez interaktywnego trybu pracy i nie korzysta z listy wyszukiwania

DNS – rekordy zasobów

A	(address record) – mapuje nazwę domeny na adres IPv4)
AAAA	(IPv6 address record) – mapuje nazwę domeny na adres IPv6
CNAME	(canonical name record) – ustawia alias domeny
MX	(mail exchange record) – mapuje nazwę domeny na nazwę serwera pocztowego oraz określa priorytet
PTR	(pointer record) – mapuje adres IP (v4/6) na nazwę kanoniczną hosta oraz umożliwia odwrotną translację adresów
NS	(name server record) – mapuje nazwę domenową na listę serwerów DNS (danej domeny)
SOA	(start of authority record) – ustala serwer DNS dostarczający autorytatywne informacje o domenie
SRV	(service record) – dodatkowe informacje dotyczące usługi udostępnianej przez serwer
TXT	(text record) – uzupełniające informacje tekstowe, np. specyfikacja

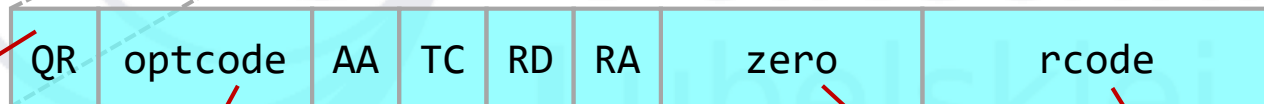
Protokół komunikacyjny DNS

- DNS jako protokół komunikacyjny zapewnia sposób łączenia klientów z serwerami DNS oraz determinuje zestaw zaleceń związanych z aktualizacjami wpisów w bazach domen internetowych
- Stosuje
 - UDP, port 53 – do komunikacji *klient – server* (pakiety $\leq 512B$)
 - TCP, port 53 – do komunikacji *server master – server slave* (pakiety $> 512B$ ze względu na dodatkowe pole – długość zapytania/odpowiedzi)
- Okresowa aktualizacja wpisów DNS = propagacja informacji nawet do kilkudziesięciu godzin

Komunikat DNS

- Komunikat DNS ma 12 bajtowy nagłówek stałej długości i cztery pola zmiennej długości:

0	8	16	24	31
Identifier		Flags and codes		
QuestionCount		Answer Record Count		
Name Server Count (Authority Rec.)		Answer Record Count		



określa czy komunikat jest zapytaniem czy odpowiedzią

rodzaj zapytania (standardowe, zwrotne, status serwera, zarezerwowane)

AA = odpowiedź autorytatywna
TC = odpowiedź nie zmieściła się w pojedynczym pakiecie UDP
RD = klient żąda rekurencji
RA = serwer obsługuje rekurencje

p. zarezerwowane

kod odpowiedzi
(brak błędu, błąd formatu/serwera/nazwy, odrzucono, zarezerwowane)

DNS – Domena .arpa

- **.arpa** to specjalna domena TLD, przeznaczona do obsługi infrastruktury sieciowej Internetu
- W ramach niej zdefiniowane zostały poddomeny:
 - `in-addr.arpa` – zapewnia mapowanie IPv4 na nazwy
 - `ip6.arpa` – mapowanie IPv6 na nazwy
 - `e164.arpa` – mapowanie numerów telefonicznych, zgodnych ze standardem E.164 na URI
- **Reverse DNS** – system serwerów pełniących funkcję odwrotną do DNS
 - realizowany dzięki domenie `in-addr.arpa`
 - wykorzystuje wskaźnik (pointer) PTR w tablicy DNS

Domena odwrotna DNS

- Mechanizm translacji z adresu IPv4 na pełną nazwę komputera, gdy jest to niezbędne
- Do tego celu stworzono specjalną domenę **.in-adress.arpa**
- Przykład:

Adres: 212.182.64.82
ReverseName: 82.64.182.212.in-addr.arpa
Nazwa hosta: olimp.pollub.pl

```
Last login: Sat Mar 28 14:54:48 on ttys001
[kgr@MBAxKGR ~ % nslookup
[> set type=ptr
[> 82.64.182.212.in-addr.arpa
Server:      192.168.1.1
Address:     192.168.1.1#53

Non-authoritative answer:
82.64.182.212.in-addr.arpa      name = olimp.pollub.pl.

Authoritative answers can be found from:
```

- Zapewnia to mechanizm bezbłędnego mapowania (mapping) adresu IP na nazwę hosta

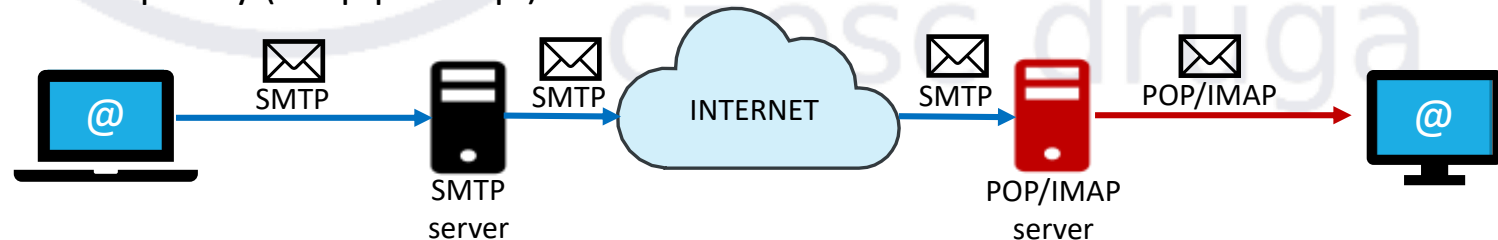
Kwestie związane z DNS

- Serwery DNS:
 - **BIND – Berkley(*Buggy*) Internet Net Domain** – najpopularniejszy, duża wydajność, <http://www.isc.org/products/BIND>
 - Konfiguracja => `/etc/named.conf`
- Bezpieczeństwo
 - Ataki **DDoS** (Distributed Denial of Service) oraz **Man-in-the-middle**
 - DNSEC – rozszerzenie DNS o uwierzytelnianie źródeł danych, oparte na podpisach cyfrowych, praktycznie nieużywany ze względu na brak akceptacji przez społeczność Internetu, a tym samym aplikacje i urządzenia

Protokoły poczty elektronicznej

- **Simple Mail Transfer Protocol, SMTP**

- stanowi podstawowy protokół transmisji poczty w Internecie, dawniej określany **Outgoing Mail Server, OMS** lub **Mail Transfer Agent, MTA**
- umożliwia wysyłanie i transport poczty elektronicznej email poprzez różne środowiska systemowe
- wyspecyfikowany w RFC 821 i RFC 532
- usługa (demon) SMTP wykorzystuje TCP port 25 (lub 587)
- bezpieczne przesyłanie SMTP zapewnia protokół TLS (= SMTPS)
- formalnie używany przez serwery pocztowe do wysyłania i odbierania wiadomości
- aplikacje klienckie (Mail, Outlook, Thunderbird, ...) używają go tylko do wysyłania poczty (smtp.pollub.pl)

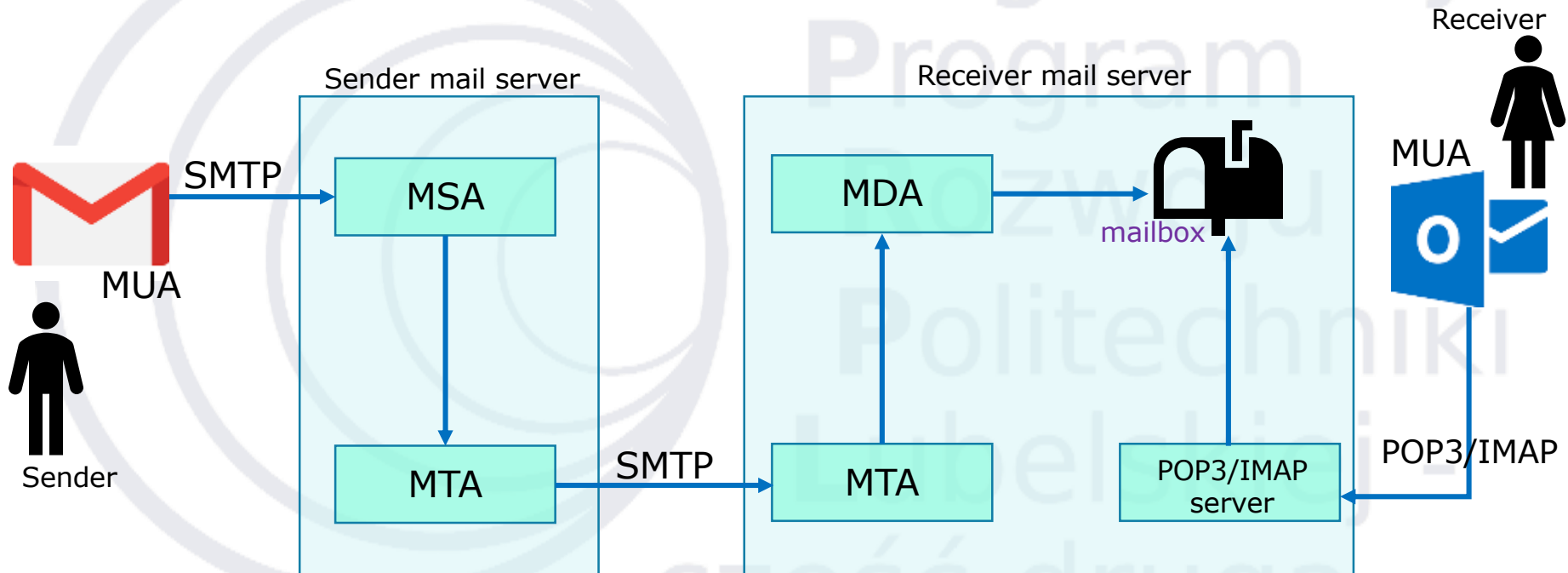


SMTP – przesyłanie wiadomości

1. Klient poczty nawiązuje połączenie ze swoim serwerem
 - Mail client (*Mail User Agent*, **MUA**) ==> łączy się z *Mail Submission Agent*, **MSA** na porcie **25** (lub 587)
2. Dalej transmisja przebiega pomiędzy serwerami SMTP (sposób określenia serwera poczty ==> DNS)
 - **MSA** – przesyła dane do *Mail Transfer Agent*, **MTA**, gdzie na podstawie rekordu **MX** z systemu DNS ustala adres serwera(ów) poczty domeny docelowej
3. Wersja bez pośredników: serwer wysyłający przekazuje pocztę do serwera odbiorcy

Przypadek z pośrednikami: dzięki postaci adresu użytkownik@firma.pl serwer wysyłający:
 - MX odbiera wiadomości dla danej domeny (jako punkt docelowy albo pośrednik) i kieruje do docelowego *Mail Delivery Agent*, **MDA**, który przechowuje wiadomość i udostępnia ją do pobrania dla użytkownika-odbiorcy
4. Klient pocztowy odbiorcy (**MUA**) nawiązuje połączenie z **MDA**, dokonuje autentykacji i pobiera wiadomość protokołem **IMAP** lub **POP3**

SMTP – przesyłanie wiadomości



*Mail User Agent, **MUA***
*Mail Submission Agent, **MSA***
*Mail Transfer Agent, **MTA***
*Mail Delivery Agent, **MDA***

Jak działa SMTP?

```
konradg@localhost:/root
Plik Edycja Widok Terminal Zakładki Pomoc
[konradg@localhost root]$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 localhost.localdomain ESMTP Sendmail 8.13.8/8.13.8; Mon, 15
Oct 2007 23:44:42 +0200
helo komputer.firma.pl
250 localhost.localdomain Hello localhost.localdomain [127.0.0.
1], pleased to meet you
mail from: uzytkownik@firma.pl
250 2.1.0 uzytkownik@firma.pl... Sender ok
rcpt to: root@localhost
250 2.1.5 root@localhost... Recipient ok
data
354 Enter mail, end with "." on a line by itself
subject: test

tresc
.
250 2.0.0 l9FLig7L002673 Message accepted for delivery
quit
221 2.0.0 localhost.localdomain closing connection
Connection closed by foreign host.
You have mail in /var/spool/mail/root
[konradg@localhost root]$
```

```
root@localhost:~
Plik Edycja Widok Terminal Zakładki Pomoc
You have new mail in /var/spool/mail/root
[root@localhost ~]# mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/root": 3 messages 3 new
>N 1 uzytkownik@firma.pl Mon Oct 15 23:49 12/428 "test"
N 2 logwatch@localhost.l Tue Oct 16 00:40 89/2786 "Logwat"
N 3 root@localhost.local Tue Oct 16 00:43 18/835 "Anacro"
&
```

