



POLITECHNIKA POZNAŃSKA

Poznań University of Technology



Distributed Systems Group

Sieci komputerowe

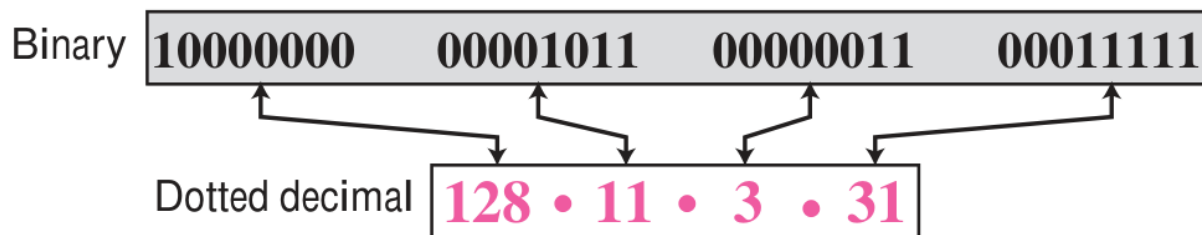
Wykład 5: adresacja IP4

16.05.2024

Wykład w całości przygotowany na
podstawie książki i slajdów: TCP/IP
Protocol suite 4ed, Behrouz A. Forouzan

Adresy IPv4

- Adresy mają rozmiar 32 bity
- Adresy są **unikalne**:
 - Każdy interfejs sieciowy ma unikalny adres IP w ramach całej sieci Internet.
- Adresy są **uniwersalne**:
 - Każdy węzeł chcący używać Internetu musi je rozumieć i umieć się nimi posługiwać
- Przestrzeń adresów:
 - $2^8 = 256$
- Zapis w dwóch notacjach:
 - binarna
 - dziesiętna



Adresacja IPv4 – zadanie 1



Zadanie 1:

Zamień na notację dziesiętną:

a) 10000001 00001011 00001011 11101111

b) 11000001 10000011 00011011 11111111

c) 11100111 11011011 10001011 01101111

d) 11111001 10011011 11111011 00001111

Adresacja IPv4 – zadanie 2



Zadanie 2:

Zamień na notację binarną:

a) 111.56.45.78

b) 221.34.7.82

c) 241.8.56.12

d) 75.45.34.78

Adresacja IPv4 – zadanie 3



Zadanie 3:

Wskaż i wytłumacz błędy w poniższych adresach (o ile występują):

- a) 111.56.045.78
- b) 221.34.7.8.20
- c) 75.45.301.14
- d) 11100010.23.14.67

Adresacja IPv4 – zadanie 4



Zadanie 4:

Znajdź liczbę adresów pomiędzy adresami:

- a) 146.102.29.0, a adresem
- b) 146.102.32.255

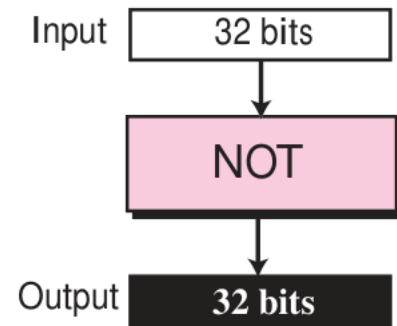
Adresacja IPv4 – zadanie 5



Zadanie 5:

Pierwszy adres w zakresie to 14.11.45.96. Podaj ostatni adres jeśli liczba adresów ma być 32?

Adresacja IPv4 – działania NOT



NOT operation

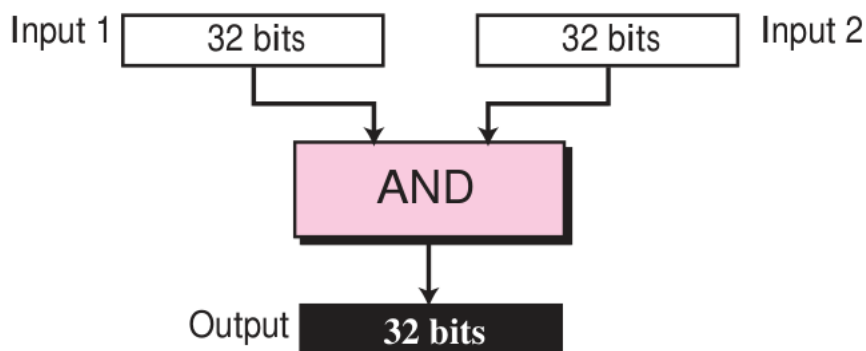
Input	Output
0	1
1	0

Operation for each bit

Original number:	00010001	01111001	00001110	00100011
Complement:	11101110	10000110	11110001	11011100

Original number:	17	.	121	.	14	.	35
Complement:	238	.	134	.	241	.	220

Adresacja IPv4 – działania AND



AND

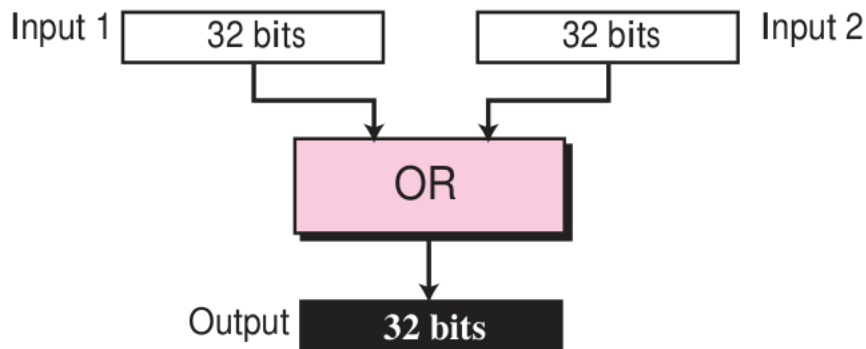
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Operation for each bit

First number:	00010001	01111001	00001110	00100011
Second number:	11111111	11111111	10001100	00000000
Result	00010001	01111001	00001100	00000000

First number:	17	.	121	.	14	.	35
Second number:	255	.	255	.	140	.	0
Result:	17	.	121	.	12	.	0

Adresacja IPv4 – działania OR



OR

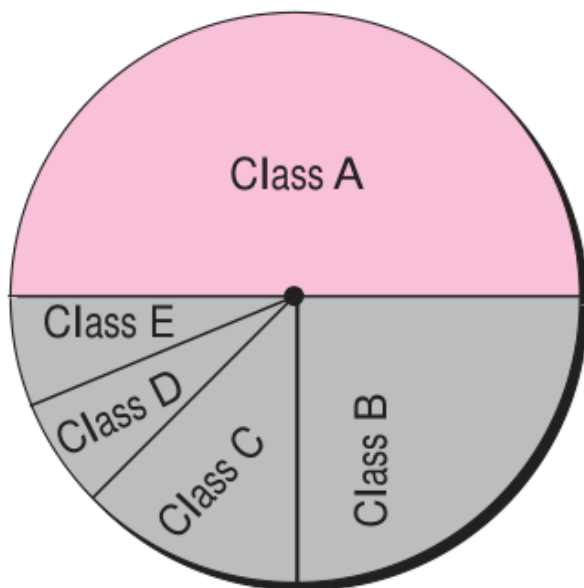
Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Operation for each bit

First number:	00010001	01111001	00001110	00100011
Second number:	11111111	11111111	10001100	00000000
Result	11111111	11111111	10001110	00100011

First number:	17	.	121	.	14	.	35
Second number:	255	.	255	.	140	.	0
Result:	255	.	255	.	142	.	35

Adresacja IPv4 – klasy adresów



Class A: $2^{31} = 2,147,483,648$ addresses, 50%

Class B: $2^{30} = 1,073,741,824$ addresses, 25%

Class C: $2^{29} = 536,870,912$ addresses, 12.5%

Class D: $2^{28} = 268,435,456$ addresses, 6.25%

Class E: $2^{28} = 268,435,456$ addresses, 6.25%

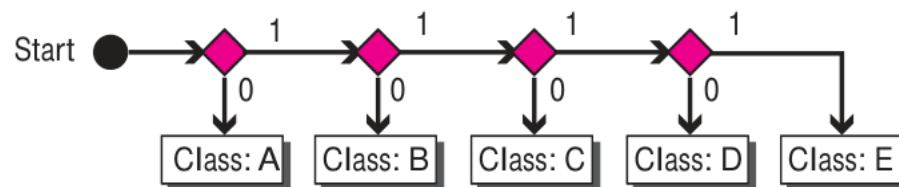
Adresacja IPv4 – klasy adresów

	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0.....			
Class B	10.....			
Class C	110.....			
Class D	1110....			
Class E	1111....			

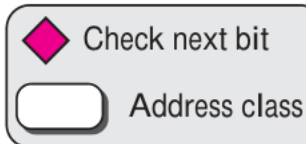
Binary notation

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–255			
Class E	240–255			

Dotted-decimal notation



Legend



Adresacja IPv4 – zadanie 6



Zadanie 6:

Podaj klasę, każdego z następujących adresów:

- a) 00000001 00001011 00001011 11101111
- b) 11000001 10000011 00011011 11111111
- c) 10100111 11011011 10001011 01101111
- d) 11110011 10011011 11111011 00001111

Adresacja IPv4 – zadanie 7

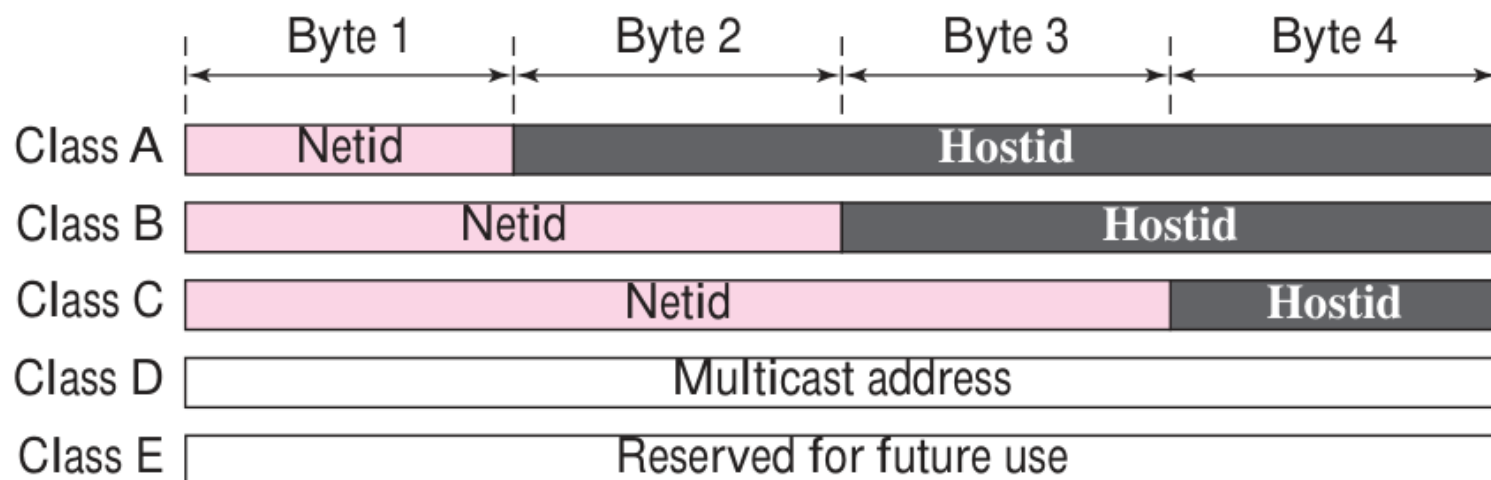


Zadanie 7:

Podaj klasę, każdego z następujących adresów:

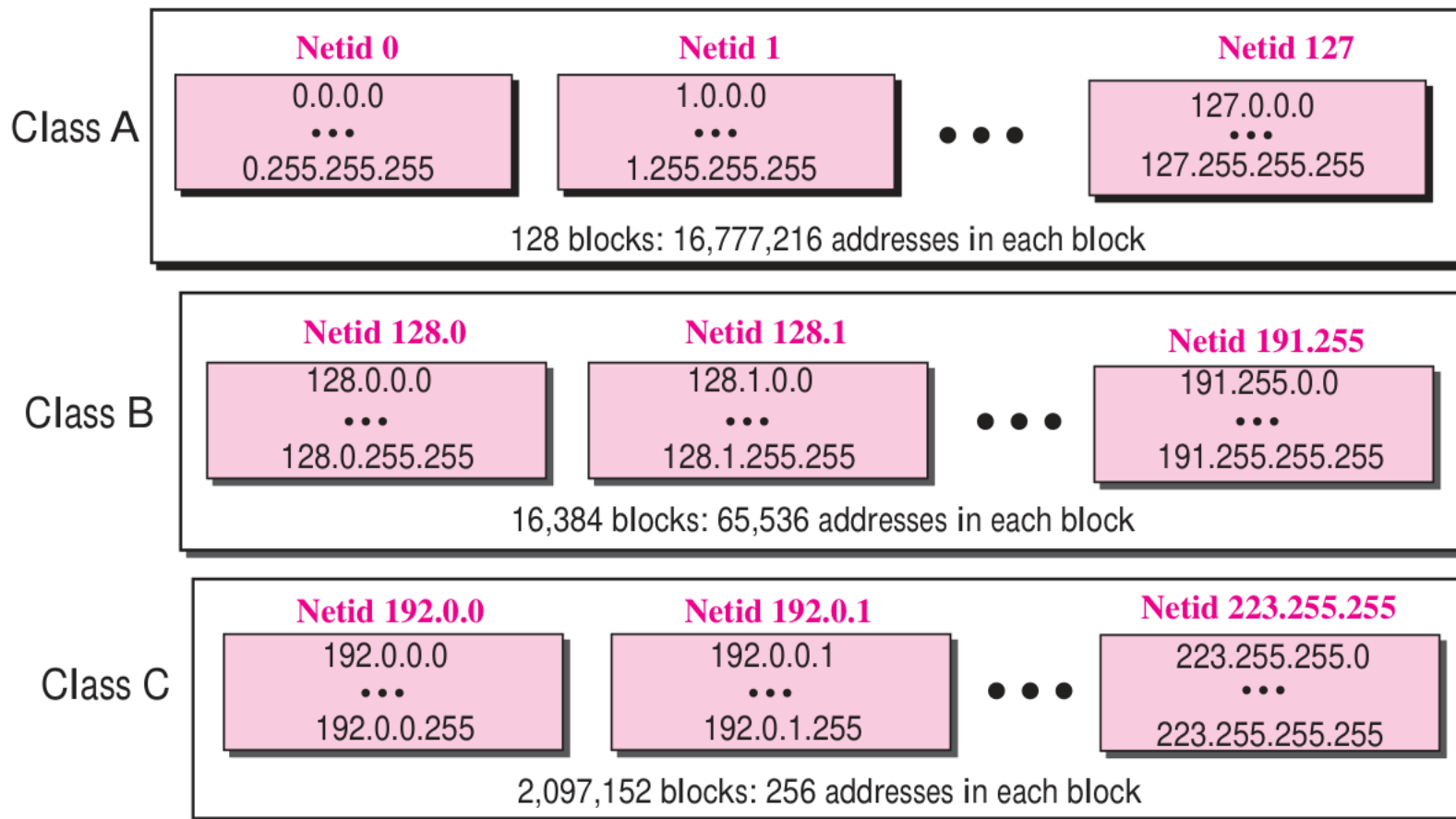
- a) 227.12.14.87
- b) 193.14.56.22
- c) 14.23.120.8
- d) 252.5.15.111

Adresacja IPv4 – netID and hostID



The network address is the identifier of a network.

Adresacja IPv4 – netID and hostID



Adresacja IPv4 – netID and hostID



Class D

224.0.0.0 ... 239.255.255.255

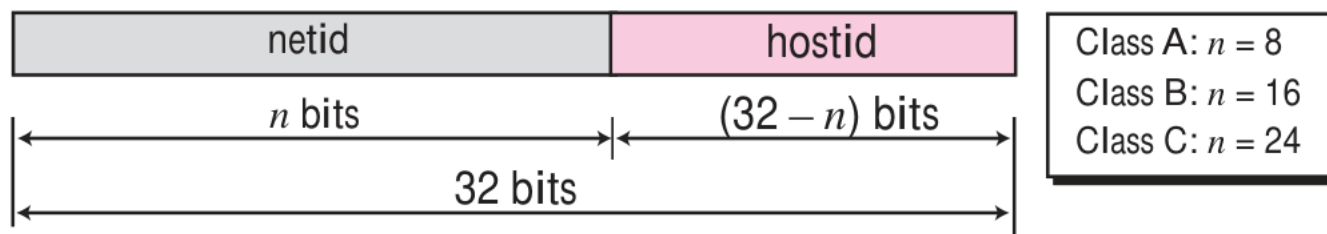
One block: 268,435,456 addresses

Class E

240.0.0.0 ... 255.255.255.255

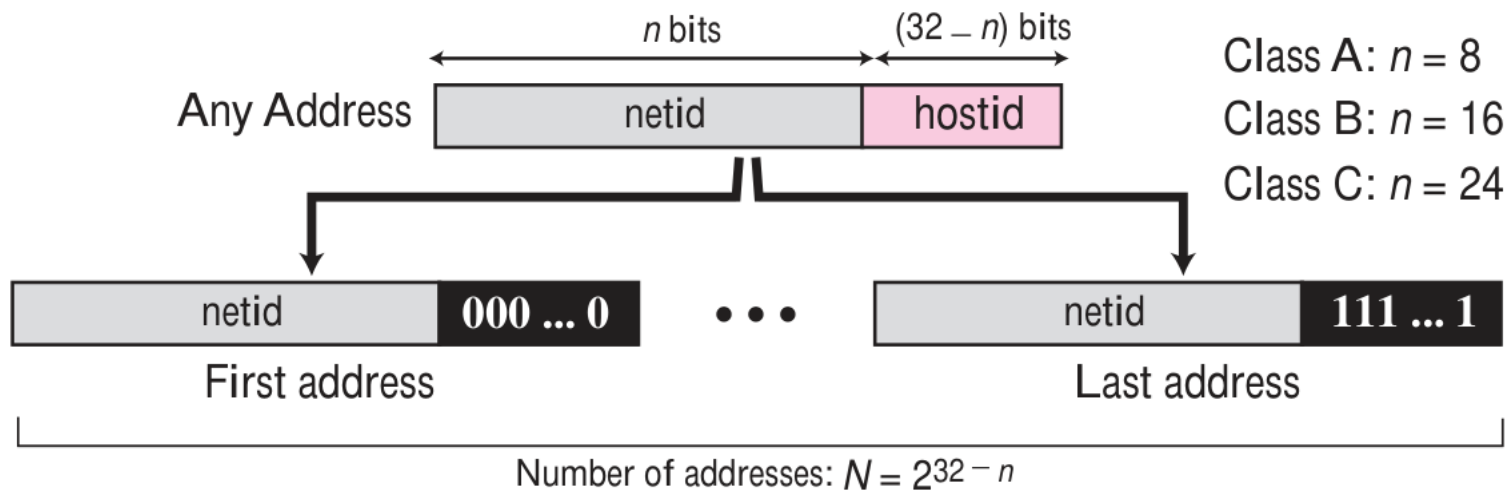
One block: 268,435,456 addresses

Adresacja IPv4 – dwupoziomowa adresacja



Jak w telefonii: +48 (61) 843 64 92

Adresacja IPv4 – 2-lvl addr. Zakres adresów



Adresacja IPv4 – zadanie 8



Zadanie 8:

Mając adres 73.22.17.25 (w pewnym zakresie), i zakładając podejście klasowe znajdź pierwszy i ostatni adres dla tego zakresu (bloku).

Zadanie 9:

j.w. ale dla adresu 180.8.17.9

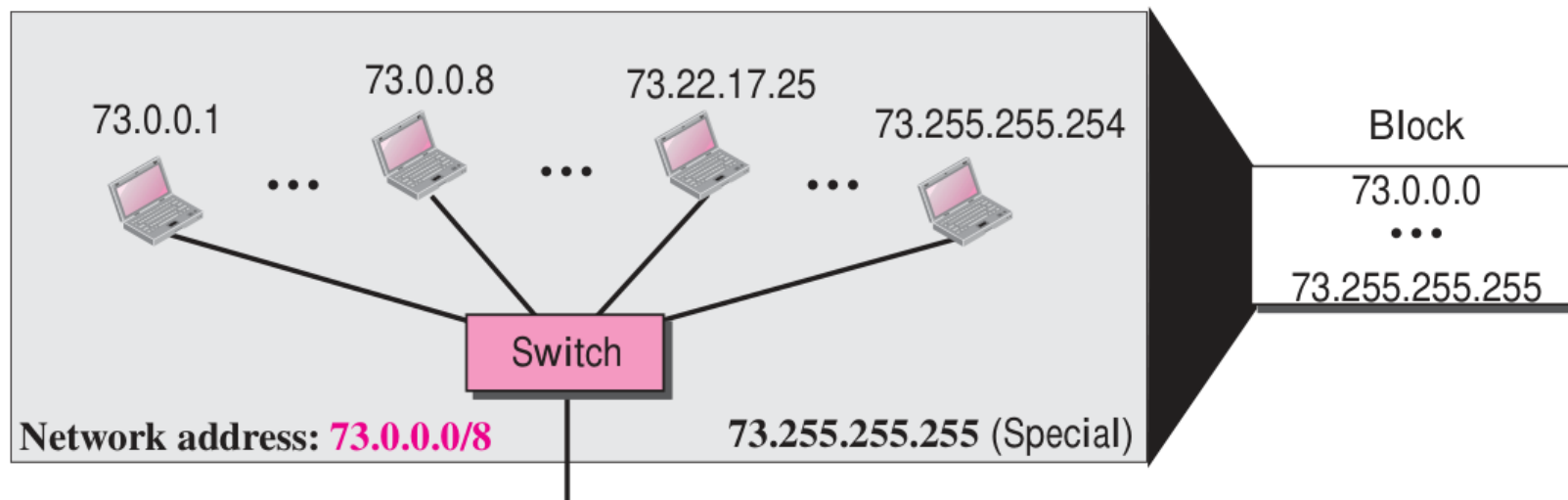
Zadanie 10:

j.w. ale dla adresu 200.11.8.45

Adresacja IPv4 – zadanie 8



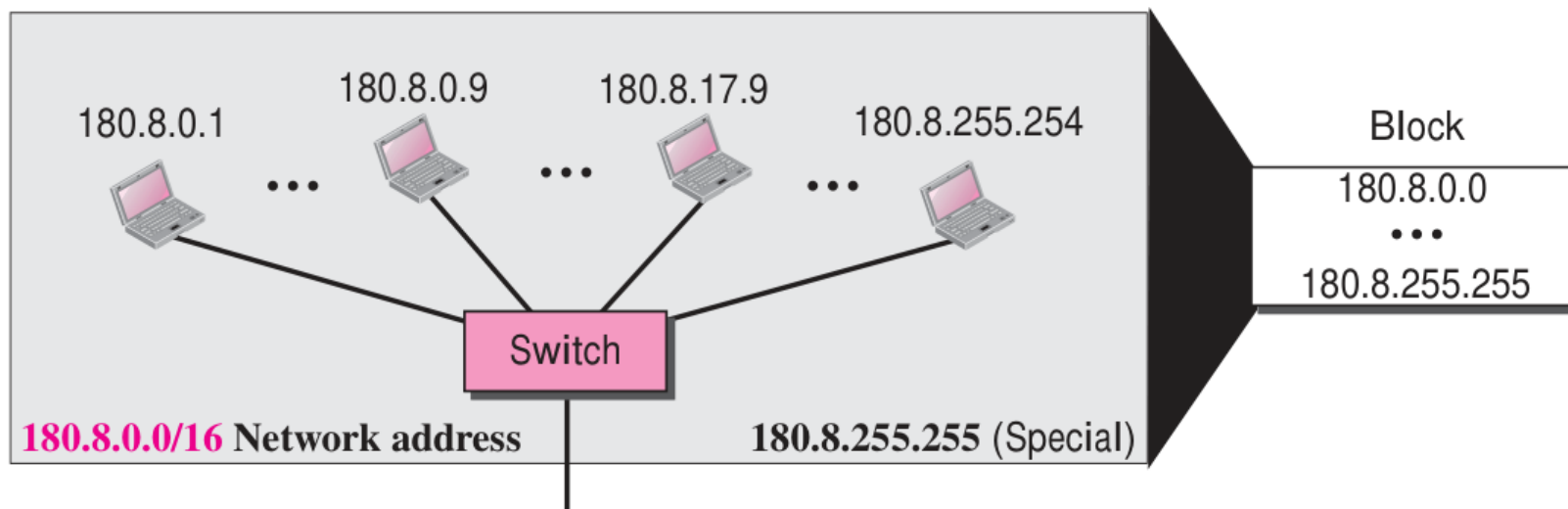
Netid 73: common in all addresses



Adresacja IPv4 – zadanie 9



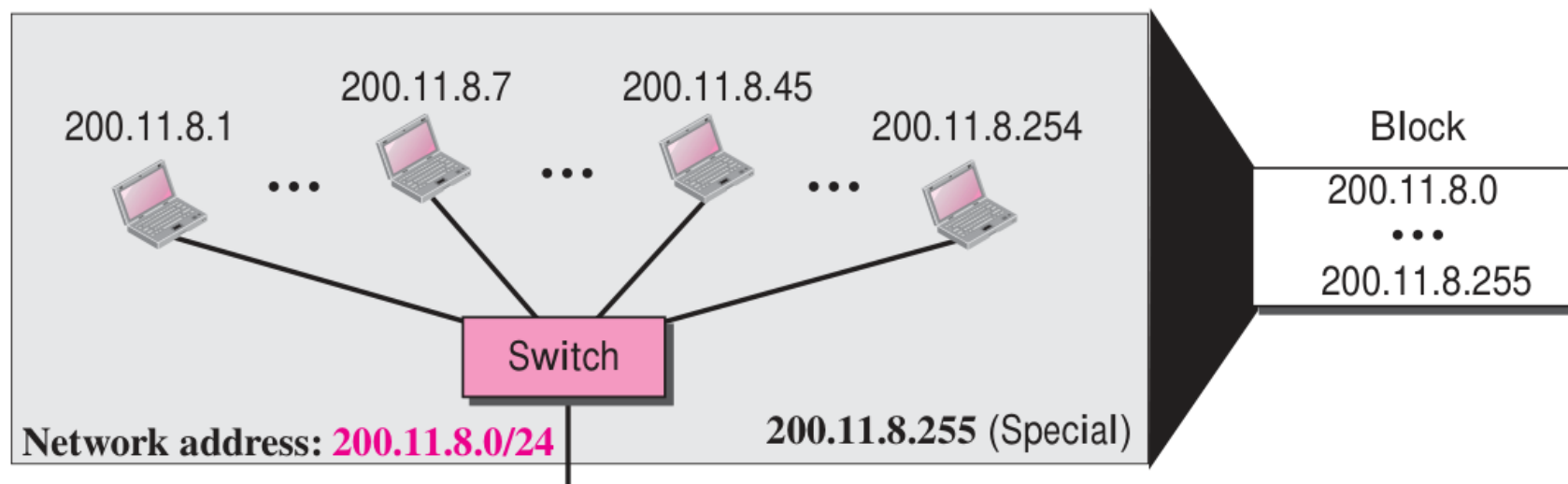
Netid 180.8: common in all addresses



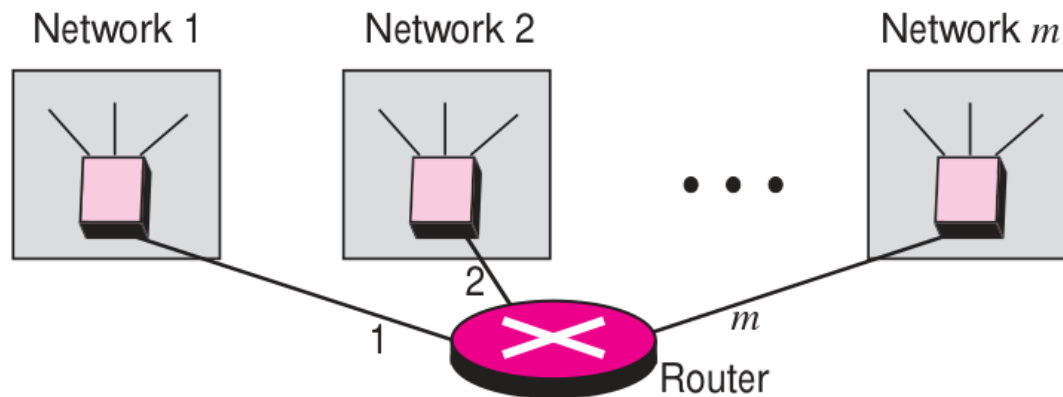
Adresacja IPv4 – zadanie 10



Netid 200.11.8: common in all addresses



Adresacja IPv4 – routing i klasy adresów



Routing Process

Destination address

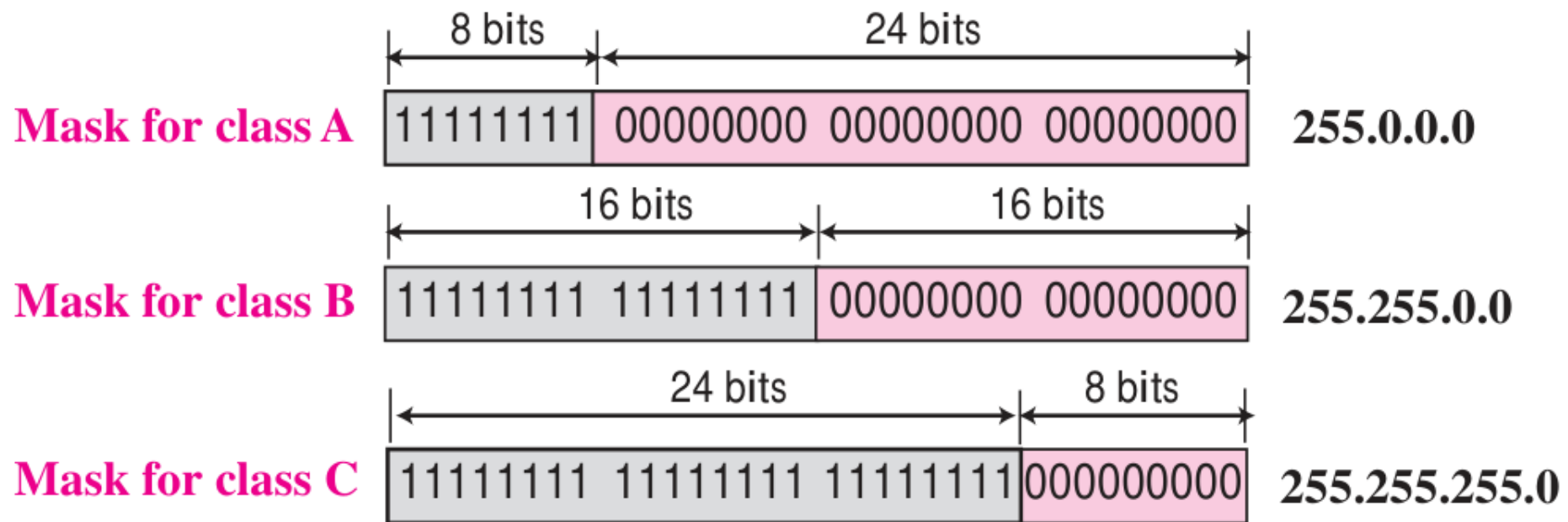
Find Network address

Routing Table

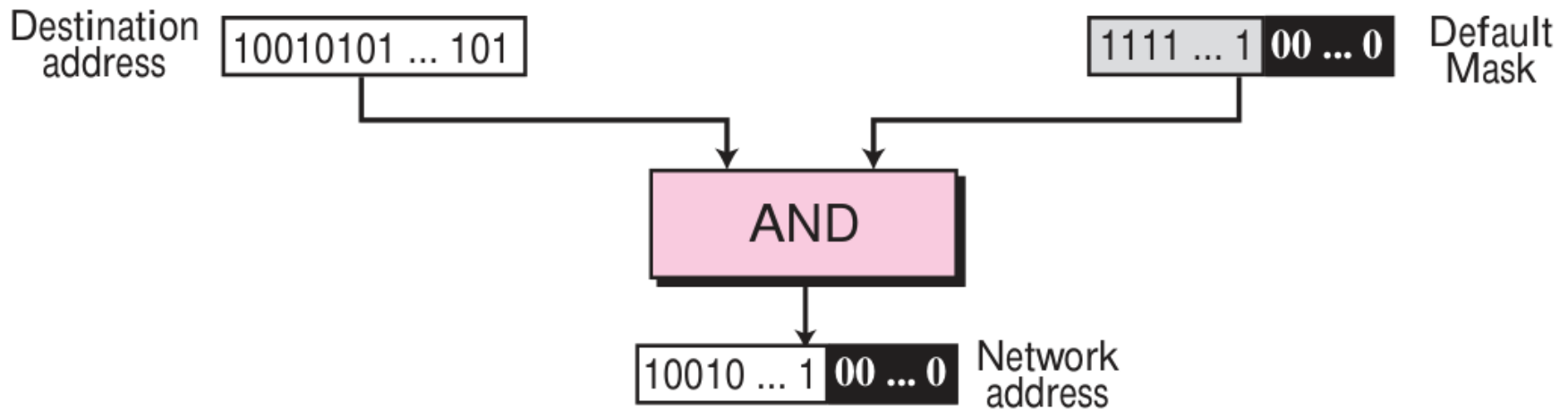
Network address	Interface
$b_1 \cdot c_1 \cdot d_1 \cdot e_1$	1
$b_2 \cdot c_2 \cdot d_2 \cdot e_2$	2