Polecenia SMTP

- Podstawowa implementacja SMTP obsługuje osiem poleceń
 - **HELO** odpowiada za identyfikację klienta na serwerze i wszczęcie komunikacji
 - **MAIL** – określa adres nadawcy
 - **RCPT** – specyfikuje adres odbiorcy
 - **DATA** – treść wiadomości
 - **QUIT** – żądanie zakończenia połączenia
 - **RSET** – przerwanie bieżącej transakcji i reset obu stron połączenia: dane o nadawcy, adresacie oraz dane wiadomości są tracone
 - **VRFY** – zapytanie wysyłającego o weryfikację adresu, pod który mają być przesłane wiadomości
 - **NOOP** – zmusza serwer do przesłania odpowiedzi OK z kodem (200)
- Rozszerzenie ESMTP wprowadza dodatkowe komendy:
 - EHLO, SEND, SOML, SAML, EXPN, HELO, TURN

Pola nagłówka SMTP

- **From:** – adres pocztowy autora, pole ustalane podczas tworzenia listu
- **Sender:** – identyfikator nadawcy listu (osoba / system / proces)
- **To:** – lista adresatów porozdzielanych przecinkami
- **Cc:** (Carbon Copy) – lista adresów, na które zostanie wysłana kopia wiadomości
- **Bcc:** (Blind Carbon Copy) – lista adresów, na które zostanie wysłana ukryta kopia
- **Subject:** – temat wiadomości
- **Date:** – data i czas wysyłania wiadomości
- **Reply-To:** – jeśli nadawca chce aby odpowiedź została wysłana na inny adres
- **Return-Path:** – pole określa trasę powrotną, dodawane przez ostatni system biorący udział w przesyłaniu wiadomości, zawiera wiersze:
- **Organization:** – pole dodatkowe na nazwę organizacji
- **Message-ID:** – unikalny identyfikator stworzony przez serwer wysyłający
- **X-nazwa_pola:** – pola nie określone przez RFC, zawierające różne dodatkowe informacje
- **In-Reply-To:** – identyfikuje poprzedni list, na który obecny jest odpowiedzią, może zawierać Message-ID:
- **Encrypted:** – w przypadku zaszyfrowania wskazuje oprogramowanie lub metodę użytą do zaszyfrowania wiadomości

SMTP– kody błędów

- Komunikując się z serwerem SMTP uzyskuje się odpowiedzi zaczynające się od numeru =>określenie źródła problemu

Kod	Komunikat EN	Komunikat PL
211	System status or system help reply	Stan systemu albo odpowiedź pomocy
214	Help message	Wiadomość pomocy
220	Domain service ready. Ready to start TLS.	Serwer gotowy
221	Domain service closing transmission channel	Serwer zamyka kanał transmisji
250	OK. queing for node started. Requested mail action okay. completed	Potwierdzenie wykonania polecenia
251	OK. no messages waiting for node . User not local. will forward to <forwardpath>.	Użytkownik nie znajduje się na tym serwerze, wiadomość przekierowana do <forwardpath>
452	Requested action not taken: insufficient system storage.	Żądana akcja nie zostanie podjęta niewystarczające zasoby pamięci system.
501	Syntax error. No parameters allowed	Błąd składni dla polecenia nie są dostępne parametry
554	Transaction failed	Transakcja nie powiodła się

Protokół MIME

- Protokół **MIME** (***M**ultipurpose **I**nternet **M**ail **E**xtensions*) jest rozszerzeniem poczty internetowej
 - zdefiniowany w RFC 2045-2049
 - definiuje różne typy przesyłanych danych (w odróżnieniu od SMTP) oraz daje możliwość dzielenia treści listu na części (różnego typu)
 - dzięki niemu można przysyłać w email różnorodne pliki traktowane jako załączniki
 - określa sposoby kodowania przesyłanych danych tak by w efekcie zawartość spełniała ograniczenia SMTP
- Dodatkowe nagłówki MIME
 - **Content-Type** – typ danych zawartych w wiadomości (text, image, audio, video, application, message, model, multipart wraz z wydzielonymi podtypami)
 - **Content-Transfer-Encoding** – określa sposób kodowania transmitowanych danych
 - **Version** – wersja standardu
 - **Content-ID** – wyznacza wiadomość właściwą
 - **Content-Description** – komentarz do zawartości
 - **Content-Features** – dodatkowe informacje o wiadomości

POP3

- **Post Office Protocol 3 (POP3)** to protokół klientów pocztowych do odbierania wiadomości email z serwerów pocztowych, nasłuchuje na porcie 110 TCP, niemniej wymaga komunikacji szyfrowanej po inicjalizacji połączenia, w oparciu o komendę STLS (STARTTLS) => POP3S stosuje połączenie TLS lub SSL na porcie 995;
- Aktualnie częściowo zastępowany przez protokół IMAP
- Sposób działania:
 - połączenie klienta z serwerem
 - pobranie wszystkich wiadomości z serwera do klienta
 - zapisanie wiadomości w systemie klienta
 - usunięcie wiadomości z zasobów serwera

IMAP

- **Internet Message Access Protocol (IMAP)** to protokół klientów pocztowych, służący do pobierania wiadomości email z serwera i zaprojektowany do obsługi skrzynki pocztowej przez wielu klientów
 - umożliwia przechowywanie wiadomości na serwerze i podzielenie ich na różne foldery => możliwość zarządzania wiadomościami
 - zdefiniowany w RFC 2060
 - korzysta z portu 143 w ramach protokołu TCP
 - IMAPS (rozszerzony o SSL) wykorzystuje port 993
- **Przewaga względem POP3**
 - możliwość równoległego podłączenia wielu klientów do jednej skrzynki wraz z zapewnieniem ich synchronizacji
 - dostęp do pojedynczych elementów wiadomości email typu MIME
 - monitorowanie stanu wiadomości z użyciem flag
 - możliwość tworzenia, zmiany nazw i usuwania folderów (mailbox)
 - wyszukiwanie wiadomości po stronie serwera bez konieczności ich pobierania

Podstawy Sieci Komputerowych

Charakterystyka wybranych protokołów usług sieciowych

Protokoły FTP i HTTP. Protokoły tunelowania.

dr hab. inż. Konrad Gromaszek

Plan wykładu

- Wprowadzenie
- Protokół komunikacyjny
- File Transfer Protocol
- Trivial File Transfer Protocol
- Hypertext Transfer Protocol
 - HTTPS
 - Transport Layer Security

Zintegrowany
Program
Rozwoju
Politechniki
Lubelskiej -
część druga

Protokół komunikacyjny

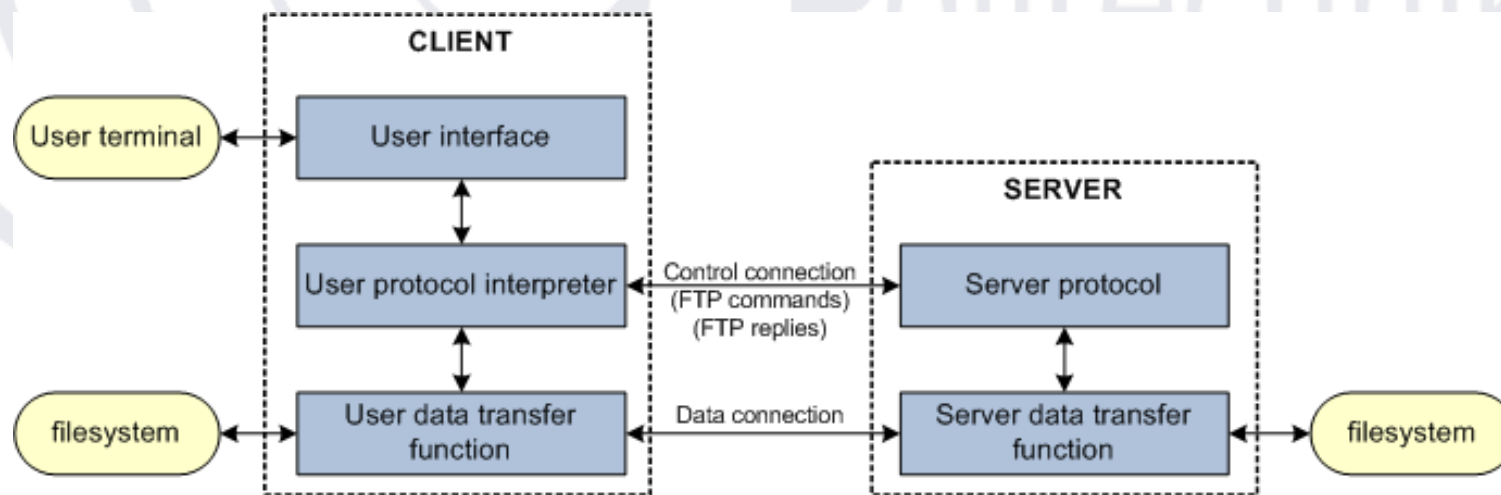
- Protokół komunikacyjny to zbiór reguł i kroków postępowania, wykonywanych automatycznie przez urządzenia komunikacyjne w celu nawiązania łączności i wymiany danych
- Elementy składowe protokołu komunikacyjnego:
 - Procedura początkowa (ustalenie parametrów połączenia: adresy, szybkość, rozmiar buforów itp.)
 - Procedura transmisji danych (format ramki)
 - Procedura analizy poprawności przekazu (sumy kontrolne)
 - Procedury retransmisji
 - Procedury zakończenia połączenia
 - Procedury (de)fragmentacji danych

File Transfer Protocol – połączenia

- **FTP (ang. *File Transfer Protocol*)** wykorzystuje dwa połączenia TCP do przesyłania plików:
 - **połączenie sterujące** (ang. ***control connection***) – zestawiane w architekturze klient/serwer:
 - serwer otwiera pasywnie dedykowany port FTP (najczęściej 21) i oczekuje na połączenie klienta
 - klient wykonuje aktywne otwarcie portu i zestawia połączenie kontrolne, które pozostaje aktywne przez cały czas trwania komunikacji klienta z serwerem
 - połączenie sterujące jest wykorzystywane do przesyłania oraz odbierania rozkazów pomiędzy klientem i serwerem
 - (posiada ustawione *minimize delay* w TOS nagłówka IPv4)
 - **połączenie danych** (ang. ***data connection***)- zestawiane każdorazowo przy transmisji pojedynczego pliku
 - (posiada ustawioną flagę *maximize throughput* w polu TOS, nagłówka IPv4)

Transfer plików przez FTP

- Zadaniem interfejsu użytkownika jest:
 - translacja wykonywanych działań/akcji (kopiowanie, zakładanie katalogów, kasowanie itp..) na komendy FTP i przesyłanie ich przez łącze sterujące
 - interpretacja otrzymywanych od serwera odpowiedzi i przedstawianie ich w formie zrozumiałej dla użytkownika (np. komunikaty błędów)

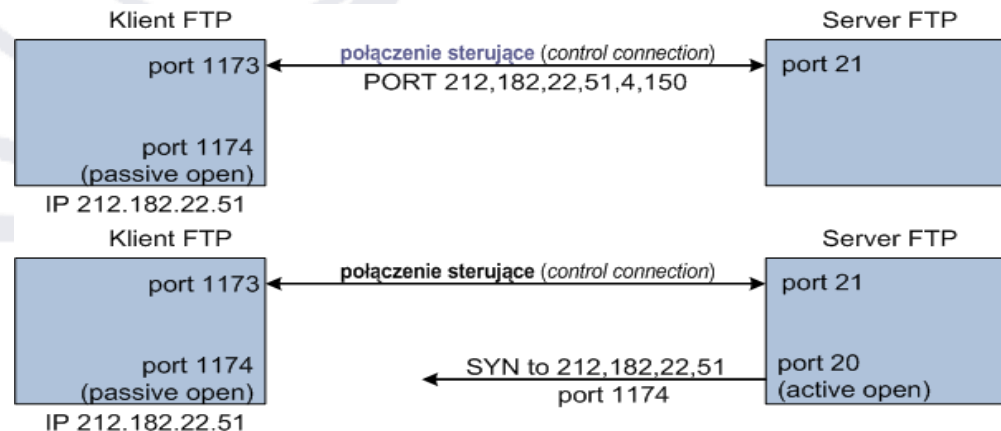


Nawiązywanie połączenia FTP_(1/2)

- Istnieją trzy sposoby wykorzystania łącza *data connection*:
 - transmisja pliku z klienta do serwera
 - transmisja pliku z serwera do klienta
 - transmisja listingu plików lub katalogów z serwera do klienta
- Proces nawiązywania połączenia:
 - wywołanie *data connection* jest kontrolowane przez klienta, ponieważ to ona wysyła komendę wymuszającą transfer pliku (pobranie pliku, zapis pliku, wylistowanie katalogu)
 - stacja klienta dokonuje wyboru portu dla łącza danych, z puli portów dostępnych i przeprowadza pasywne otwarcie wskazanego portu (prowadzi nasłuch)
 - stacja klienta wysyła wybrany numer portu przez łącze sterujące komendą PORT do serwera, serwer odbiera numer i otwiera aktywne połączenie przez wskazany port ze stacją klienta

Nawiązywanie połączenia FTP_(2/2)

- Stacja klienta wybiera numery portów dla łącz odpowiednio: 1173 – kontrolne, 1174 – danych i otwiera pasywnie port 1174.
- Następnie wysyłana jest komenda PORT, której argumentami jest sześć liczb 8-bit w kodzie ASCII oddzielonych przecinkami. Pierwsze cztery liczby określają adres IP stacji klienta (tu 140.252.13.34), a następne dwie określają 16-bit numer portu liczony jako: $4 \times 256 + 150 = 1174$.
- W tym momencie serwer otwiera aktywnie port 20 (domyślnie) dla łącza danych stacji klienta



Komendy FTP (linia sterująca)

- Wykorzystanie protokołu FTP opiera się na przesyłaniu komend do serwera oraz odbieraniu od niego odpowiedzi poprzez połączenie sterujące
- Rozkazy są przesyłane jako znaki w formacie NVT ASCII i muszą być zakończone parą znaków kontrolnych CR/LF
- Długość rozkazu wynosi 3 lub 4 bajty i składa się z drukowanych znaków ASCII, czasem z dodatkowymi argumentami

Rozkaz	Opis
ABORT	Przerwanie wcześniejszego rozkazu lub transferu danych
LIST filelist	Wylistowanie katalogów lub plików
PASS password	Hasło na serwerze
PORT n1,...,n6	Adres IP klienta (n1.n2.n3.n4) i nr portu (n5x256+n6)
QUIT	Zamknięcie sesji z serwerem
RETR filename	Otrzymanie (get) pliku
STOR filename	Wysłanie (put) pliku
SYST	Serwer zwraca typ pliku
TYPE type	Określenie typu pliku: A=ASCII, I=image
USER username	Użytkownik na serwerze

FTP – reprezentacje danych

- **Reprezentacje danych**

- **ASCII** – wykorzystywany do transmisji danych tekstowych, konwertowany do 8 bitowego kodu ASCII
- **Image/Binary** – wysyłanie pliku jako strumienia bajtów,
- **EBCDIC** – wysyłanie tekstu (plain text) w dla systemu kodowania EBCDIC
- **Local mode** – tryb dla dwóch komputerów w identycznej konfiguracji we własnym trybie kodowania bez konieczności konwersji na ASCII

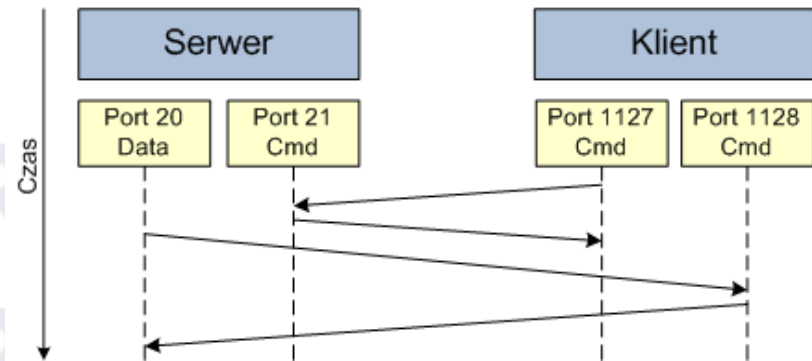
- **Tryby transferu danych**

- *Transmisja strumieniowa* – bez dodatkowego przetwarzania, pozostałe funkcjonalności przerzucone na protokół TCP
- *Transmisja blokowa* – FTP dzieli dane na bloki (nagłówek, długość, dane) i takie segmenty przekazuje do przesłania przez TCP
- *Transmisja skompresowana*

FTP – tryb aktywny

- W trybie aktywnym, klient korzystając z protokołu TCP łączy się z nieuprzywilejowanym portem $N > 1024$ na port 21 (command) serwera
- Następnie klient zaczyna nasłuchiwanie na porcie $N+1$
- Wówczas serwer z portu 20 (data) powinien nawiązać połączenie TCP do klienta na podany mu port $N+1$

TRYB AKTYWNY



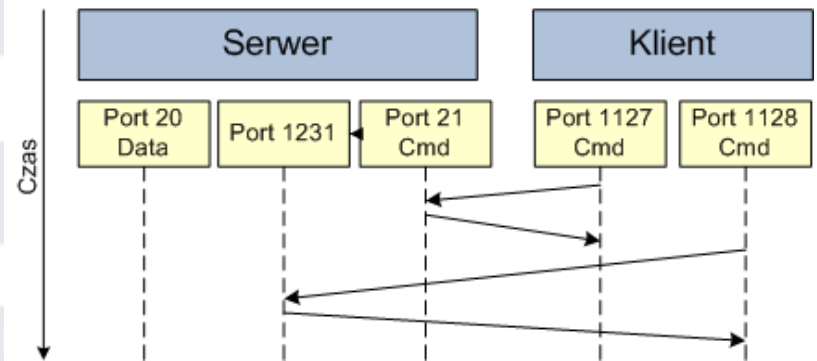
Problemy trybu aktywnego:

- nawiązywanie połączenia z portu 20 serwera na wysoki port klienta (FIREWALL !!!)
- Przechodzenie takich połączeń przez NAT

FTP – tryb pasywny

- Rozpoczynając transmisję klient otwiera dwa lokalne nieuprzywilejowane porty ($N > 1024$ i najczęściej $N+1$)
- Z pierwszego z nich połączenie nawiązywane jest na port 21 serwera z poleceniem PASV
- Serwer otwiera nieuprzywilejowany port $M > 1024$ a następnie wysyła do klienta polecenie PORT M
- Klient nawiązuje połączenie ze swojego drugiego portu $N+1$ na M serwera w celu dokonania transferu danych
- Port serwera 20 nie uczestniczy w połączeniu pasywnym

TRYB PASYWNY



Problemy trybu aktywnego:

- Tryb pasywny otwiera dużą dziurę w systemie zabezpieczeń serwera FTP

Rozwiązanie:

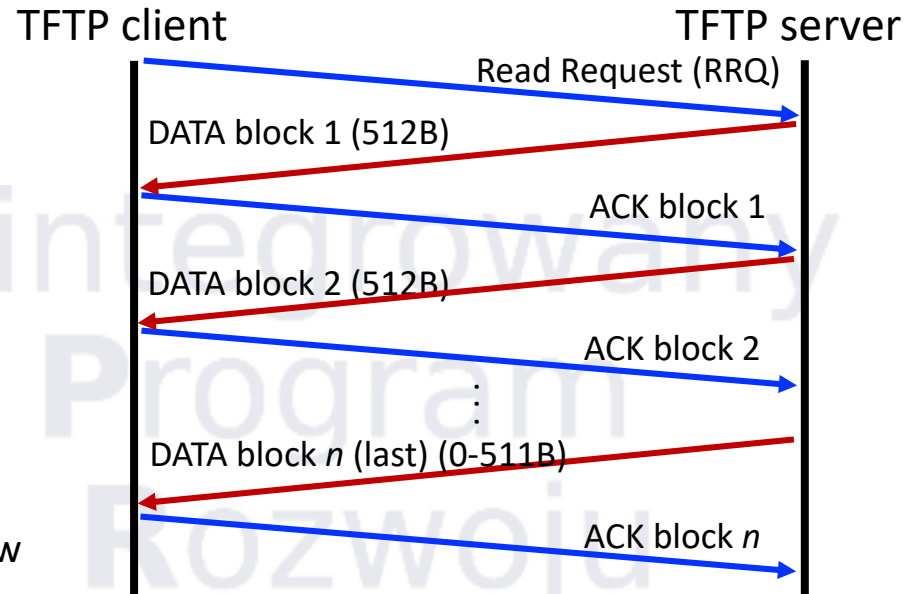
- wykorzystanie określonego zakresu portów wysokich + konfig. firewalla

File Transfer Protocol – podsumowanie

- **FTP (ang. File Transfer Protocol)** – protokół transferu plików, RFC 959 typu klient-serwer wykorzystujący dwa połączenia TCP
 - Połączenie kontrolne – przesyłanie poleceń
 - Połączenie do transmisji danych
- **Może działać w dwóch trybach**
 - Tryb aktywny: Port 21 – linia poleceń (zest. klient) i Port 20 – transfer danych (zest. serwer)
 - Tryb pasywny: Port 21 – linia poleceń (zest. klient) i Port >1024 – transfer danych (zest. klienta)
- **Tryby dostępu**
 - Anonimowy – bez hasła uwierzytelniającego
 - Autoryzowany – w oparciu o login i hasło (niezaszyfrowane)
- **Wersje bezpieczne**
 - FTPS – z wykorzystaniem SSL/TSL
 - SFTP – z wykorzystaniem SSH

TFTP

- **Trivial File Transfer Protocol**
- Uproszczony względem FTP
 - brak wyświetlania katalogów
 - brak uwierzytelniania użytkowników
- Podstawowe funkcje
 - odczyt/wysyłanie plików w blokach po 512B z/na zdalny host
 - realizowany na porcie 69 UDP (nasłuch), a transmisja danych na innych portach
- Gwarancja niezawodności transmisji
 - wymóg odesłania pakietu potwierdzającego dla każdego wysłanego pakietu
 - retransmisja ostatniego pakietu po „przeterminowaniu” czasu oczekiwania
- Tryby transmisji
 - netascii – 7-bitowy kod ASCII + specyfikacja protokołu Telnet (RFC 854)
 - oktet – przesyłanie informacji bit po bicie (jak tryb binarny w FTP)



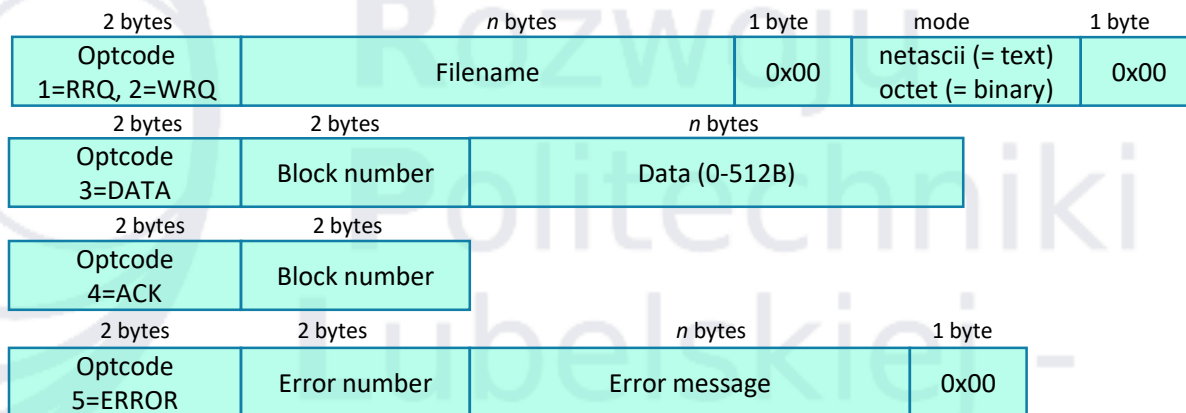
TFTP

• Pięć typów pakietów

- **RRQ** – żądanie odczytu
- **WRQ** – żądanie zapisu
- **DATA** – dane
- **ACK** – potwierdzenie
- **ERROR** – błąd

Pola pakietów RRQ i WRQ

- Opcode – kod operacji.
- Filename – nazwa pliku, który ma zostać przesłany
- Filename terminator – 8-bitowe pole (0) - koniec pola Filename.
- Mode – pole zawierające tekst: "netascii", "octet" lub "mail"
- Mode terminator – 8-bitowe pole zawiera wartość zero (koniec pola Mode)



1. Niezdefiniowany. Sprawdź komunikat o błędzie (o ile jest)
2. Nie znaleziono pliku
3. Naruszenie dostępu
4. Dysk pełny / przekroczona alokacja
5. Zabroniona operacja TFTP
6. Nieznana tożsamość transmisji
7. Plik już istnieje
8. Nie ma takiego użytkownika

Kody błędów

Zestawienie cech FTP i TFTP

cecha	FTP	TFTP
autentykacja	w oparciu o login i hasło	brak
połączenie	TCP	UDP (+ mechanizmy TFTP)
algorytm protokołu	mechanizmy TCP (sliding window, flow control)	Potwierdzenia przesyłanych pakietów
złożoność	złożony	Prosty, z małym narzutem (wykorzystywany w bootloaderach)
kanały transmisji danych	Dwa odrębne kanały (linia danych i sterująca)	Dane i informacje przesyłane w oparciu o jedno połączenie

Hypertext Transfer Protocol

- **Hypertext Transfer Protocol**

- „protokół przesyłania dokumentów hipertekstowych w sieci WWW”
- dokumenty hipertekstowe – „ustrukturalizowany tekst z hiperlinkami pomiędzy węzłami zawierającymi tekst”
- działa jako protokół żądania(klient) – odpowiedzi(server)
- jest protokołem bezstanowym – serwer nie monitoruje zmian/statusów/historii
- swojej komunikacji z klientem (server side sessions: cookies, hidden forms vars)