...	...
$b_m \cdot c_m \cdot d_m \cdot e_m$	m

Interface number

Adresacja IPv4 – maska sieci i klasy adresów



Adresacja IPv4 – maska sieci i klasy adresów



Adresacja IPv4 – zadanie 11



Zadanie 11:

Do routera przychodzi pakiet o adresie docelowym: 201.24.67.32. Jak router znajduje adres docelowej sieci dla tego pakietu?

Adresacja IPv4 – zadanie 11



Destination address

→

201

.

24

.

67

.

32

Default mask

→

255

.

255

.

0

.

0

Network address

→

201

.

24

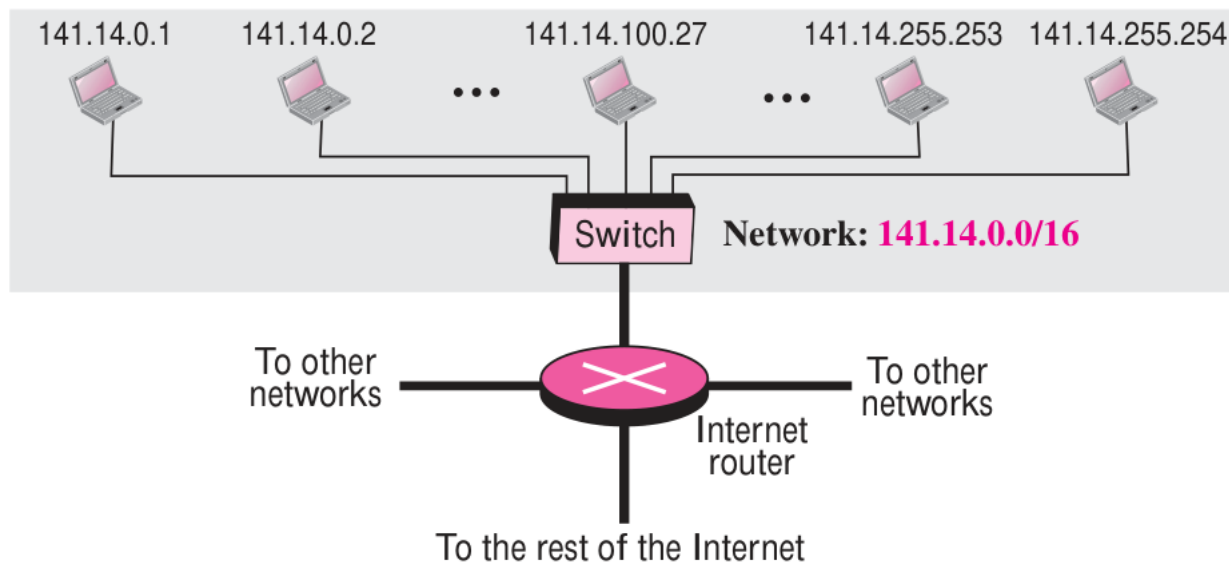
.

0

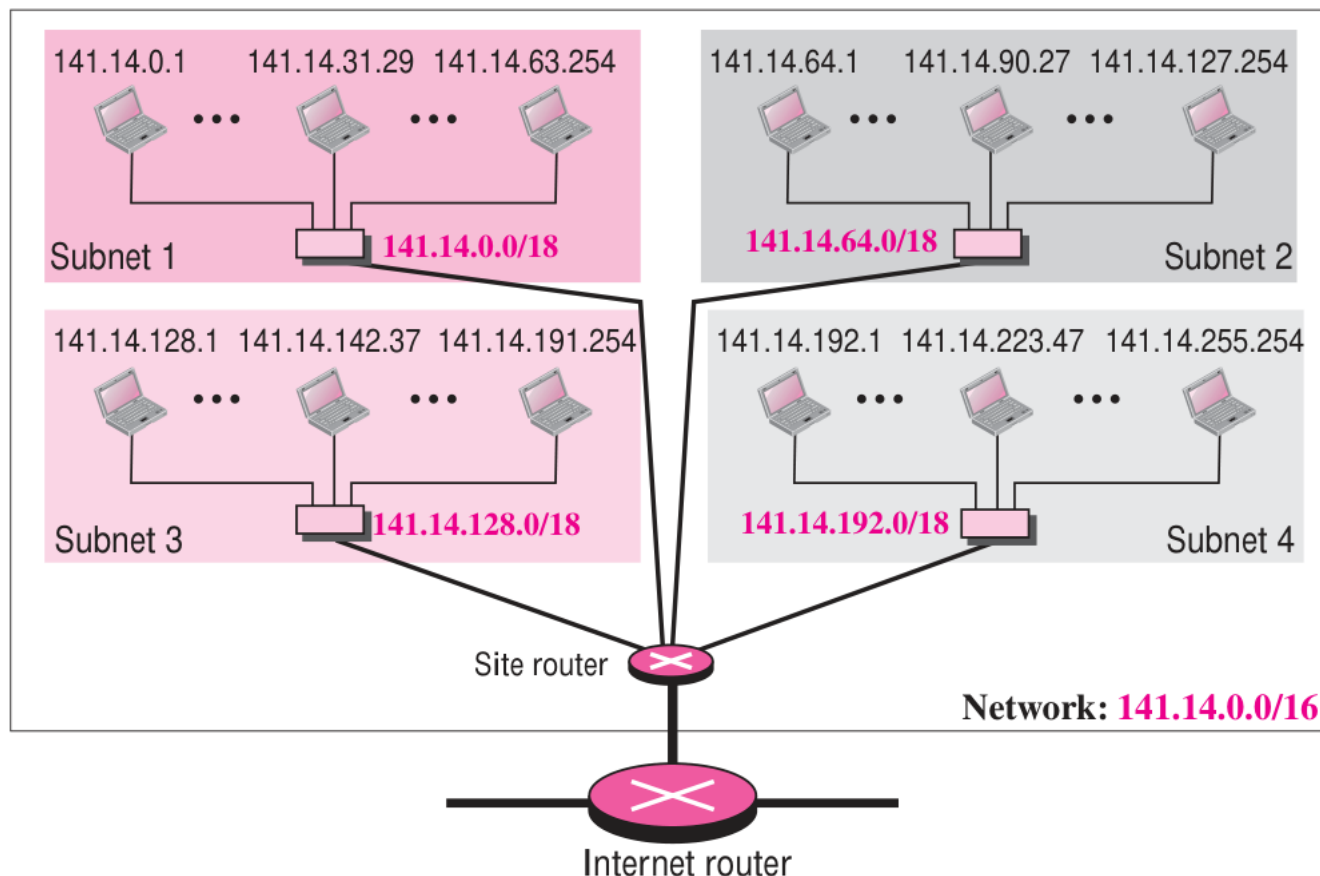
.

0

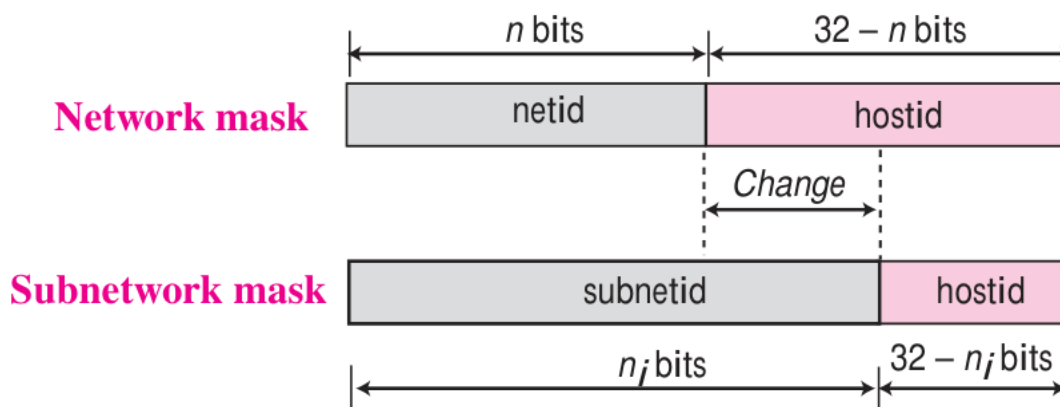
Adresacja IPv4 – podział sieci na podsieci (klasy)