- **Sesja HTTP**

- sekwencja transakcji żądania-odpowiedzi (request-response)
- zwykle w oparciu o protokół TCP i port 80
- serwer nasłuchuje na porcie i odpowiada na żądania klienta (ze statusem oraz treścią wiadomości, najczęściej zasobem/contentem)

Metody HTTP

- **Metody HTTP** – określają akcje jakie mogą zostać wykonane na zasobach

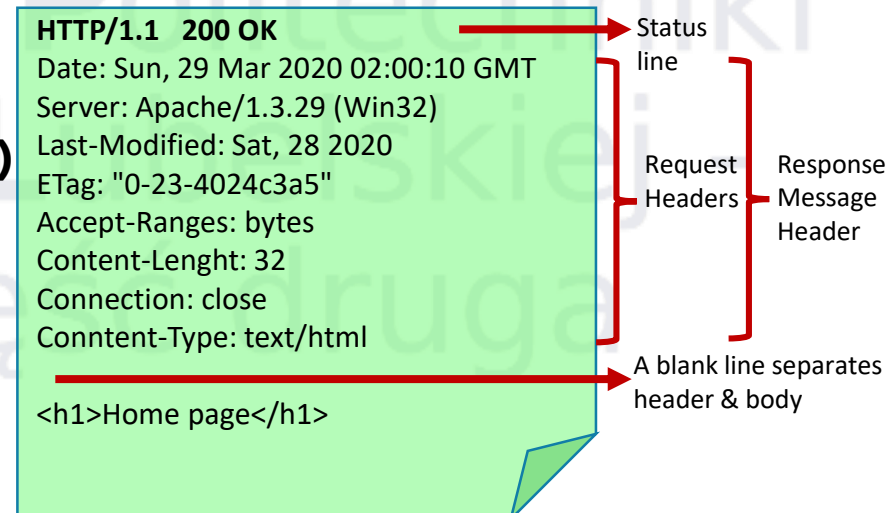
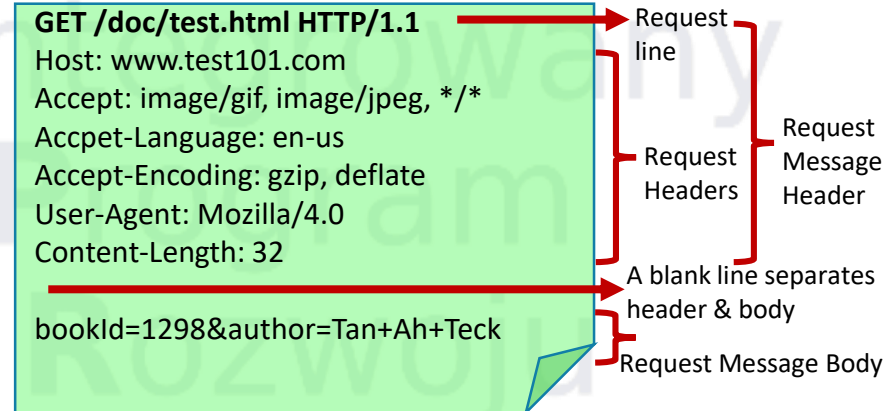
- | | |
|--------------------|---|
| HTTP 1.0
(1996) | <ul style="list-style-type: none">• GET – pobranie z serwera do klienta zasobu wskazanego przez URI• POST – żądanie odebrania przez serwer danych (np. formularza, komentarza, rekordu bazy danych) przesłanych od klienta w ramach zapytania identyfikowanych poprzez URI (nie jest idempotentne) |
| HTTP 1.1
(1997) | <ul style="list-style-type: none">• HEAD – pobranie informacji o danym zasobie z serwera (ale bez jego zawartości)• OPTIONS – informacje o metodach obsługiwanych w ramach danego zasobu• PUT – żądanie odebrania i utworzenia/zaktualizowania danych wysłanych od klienta do serwera (jest idempotentne)• DELETE – żądanie usunięcia zasobu z serwera• TRACE – odesłanie przez serwer odebranego zapytania w celu przetestowania ewentualnych narzutów/modyfikacji serwerów pośredniczących• CONNECT – konwersja żądania na tunel TCP/IP (stos. komunikacji szyfrowanej)• PATCH – częściowa modyfikacja zasobu |

Metody HTTP

- **Kody stanu** zawarte w pierwszej linii odpowiedzi (status + tekstowe rozszerzenie)
 - 1XX – Informacyjny
 - 2XX – Powodzenie
 - 3XX – Przekierowanie
 - 4XX – Błąd klienta
 - 5XX – Błąd serwera
- **Persystencja połączeń**
 - mechanizm KeepAlive
 - podtrzymywane połączenia do obsługi więcej niż jednego żądania
 - zmniejszenie odczuwalnego opóźnienia
 - brak konieczności zestawiania TCP 3-way-handshake dla każdego requestu
 - dodatkowe mechanizmy przyspieszające transmisję
 - **chunked transfer encoding** – przesyłanie strumieniowe zamiast buforowanego
 - **HTTP Pipelining** – wysyłanie wielu requestów przed odebraniem pojedynczych odpowiedzi
 - **byte serving** – transmisja konkretnego fragmentu danych z serwera wskazanego przez klienta
- **Szyfrowanie** – rozszerzenie w ramach HTTPS

Wiadomość HTTP

- **Komunikaty w formie tekstu w kodzie ASCII**
- **Request message (żądanie)**
 - Linia żądania, np.:
GET /images/logo.png HTTP/1.1
 - Nagłówek żądania (np kodowanie)
 - Pusta linia
 - Opcjonalna treść żądania
- **Response message (odpowiedź)**
 - Linia statusu (kod, treść), np.: HTTP/1.1 200 OK
 - Nagłówki odpowiedzi
 - Pusta linia
 - Opcjonalna treść odpowiedzi



HTTPS

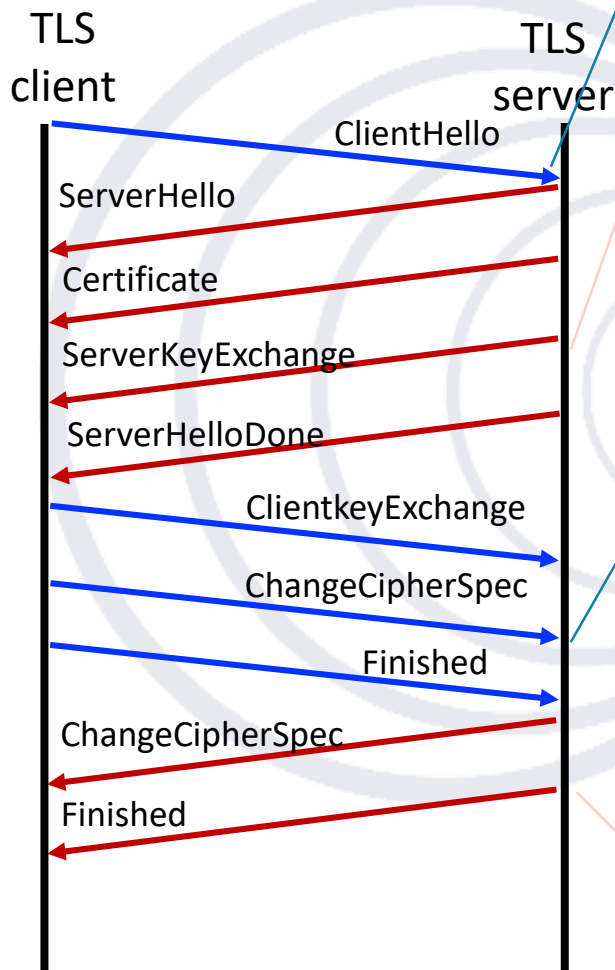
- Rozszerzenie HTTP w celu:
 - autentyfikacji strony WWW
 - zapewnienia prywatności danych (dostępu)
 - zapewnieniu integralności danych w procesie transmisji (zapobieganie atakom tyłu Man-In-The-Middle)
- Wymaga Certyfikatów Cyfrowych po stronie serwera
 - kiedyś kosztowne – metody płatności
 - od 2016 dostępne także darmowo – Let's encrypt (ISRG)
- Metody szyfrowania oparte o SSL (pierwotnie) oraz w oparciu o TLS (obecnie)
- Działa na porcie 443 (protokół TCP), w przeglądarce jako https://
- Sposób działania: wymiana kluczy TLS → żądanie HTTP zakodowane w kanale TLS
- Problem z serwowaniem wielu domen rozwiązany przez mechanizm SNI – Server Name Identification

Transport Layer Security, TLS

- Stanowi rozwinięcie protokołu *Secure Socket Layer*, SSL 3.1 → TLS1.0
- Obejmuje zestaw algorytmów, technik i schematów szyfrujących
- Funkcjonuje w warstwie prezentacji, może zabezpieczać protokoły z warstwy aplikacji (HTTP, POP3, IMAP, FTP, NNTP, SIP)
- Zapewnia
 - poufność i integralność transmisji
 - uwierzytelnianie serwera
- Opiera się na
 - szyfrowaniu asymetrycznym
 - certyfikatach X.509 – Infrastruktura klucza publicznego (Urzędy Certyfikacji)
- Wykorzystuje (TLS3.0) algorytmy/protokoły: AES, DES, 3DES, IDEA, RC2, RC4, RSA, DSS, Diffiego-Hellmana

TLS

(zestawianie kanału komunikacji)



- Klient wysyła do serwera zgłoszenie zawierające m.in. wspieraną wersję SSL, dozwolone sposoby szyfrowania i kompresji danych, identyfikator sesji oraz **liczbę losową** używaną potem przy generowaniu klucza

- Serwer odpowiada komunikatem (zawierającym m.in.: wersję protokołu SSL, rodzaj szyfrowania i kompresji, oraz **liczbę losową**)
- Serwer wysyła swój certyfikat pozwalając klientowi na sprawdzenie swojej tożsamości (opcja)
- Serwer wysyła informację o swoim **kluczu publicznym** (rodzaj i długość klucza określa typ algorytmu, przesłany w poprzednim komunikacie)
- Serwer zawiadamia, że klient może przejść do następnej fazy zestawiania połączenia

- Klient Klient wysyła serwerowi **wstępny klucz sesji**, zaszyfrowany za pomocą **klucza publicznego serwera**
- W oparciu o ustalone w poprzednich komunikatach 2 losowych liczbach (klient + serwer) oraz wstępny klucz sesji (ustalony przez klienta) obie strony generują **klucz sesji** (typowo DES) używany do faktycznej wymiany danych
- Klient zawiadamia serwer o możliwości przełączenia się na komunikację szyfrowaną
- ... oraz o gotowości odbioru zaszyfrowanych danych

- Serwer zawiadamia o wykonaniu polecenia (odtąd tylko zaszyfrowane dane) ...
- ...i od razu wypróbowuje mechanizm, bo ostatni komunikat jest już wysyłany bezpiecznym kanałem

Podstawy Sieci Komputerowych

Podstawy bezpieczeństwa w sieciach komputerowych

dr hab. inż. Konrad Gromaszek

Plan wykładu

- Wprowadzenie
- Aspekty bezpieczeństwa
- Bezpieczeństwo sieciowe względem modelu OSI
- Adres MAC
- Adresacja w warstwie sieciowej
- Charakterystyka wybranych algorytmów i protokołów bezpieczeństwa, stosowanych w sieciach komputerowych
 - IPSec, VPN, MD5, HMAC, Kerberos, PGP

Wprowadzenie

- Marginalne znaczenie bezpieczeństwa w pierwszych sieciach
- Społeczeństwo informacyjne => wzrost znaczenia bezpieczeństwa
- Bezpieczeństwo przesyłanej informacji przez sieć jest zadaniem obszernym i sprowadza się do uniemożliwienia odczytu tej informacji przez ludzi, którzy nie mają uprawnień.
- Wymaganie hermetyczności informacji:
 - zapewnienie autentyczności informacji
 - wymaganie legimatyizacji przesyłanej informacji
- Wiele problemów z bezpieczeństwem w sieciach komputerowych powodowanych jest przez świadome działania dokonywane przez różnych ludzi, z rozmaitych pobudek.
 - najczęściej osoby wtajemniczone, pałające żądzą zemsty

Aspekty bezpieczeństwa

BEZPIECZEŃSTWO

POUFNOŚĆ

Ochrona informacji
przed dostaniem się w
niepowołane ręce

IDENTYFIKACJA

Możliwość określenia z
kim ma się do czynienia
przed powierzeniem
ważnych materiałów

NIEZAPRZECZALNOŚĆ

Niemożliwość wyparcia
się podjętych wcześniej
zobowiązań

INTEGRALNOŚĆ

Zapewnienie
nienaruszalności
wysyłanej informacji

Bezpieczeństwo sieciowe w kontekście modelu odniesienia

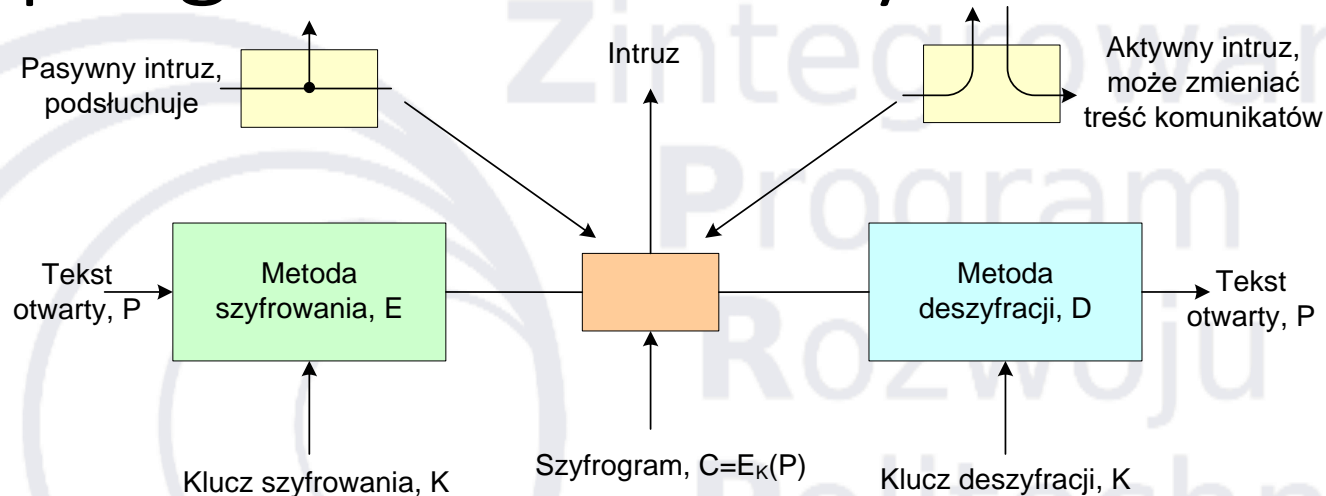
- **Warstwa fizyczna** – zamykanie linii transmisyjnych w rurociągach wypełnionych gazem pod dużym ciśnieniem
- **Warstwa łącza danych** – szyfrowanie łącza (*link encryption*) na maszynie wysyłającej i rozszyfrowanie na maszynie docelowej, => uniemożliwia ochronę wybranych sesji (nr kart kredytowych)
- **Warstwa sieciowa** – bezpieczeństwo poprzez firewalle, dokonujące selekcjonowania pakietów + zabezpieczenia protokołu IP
- **Warstwa transportowa** – szyfrowanie kompletnych połączeń na całej drodze przepływu pomiędzy komunikującymi się procesami
- **Warstwa aplikacji** – identyfikacja i odpowiedzialność, wymuszane przez procesy logowania

Za wyjątkiem **w. fizycznej** mechanizmy zabezpieczające bazują technikach kryptograficznych

Kryptografia – geneza

- Kryptografia = gr. „pisanie w ukryciu”
- Techniki kryptograficzne:
 - **Szyfrowanie** (ang. *ciphering*) – transformowanie informacji wejściowej bit po bicie / znak po znaku, bez względu na strukturę językową
 - **Kodowanie** (ang. *coding*) – polega na zastępowaniu każdego słowa informacji wejściowej innym słowem
- Historycznie kryptografią zainteresowane były grupy ludzi:
 - wojskowi
 - pamiętnikarze
 - kochankowie
- Kryptografię kształtowały przez lata zastosowania militarne:
 - słabo opłacani urzędnicy
 - stosowanie algorytmów na tyle prostych by poradzić mógł sobie z nimi nawet słabo wyposażony szyfrant na polu walki

Kryptografia – terminy



- Sztuka łamania szyfrów nazywana jest **kryptoanalizą**, a całokształt środków podejmowanych w celu jej utrudniania nazywa się kryptologią
- Zapis formalny: $C=E_K(P)$ (tekst otwarty P, podlega szyfrowaniu przez funkcję E z parametrem – kluczem K, wynikiem jest szyfrogram C)
- W celu odtworzenia tekstu: $P=D_K(C) = D_K(E_K(P))$
- Każda metoda szyfrowania może być złamana przez **kryptoanalityka**

Kryptografia – zagadnienia

- Zasady kryptografii
 - zasada Kerckhoffa
 - redundancja i aktualność
- Szyfry podstawieniowe i przestawieniowe
- System **kluczy jednorodnych** (ang. *one-time-pad*)
- Szyfrowanie kluczami symetrycznymi (s-box i p-box) i AES
- Klucze asymetryczne (Diffie i Hellman – 1976) i RSA
- Zasady projektowania protokołu uwierzytelniania

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
P	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	i	o	n	
d	o	l	a	r	s	t	
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Tekst otwarty:

pleasetransferonemilliondollarsto
myswissbankaccountsixtwo

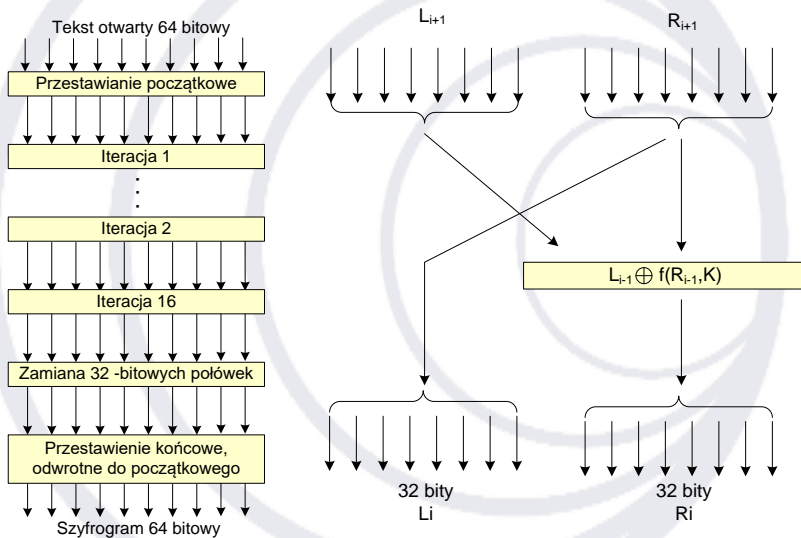
Tekst zaszyfrowany:

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUEOERIRICXB

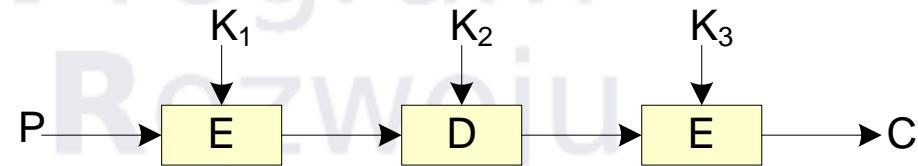
Alicja	↑	↘	↑	↘	↔	↗	↔	↘	↑	↘
	1	0	1	0	1	1	0	0	1	0
Bogdan	+	+	×	+	+	×	+	+	×	×
	1	0	1	0	1	1	0	1	0	0
Weryfikacja	✓			✓		✓	✓			✓
Klucz	1			0		1	0			0

Data Encryption Standard (DES) I

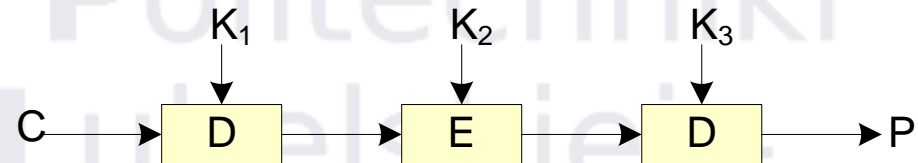
Tripple DES (3DES)



a) szyfrowanie



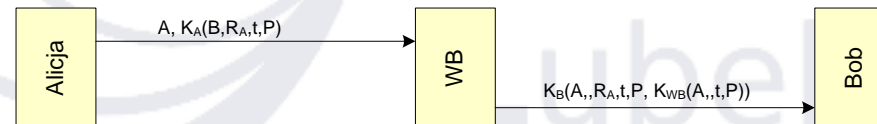
b) odszyfrowanie



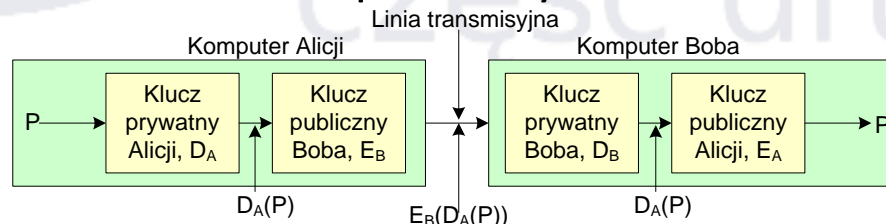
- Trzykrotne wykonanie DES'a, znany jako standard 8732
- Użycie schematu EDE (zamiast EEE) jest podyktowane względami kompatybilności wstecz, gdy $K_1=K_2$ to uzyskuje się DES z kluczem K_1 .

Podpis cyfrowy

- Rezultaty potrzeb metody umożliwiającej podpisywanie dokumentów w sposób wykluczający fałszerstwo
- Związany z następującymi wymaganiami:
 - odbiorca może zweryfikować autentyczność podpisu nadawcy
 - nadawca nie może wyprzeć się przesłania komunikatu o określonej treści
 - odbiorca nie ma możliwości zmiany treści komunikatu po jego otrzymaniu
- Podpisy oparte na kluczach symetrycznych



- Podpisy oparte na kluczach publicznych



Skróty komunikatów MD

- Jedną z krytykowanych cech podpisu cyfrowego jest fakt, że łączy on w sobie dwie oddzielne funkcje; uwierzytelnienie i tajność
- W wielu przypadkach uwierzytelnienie jest pożądane, a tajność – nie
- Idea schematu opiera się na jednokierunkowej funkcji haszującej, otrzymującej jako argument dowolnie duży fragment tekstu otwartego i zwraca łańcuch znaków o ustalonej długości
- Funkcja ta to tzw. **abstrakt komunikatu** (*message digest*, MD)
- Posiada następujące własności (jak MD5, czy SHA-1):
 - obliczenie MD(P) na podstawie P daje się wykonać łatwo
 - obliczenie P na podstawie MD(P) jest w praktyce niewykonalne
 - dla danego P nie da się znaleźć takiego Q <> P, że MD(P) = MD(Q)
 - zmiana nawet 1 bitu w tekście wejściowym powoduje drastyczne zmiany w wartości funkcji

Bezpieczeństwo komunikacji

- IETF zdawała sobie sprawę z niedostatecznego poziomu bezpieczeństwa w Internecie, a dodanie mechanizmów zabezpieczających => trudne ze względu na różnice w poglądach co do ich umiejscowienia
 - większość ekspertów: prawdziwe bezpieczeństwo tylko na całej drodze przepływu => ale konieczność zmiany istniejących aplikacji (nierealne)
 - przeciwnicy end-to-end: ukrycie mechanizmów zabezpieczeń na poziomie warstwy sieciowej
- Rezultat: IPsec (RFC 2401,2402, 2406,...)
 - nie wszyscy użytkownicy życzą sobie szyfrowania (koszt obliczeniowy), zamiast uczynić je opcjonalnym, uczyniono je obowiązkowym, umożliwiając jednocześnie użycie neutralnego algorytmu szyfrowania (*null algorithm*), szybkiego i prostego w implementacji

Bezpieczeństwo komunikacji

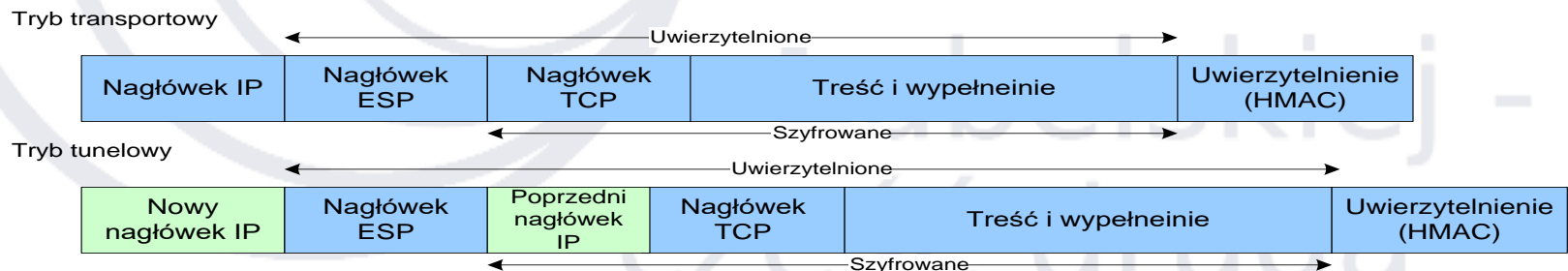
- Marginalne znaczenie bezpieczeństwa w pierwszych sieciach
- IPSec – zespół usług i granulacji do wyboru. Podstawowe usługi to: poufność, kontrola integralności danych oraz ochrona przed atakiem powtarzającym, zrealizowane na bazie szyfrowania z kluczem symetrycznym ze względu na efektywność.
- Zorientowany na połączenie
- Korzysta z 2 tybów ISAKMP (Internet Security Association and Key Management Protocol) oraz IKE (Interent Key Exchange)
- Zapory sieciowe