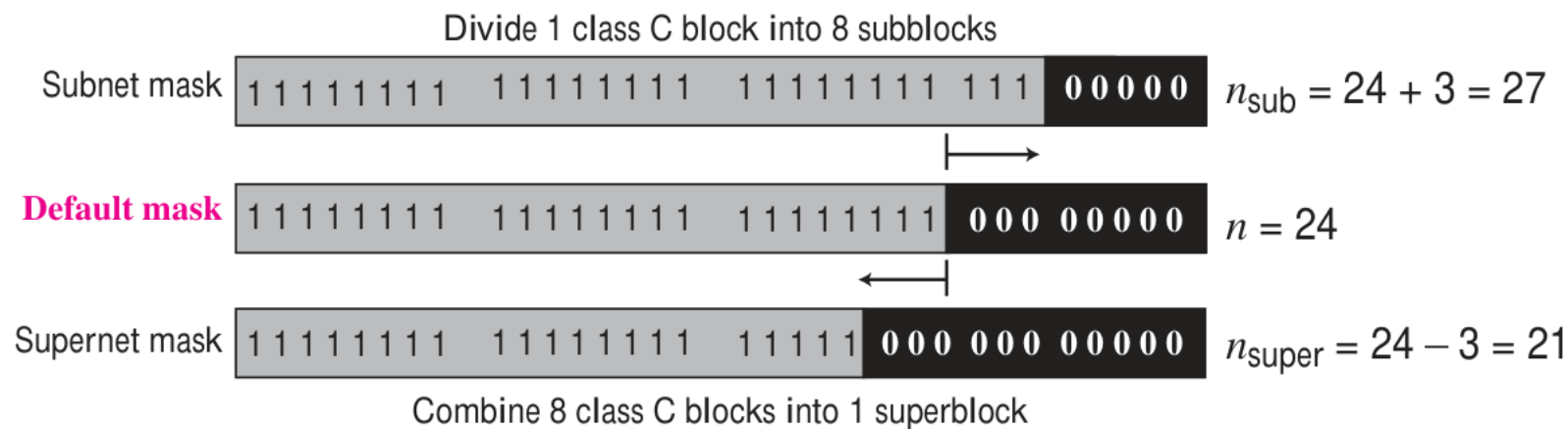
Adresacja IPv4 – podział sieci na podsieci (klasy)



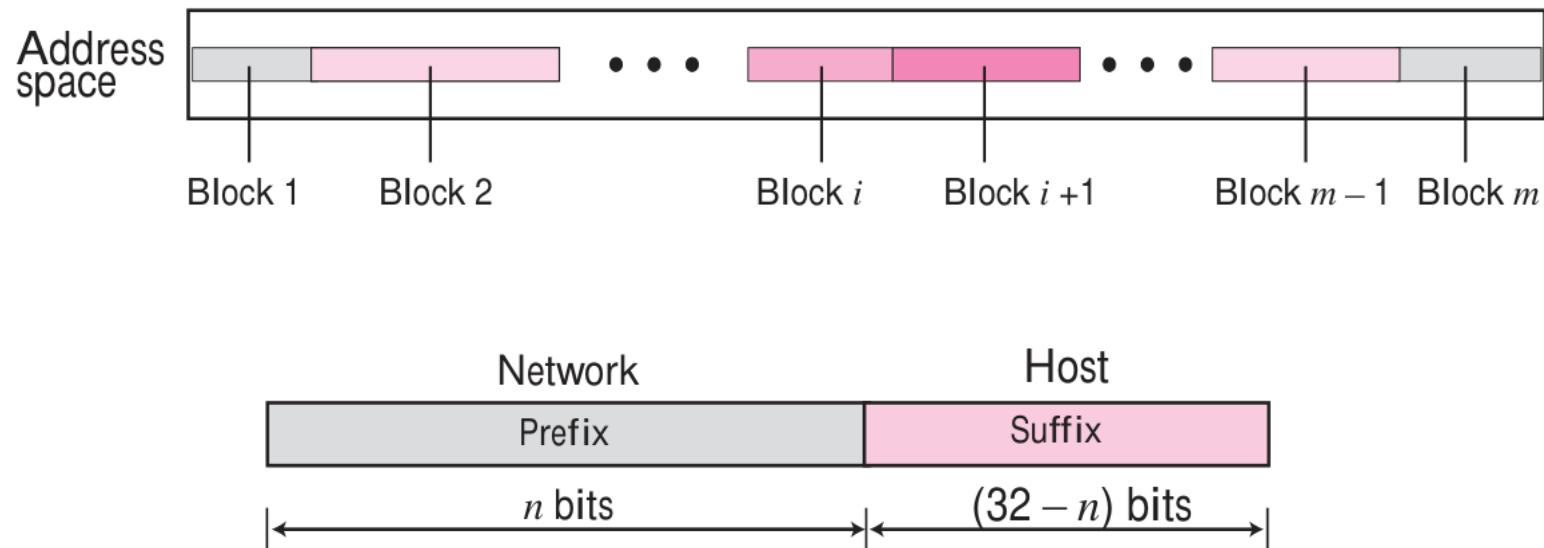
Adresacja IPv4 – podział sieci na podsieci (klasy)



Adresacja IPv4 – podsieć, nadsieć

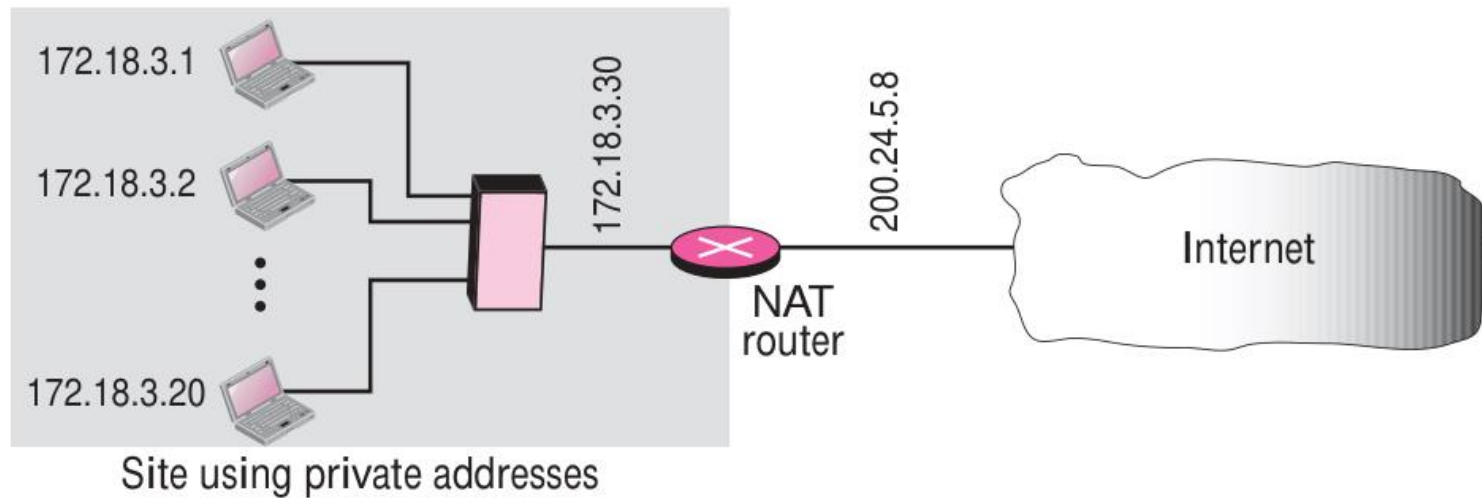


Adresacja IPv4 – bezklasowa zmienna długość bloków

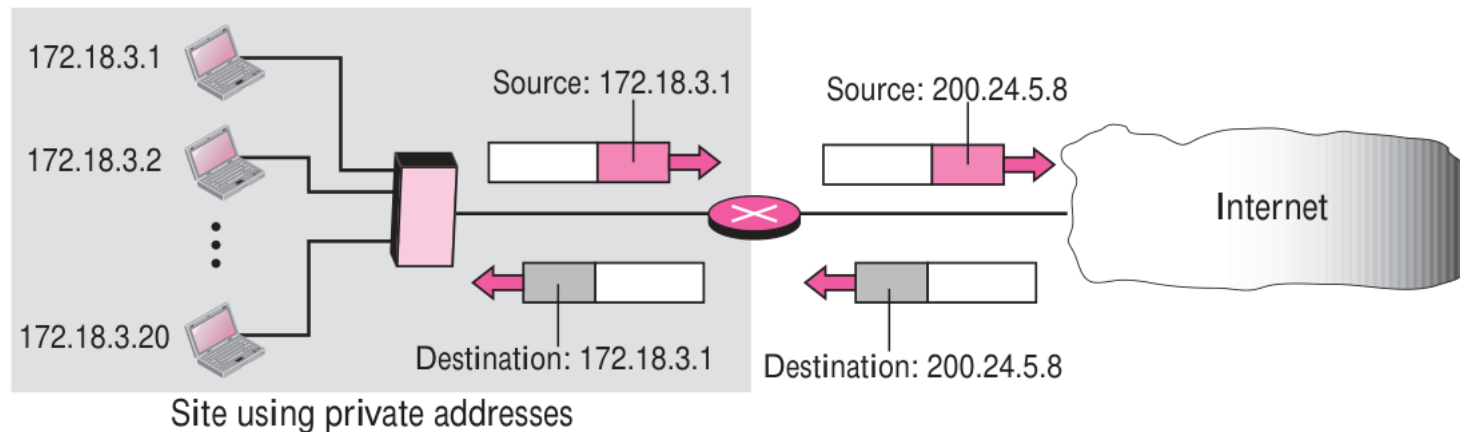


Długość prefixu w podejściu bezklasowym może być w zakresie od 1 do 32.