IPSec

- IPsec – zespół usług i granulacji do wyboru. Podstawowe usługi to: poufność, kontrola integralności danych oraz ochrona przed atakiem powtarzającym, zrealizowane na bazie szyfrowania z kluczem symetrycznym ze względu na efektywność
- Wybór algorytmów szyfrowania (gdyby złamano obecnie używany)
- Zorientowany na połączenie
- Zróżnicowanie **granulacji** = wybór rodzaju ochrony połączeń
- Złożony z 2 części: **ISAKMP** (*Internet Security Association and Key Management Protocol*) oraz **IKE** (*Internet Key Exchange*)

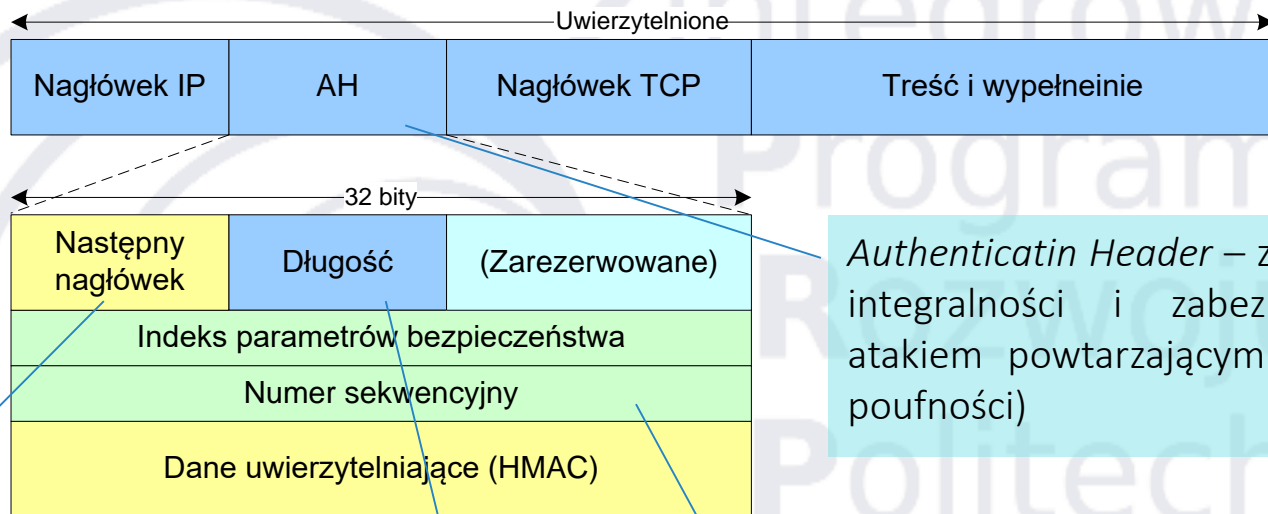
IPSec

- Korzysta z 2 trybów:
 - **transport mode** – między nagł. IP a nagł. TCP wstawiany jest nagłówek IPsec, zawierający identyfikator SA, oczekiwany nr. sekwencyjny, następnego pakietu i dodatkową informację weryfikującą integralność pakietu
 - **tunnel mode** – cały pakiet IP wraz z dodatkowym nagłówkiem obudowywany jest w formę nowego pakietu IP (np. w bramie stanowiącej firmową zaporę sieciową)



- Tunelowanie w celu **agregacji** kilku połączeń TCP do jednego strumienia w kontekście ukrywania informacji n.t. intensywności wymiany informacji między określonymi węzłami (*traffic analysis*)

IPSec – enkapsulacja



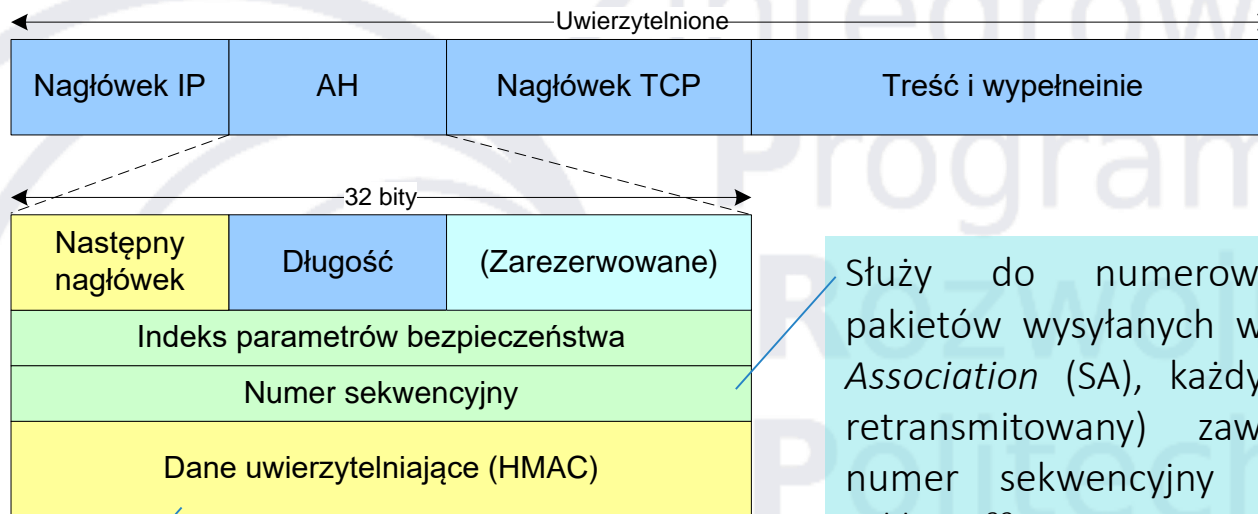
Authenticatin Header – zapewnia kontrolę integralności i zabezpieczenie przed atakiem powtarzającym (nie gwarantuje poufności)

Zawiera wartość przepisaną z pola *Protokół* nagłówka IP (zwykle 6)

Zawiera liczbę 32b słów składających się na nagłówek AH, pomniejszoną o 2

Jest identyfikatorem połączenia, wartość ustalana przez nadawcę w celu wskazania w bazie nadawcy konkretnego rekordu zawierającego rozmaite informacje związane z połączeniem

IPSec – enkapsulacja

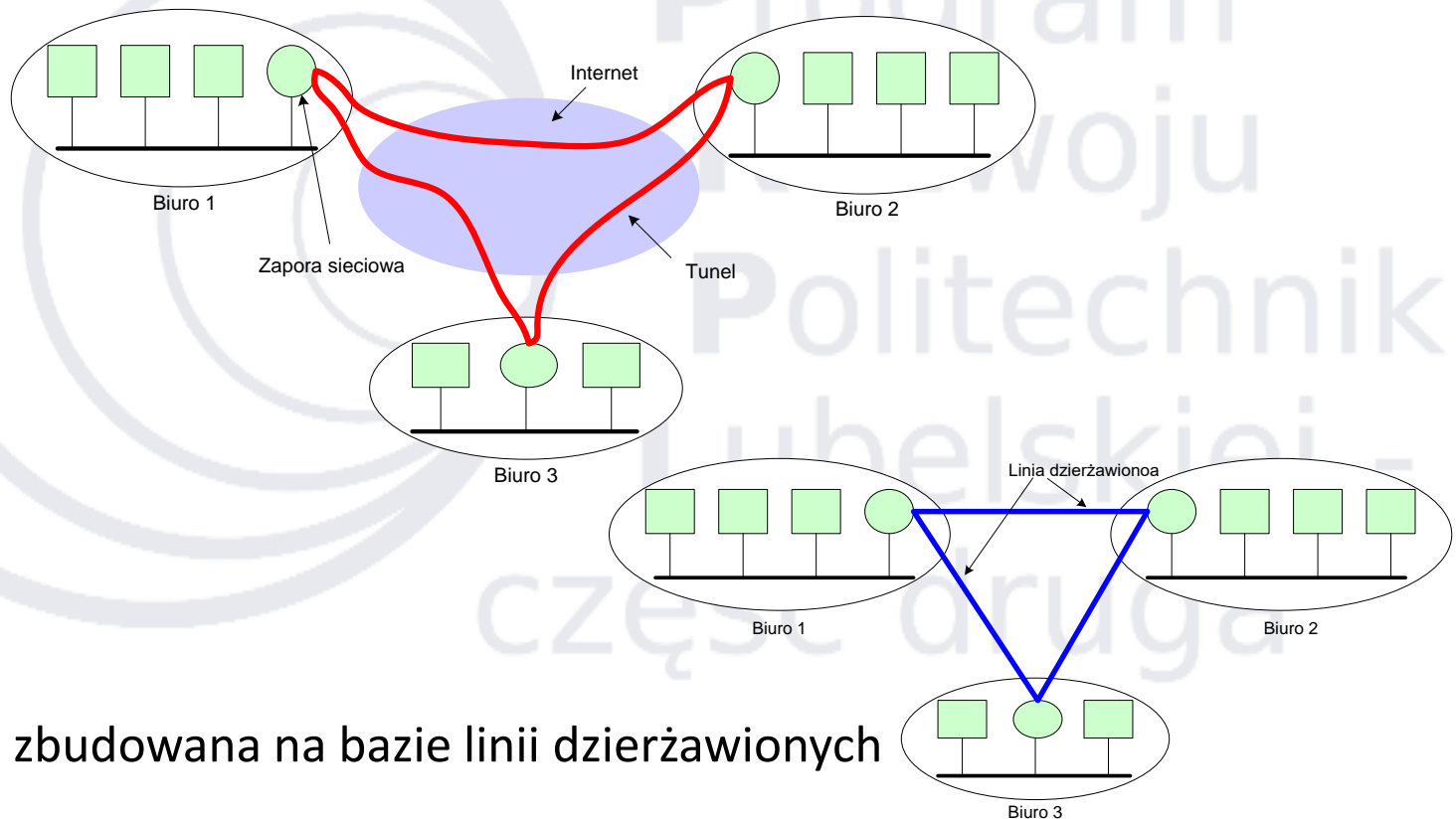


Służy do numerowania wszystkich pakietów wysyłanych w ramach *Security Association* (SA), każdy pakiet (również retransmitowany) zawiera unikalny numer sekwencyjny IPsec, zawinięcie cyklu - 2^{32}

Pole zmiennej długości zawierające cyfrową sygnaturę (podpis) pakietu. IPsec oparty na algorytmach szyfrowanych kluczami symetrycznymi => negocjacja wspólnego klucza wykorzystywanego w procesie obliczania wspólnej sygnatury. Często => obliczenie funkcji haszującej dla całego pakietu i klucza łącznie (klucz nie wpisywany do pakietu!)
Hashed Message Authentication Code

Prywatne sieci wirtualne (VPN)