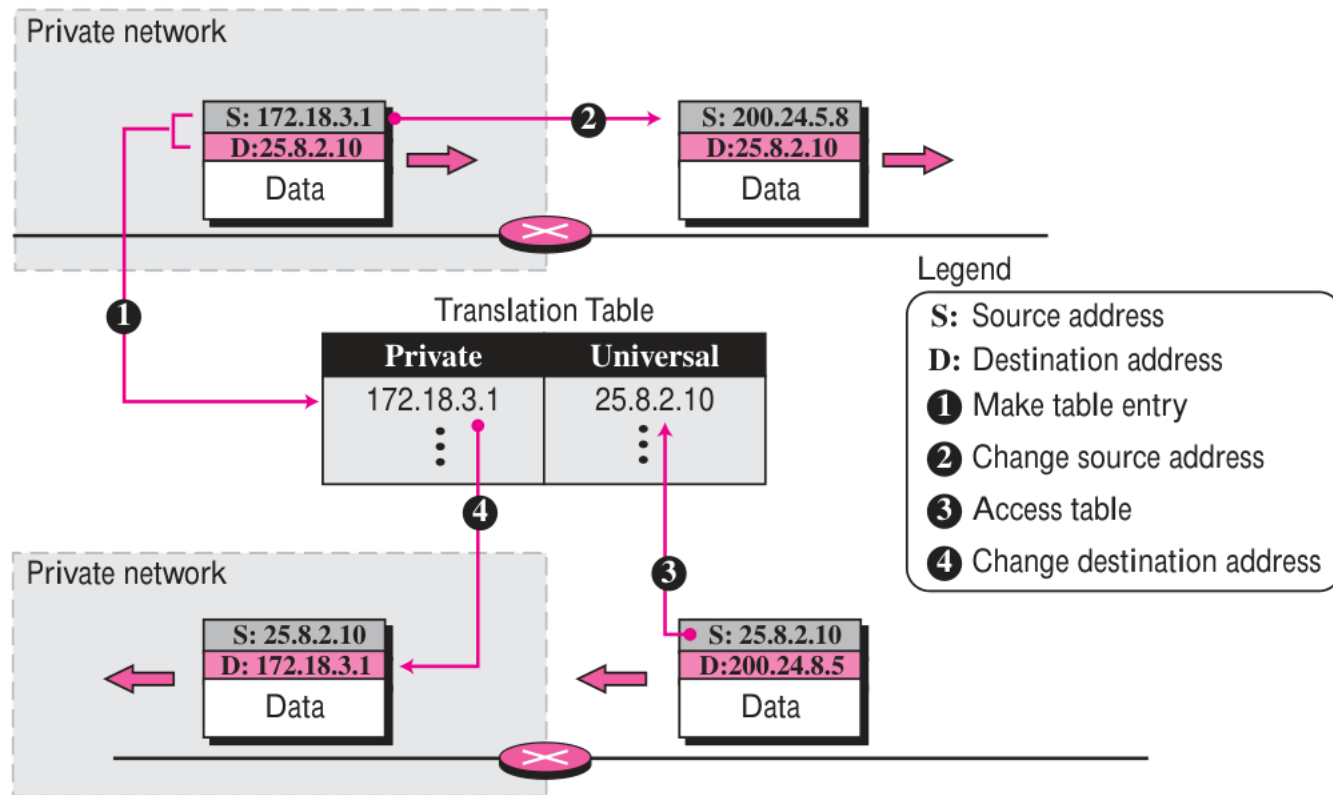
Adresacja IPv4 – NAT



Adresacja IPv4 – translacja adresów NAT/DNAT

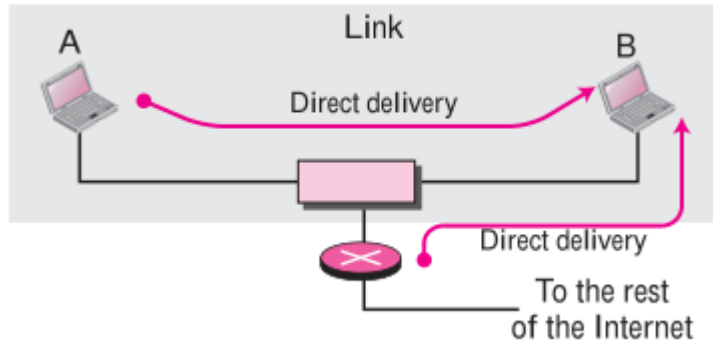


Adresacja IPv4 – tablica translacji NAT/DNAT

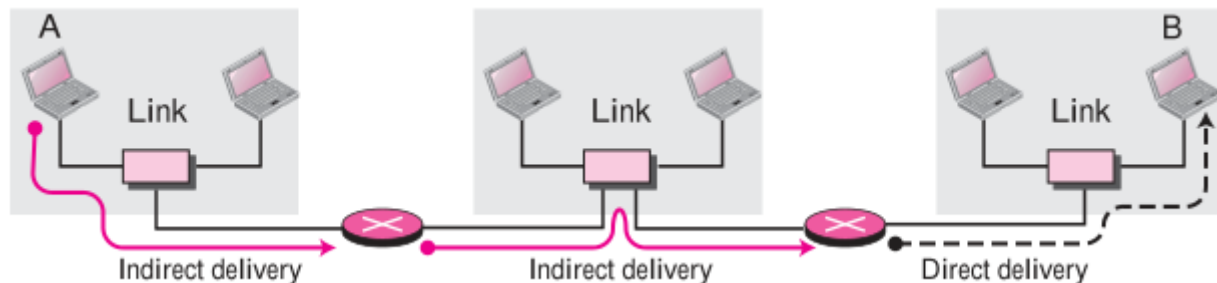


- **Dostarczanie pakietów w sieci IP (do warstwy):**

- **bezpośrednie:**



- **pośrednie:**



IP – przekazywanie (ang. forwarding)



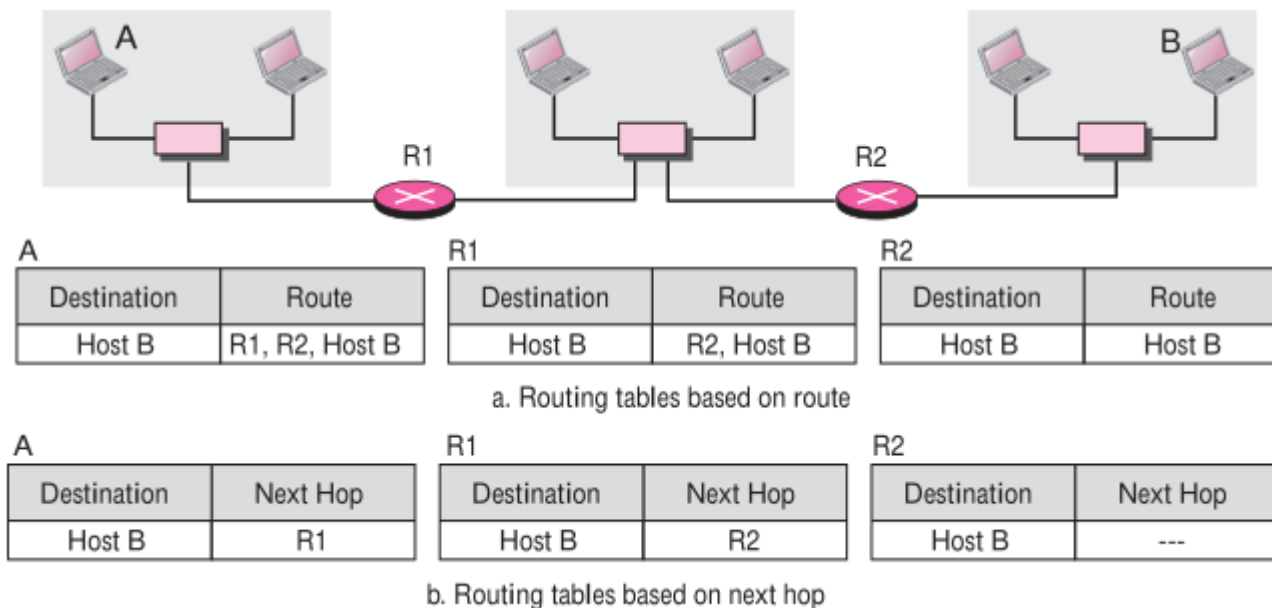
przekazywanie (ang. forwarding)

Przekazywanie pakietów w sieci IP pomiędzy węzłami pośredniczącymi (przekazywanie do sąsiada oznacza przesłanie i dostarczenie pakietu do „next-hop”).

- **usługa połączeniowa** -> przekazywanie na podstawie **adresu docelowego** w pakiecie
- **usługa bezpołączeniowa** -> przekazywanie na podstawie **etykiety**

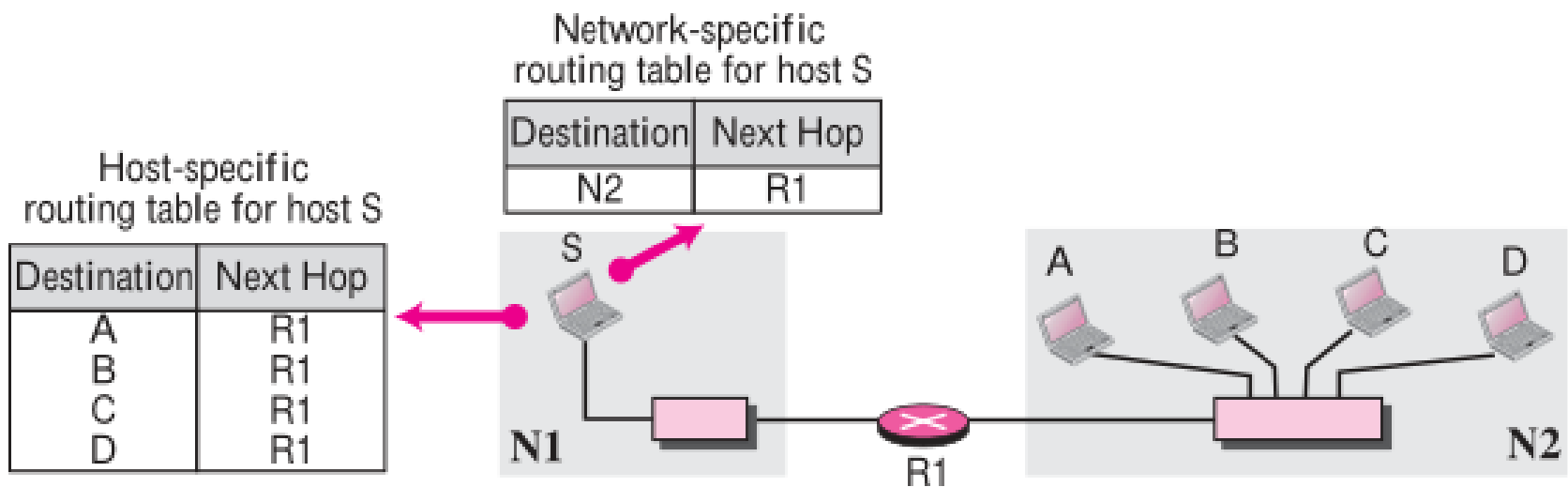
Forwarding – na bazie adresu docelowego

- informacja o sąsiedzie (ang. next-hop)



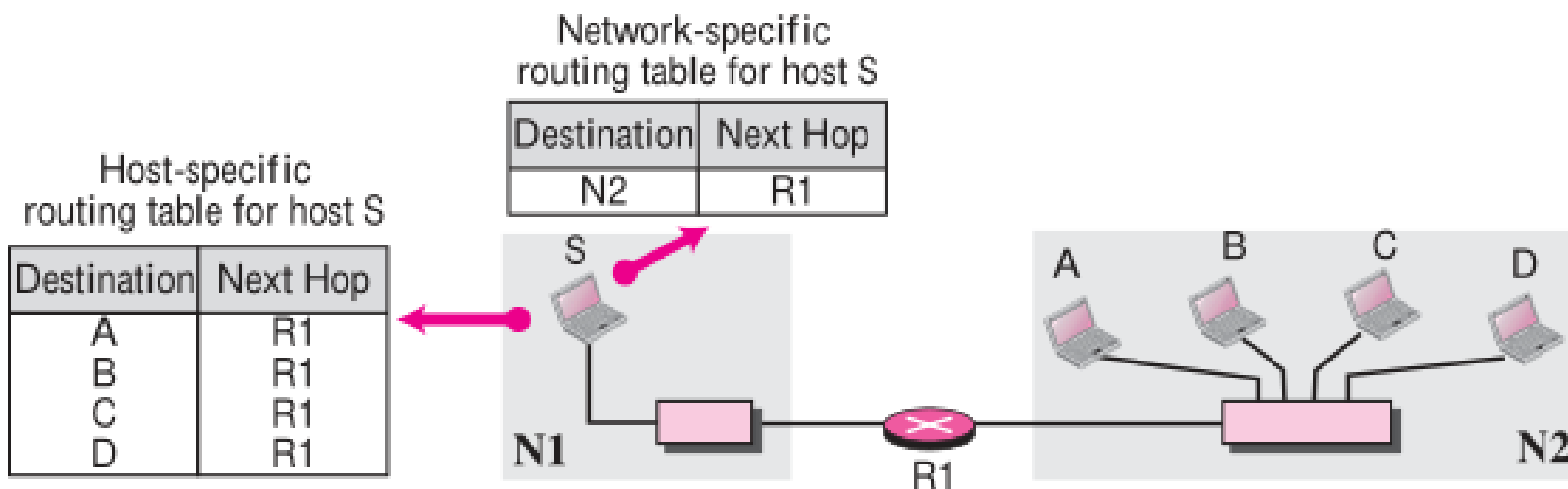
Forwarding – na bazie adresu docelowego

- u oparciu o adres sieci (ang. network-specific)