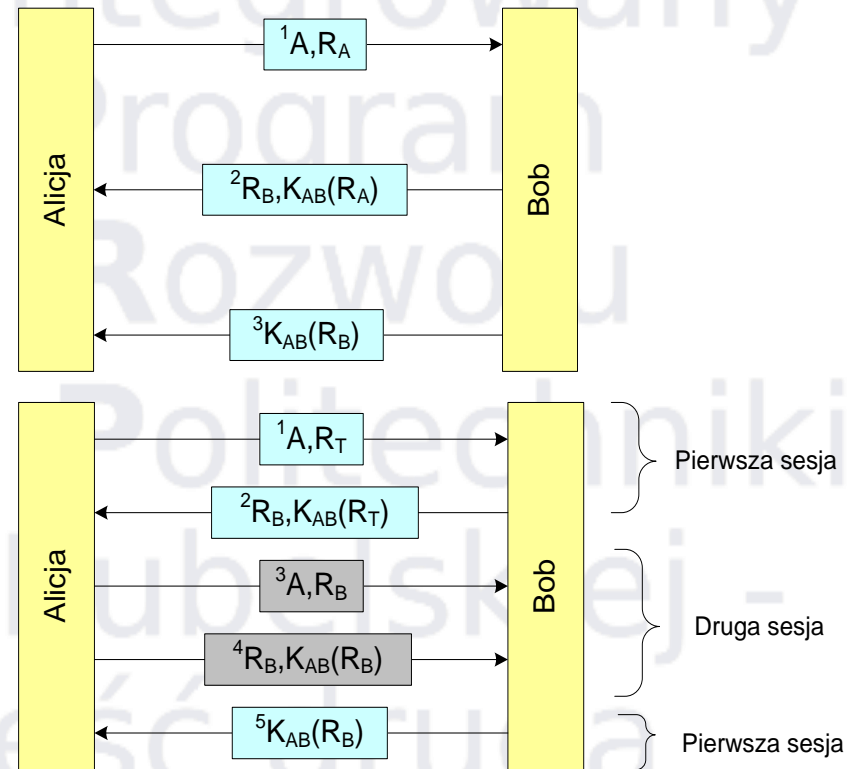
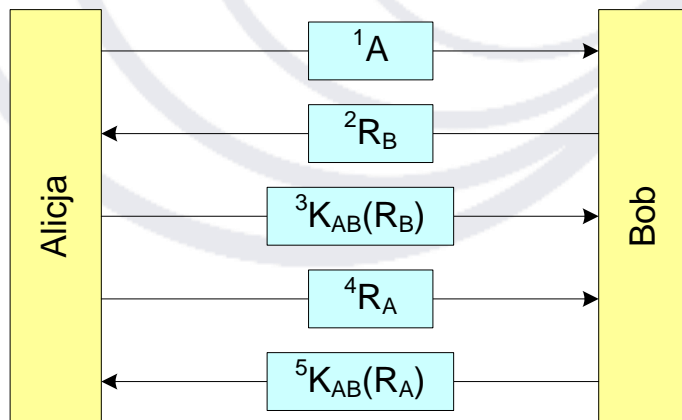
- Prywatna sieć wirtualna



- Sieć zbudowana na bazie linii dzierżawionych

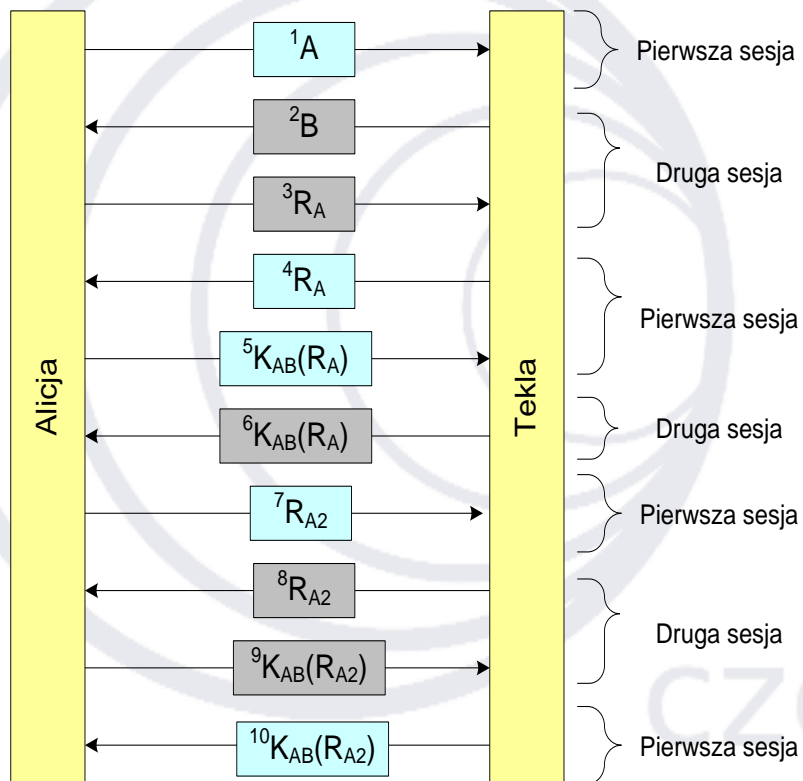
Uwierzytelnianie w oparciu o współdzielony tajny klucz

- A, B – oznaczają tożsamość Alicji i Boba
- R_i – oznacza wyzwanie; losową liczbę wysyłaną partnerowi; indeks identyfikuje stronę wysyłającą (*number used once*)
- K_i – oznacza klucz, a indeks jego wściela
- K_S – oznacza klucz sesji



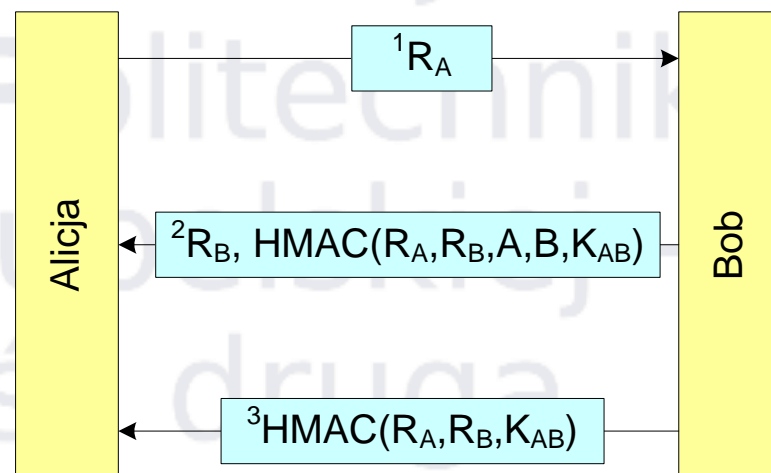
Zmniejszenie ruchu w sieci + przyspieszenie procesu uwierzytelniania, ale również możliwość narażenia na atak lustrzany (*reflection attack*)

Atak lustrzany dla 5 komunikatów i metoda HMAC

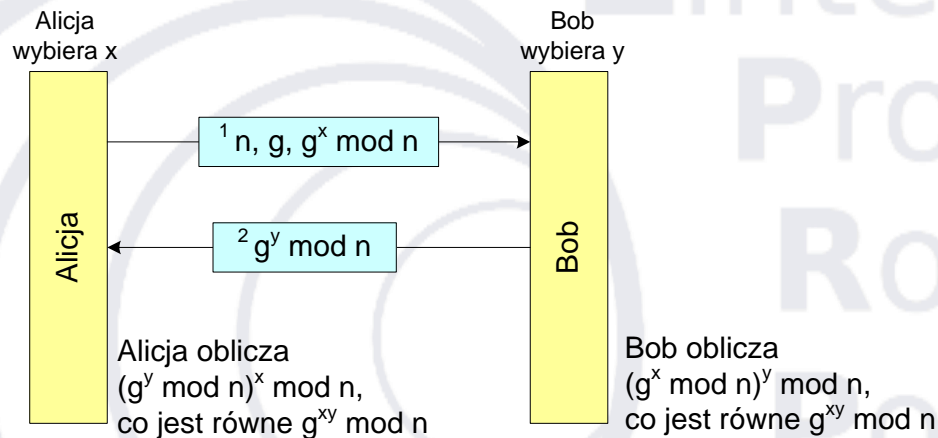


Tekla nie może spowodować żadnej ze stron do szyfrowania żądanych wartości

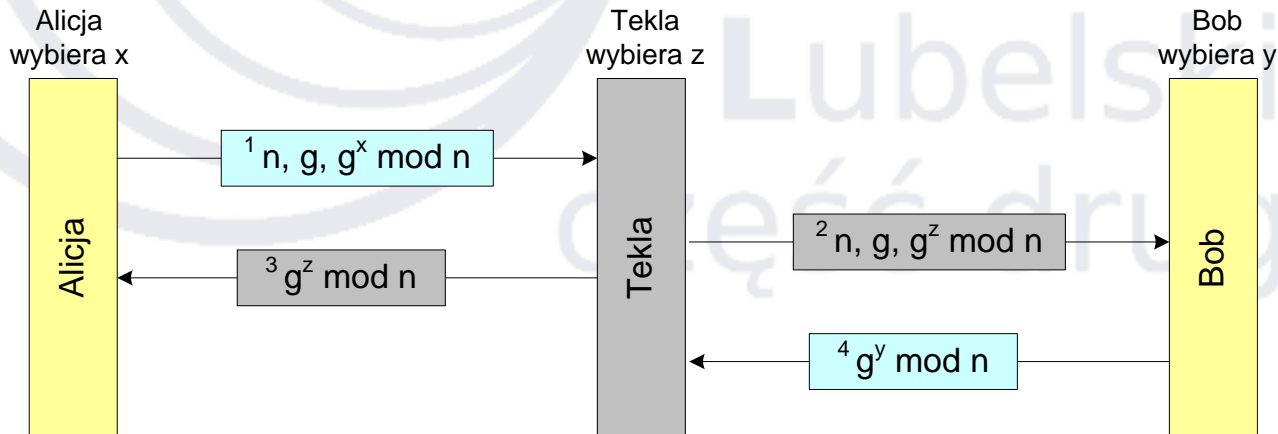
Alternatywą HMAC może być szyfrowanie sekwencji zestawu parametrów w trybie wiązania bloków (zamiast obliczania funkcji haszującej)



Metoda wymiany kluczy Diffiego-Hellmana

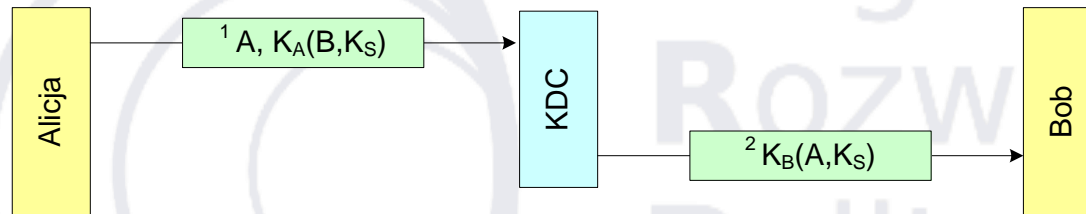


Atak brygady kubełkowej
(*bucket brigade attack* /
man-in-the-middle attack)

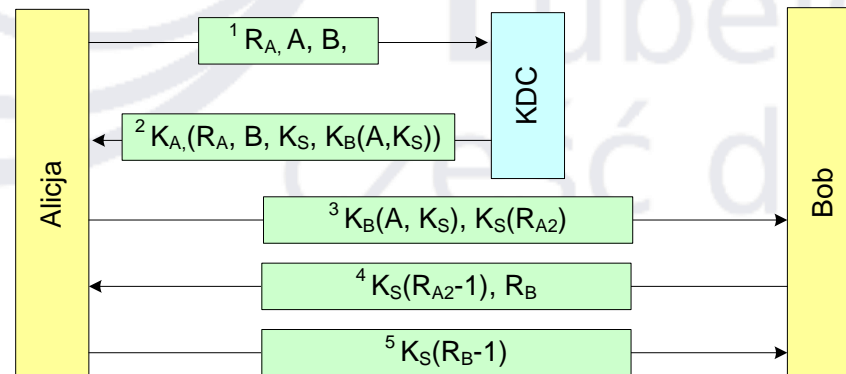


Uwierzytelnianie z udziałem KDC

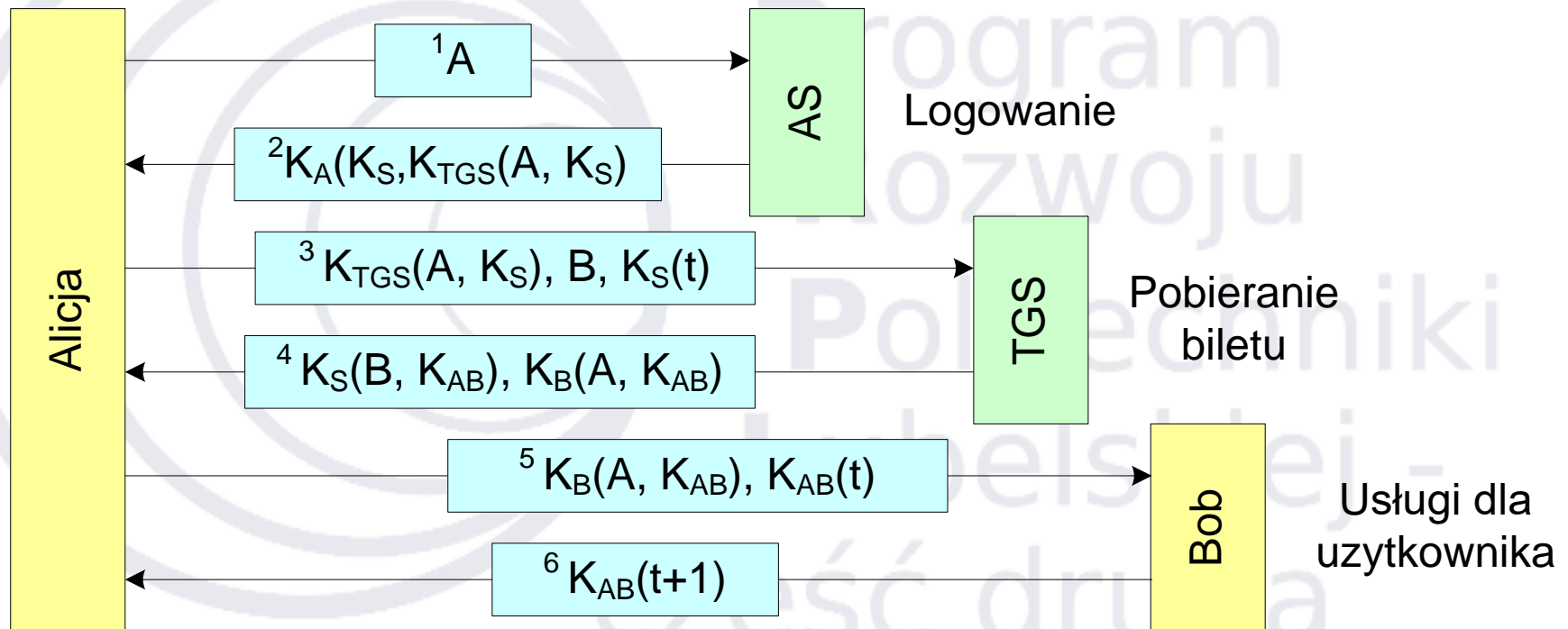
- KDC – (*Key Distribution Center*) centralny dystrybutor kluczy – w najprostszej wersji – mało odporny na atak powtarzający