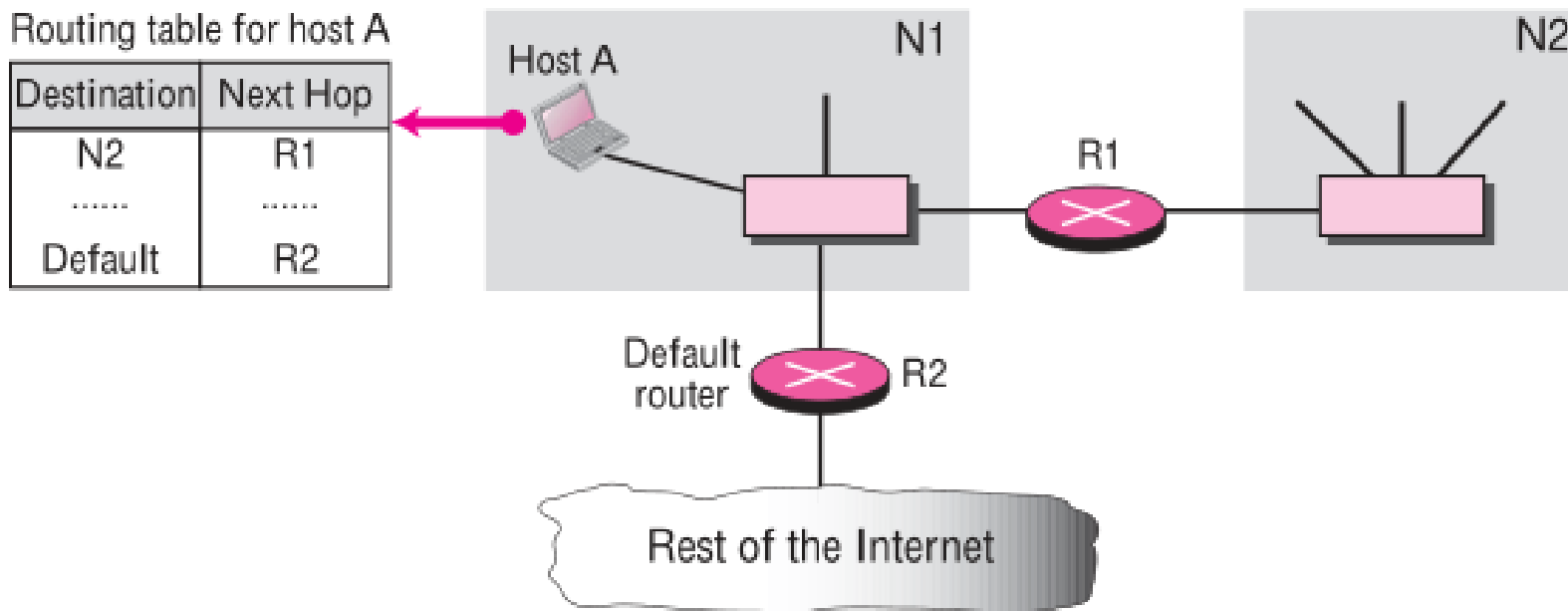
Forwarding – na bazie adresu docelowego

- u oparciu o adres sieci (ang. network-specific)

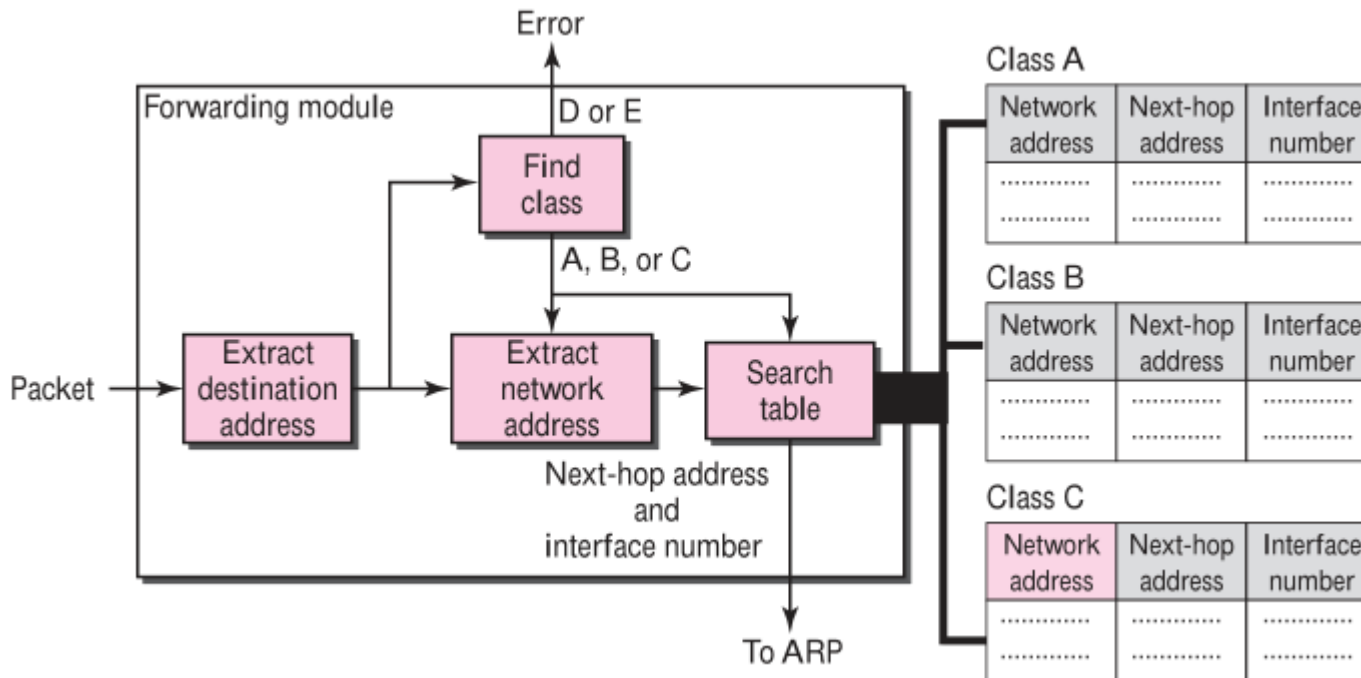


Forwarding – na bazie adresu docelowego

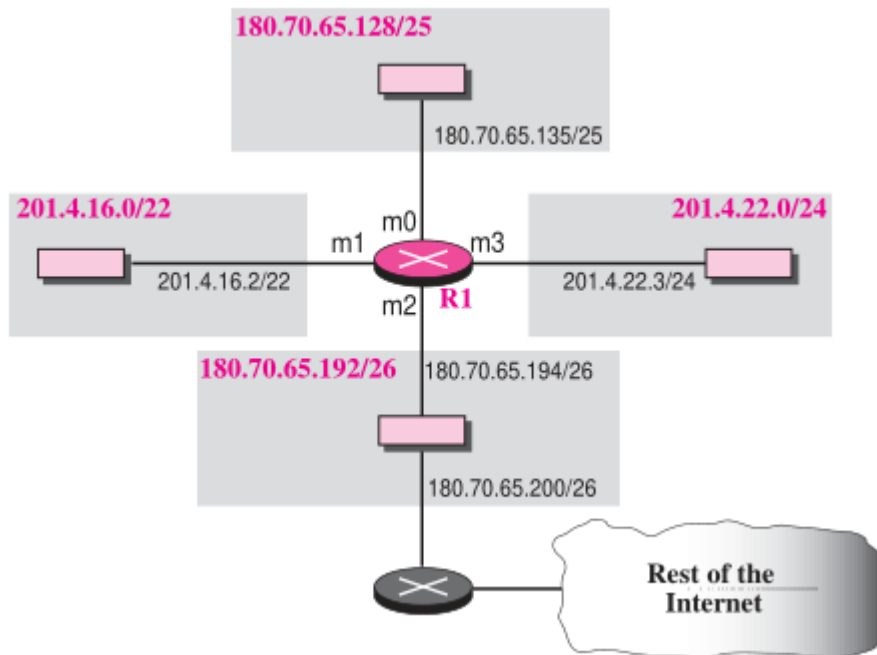
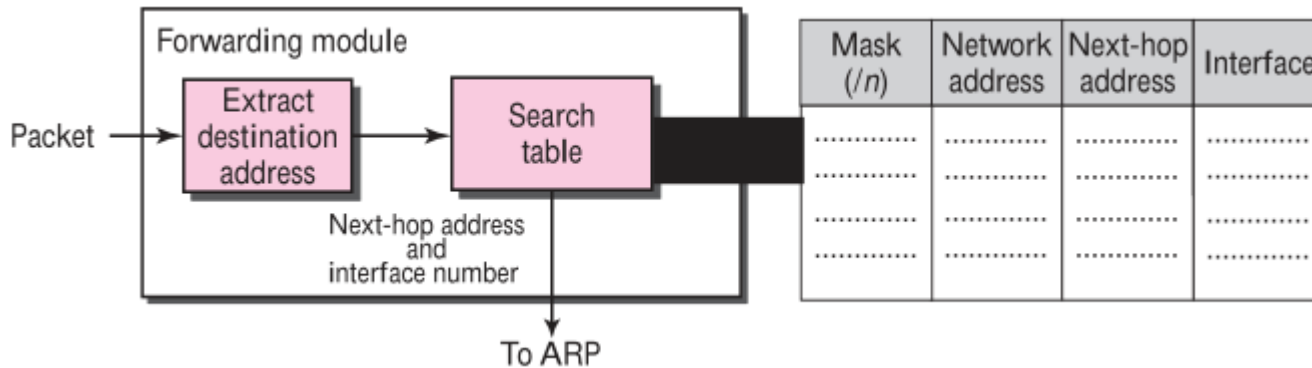
- u oparciu o adres domyślny (ang. default)