- Bardziej wyrafinowane podejście, oparte na wielokierunkowym protokole: **protokół Needham-Schroedera**

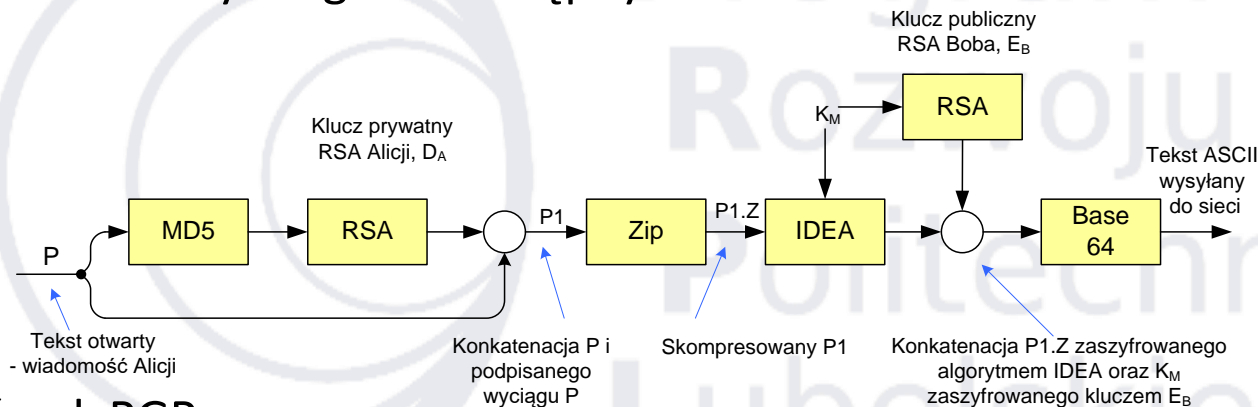


Uwierzytelnianie Kerberos

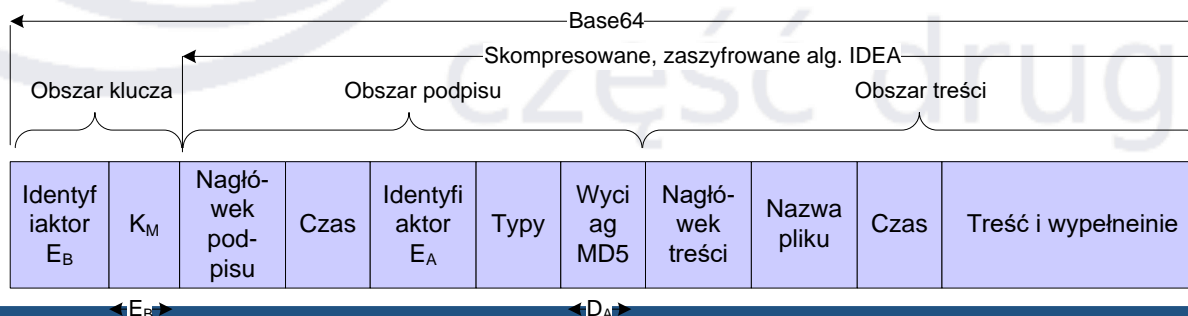


Bezpieczeństwo poczty elektronicznej – PGP

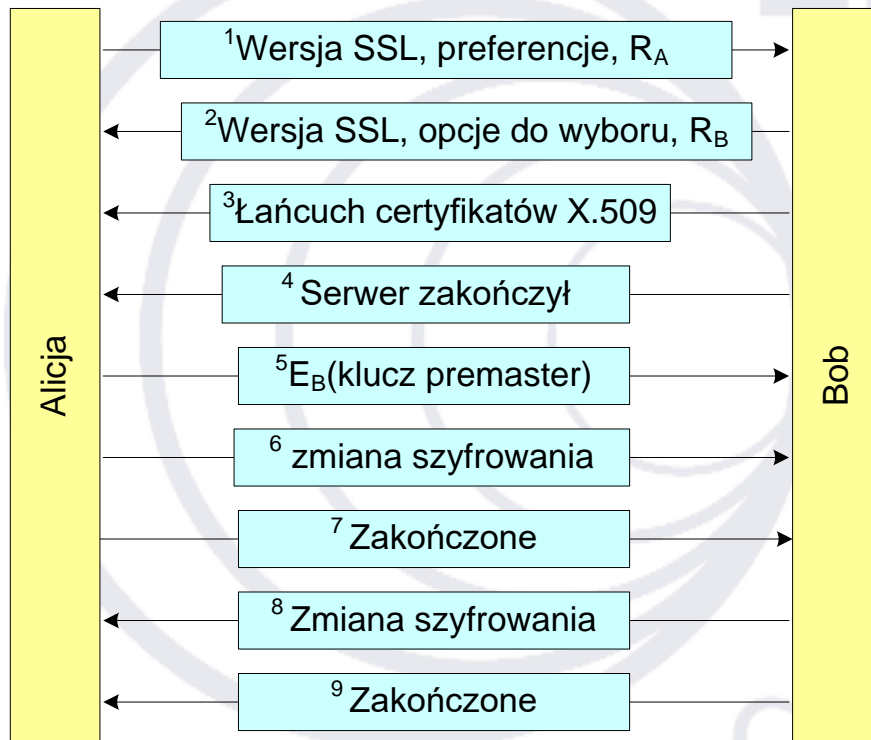
- PGP – *Pretty Good Privacy* (Zimmerman 1991) – pakiet wraz z całym kodem źródłowym ogólnodostępny za darmo w Internecie



- Nagłówek PGP

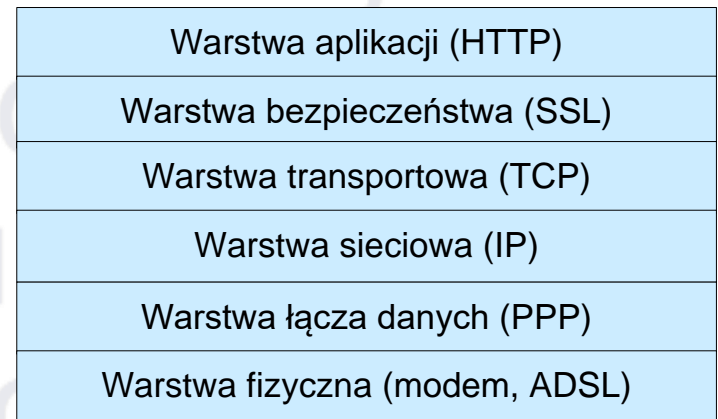


Bezpieczeństwo WWW – SSL



Uproszczony schemat nawiązywania bezpiecznego połączenia w ramach protokołu SSL

Warstwowy model przeglądarki WWW, wykorzystującej SSL



Podstawy Sieci Komputerowych

Kierunki rozwoju lokalnych sieci komputerowych

dr hab. inż. Konrad Gromaszek

Kierunki rozwoju lokalnych sieci komputerowych

- Do przewidywanych trendów rozwoju sieci lokalnych zalicza się:
 - Single Pair Ethernet (SPE)
 - uniform, application and manufacturer-independent continuous IP-based transmission
 - All over IP
 - an integrated approach to networking smart buildings
 - The digital ceiling
 - cellular technology is starting to migrate from smartphones and into the IoT as a viable option for low-power wide-area-network (LPWAN) connectivity
 - LoRaWAN
 - ?

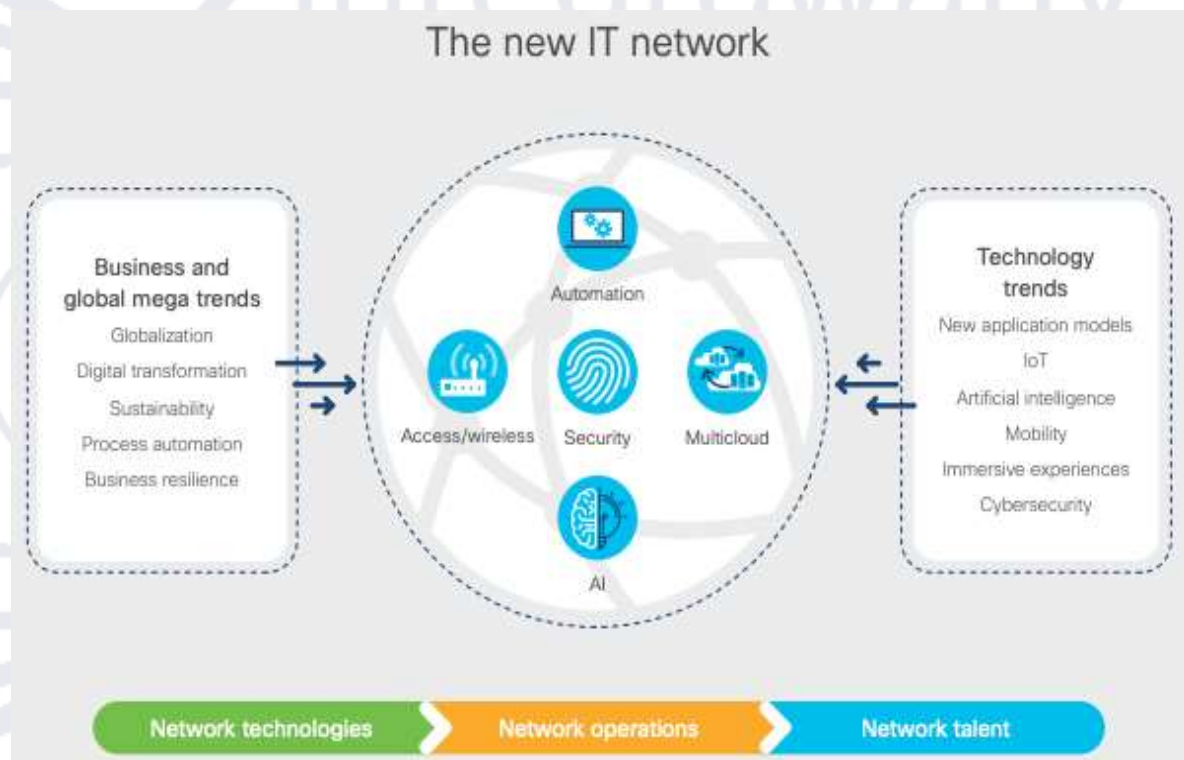
Kierunki rozwoju lokalnych sieci komputerowych

- Cisco - 2020 Global Networking Trends Report

By 2020, leading-edge networking teams will have intent-based networks operating across domains-campus, branch, WAN, data center, cloud, service provider and security. Their networks will be able to comprehend business and application requirements and translate them to network and security policies. Agility will be dramatically improved through the network's intelligent automation and networks will operate with a powerful feedback loop that provides continuous monitoring, assurance, and optimization. The intent-based network will ensure that business services are continuously delivered and protected across the network. These advances will lead to significant benefits for organizations and also for society at large.

- **John Apostolopoulos**, CTO for enterprise networking, Cisco

Kierunki rozwoju sieci komputerowych



https://www.cisco.com/c/dam/m/en_us/solutions/enterprise-networks/networking-report/files/GLBL-ENG_NB-06_0_NA_RPT_PDF_MOFU-no-NetworkingTrendsReport-NB_rpten018612_5.pdf

Materiały zostały opracowane w ramach projektu
„Zintegrowany Program Rozwoju Politechniki Lubelskiej – część druga”,
umowa nr **POWR.03.05.00-00-Z060/18-00**
w ramach Programu Operacyjnego Wiedza Edukacja Rozwój 2014-2020
współfinansowanego ze środków Europejskiego Funduszu Społecznego