Forwarding – w podejściu klasowym



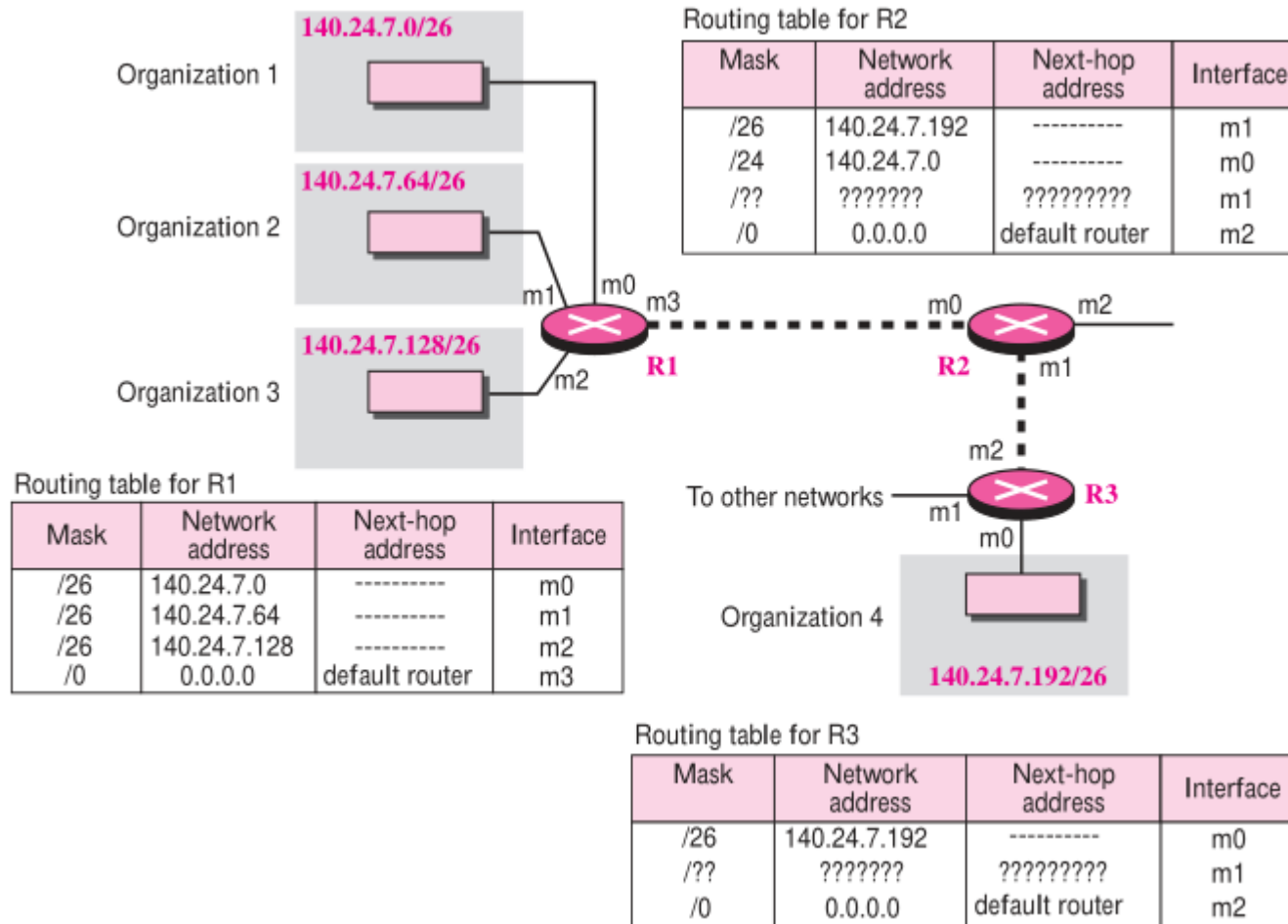
Forwarding – w podejściu bezklasowym



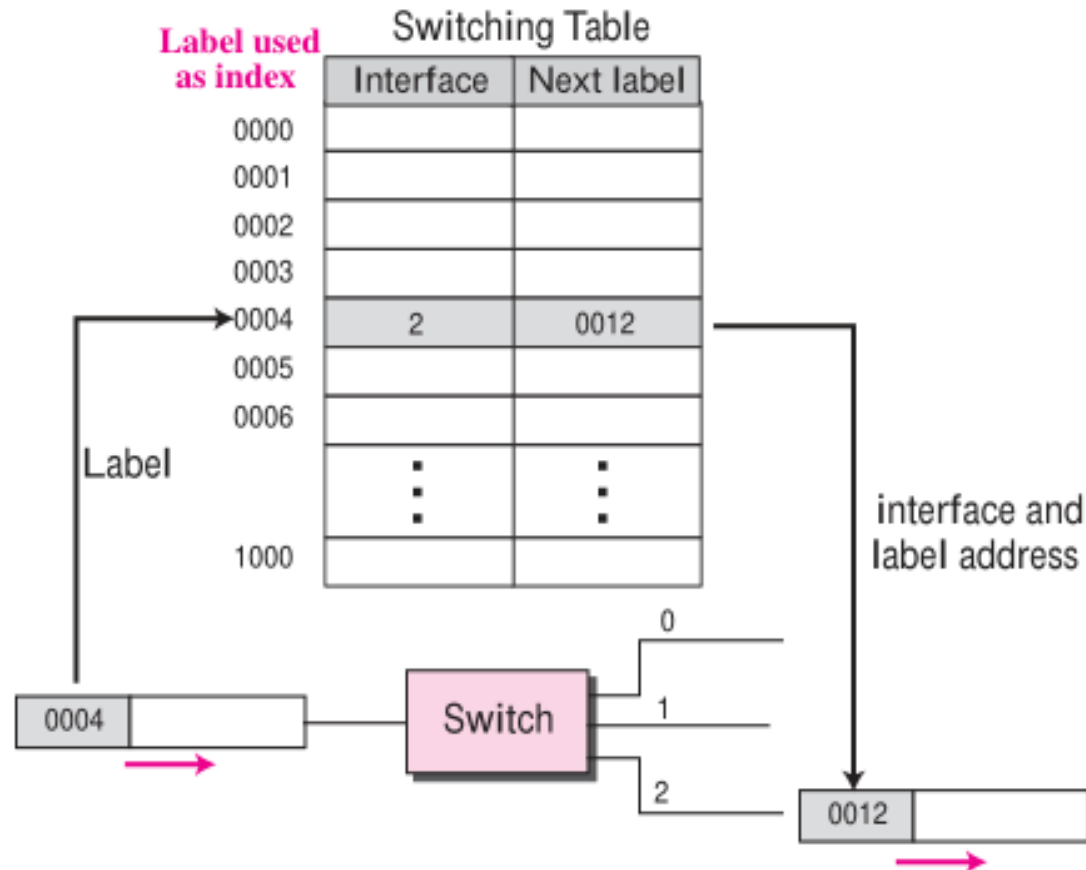
Mask	Network Address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3
/22	201.4.16.0		m1
Default	Default	180.70.65.200	m2



Forwarding – najdłuższa maska

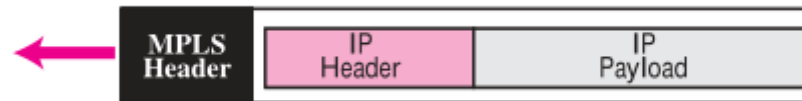


Forwarding – MPLS

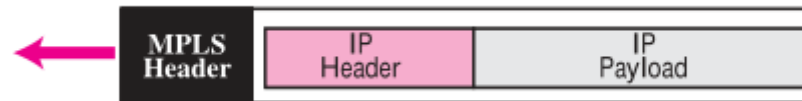


MPLS – przekazują pakiety po adresie docelowym (jak router) lub za pomocą etykiety (jak switch)

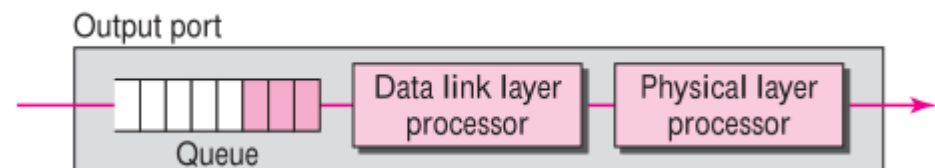
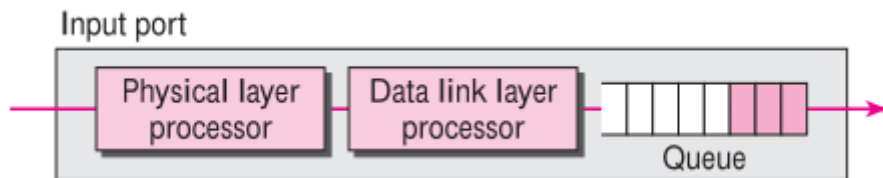
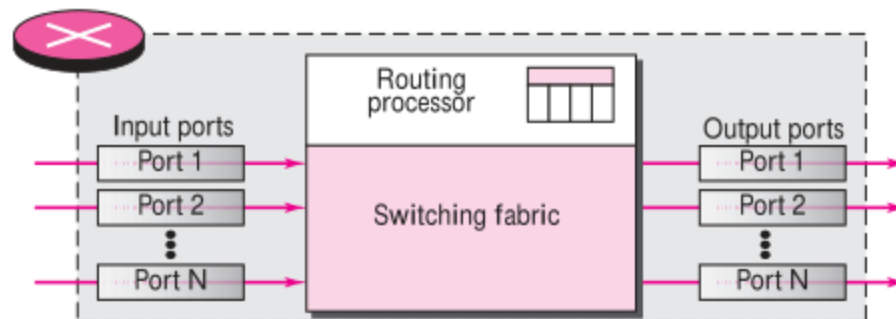
Forwarding – MPLS



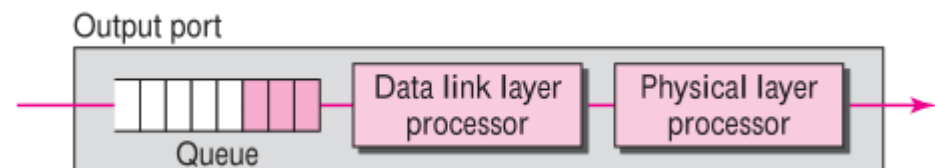
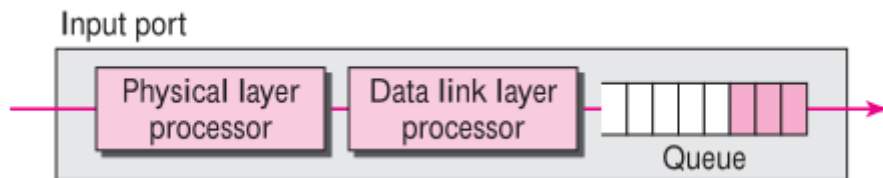
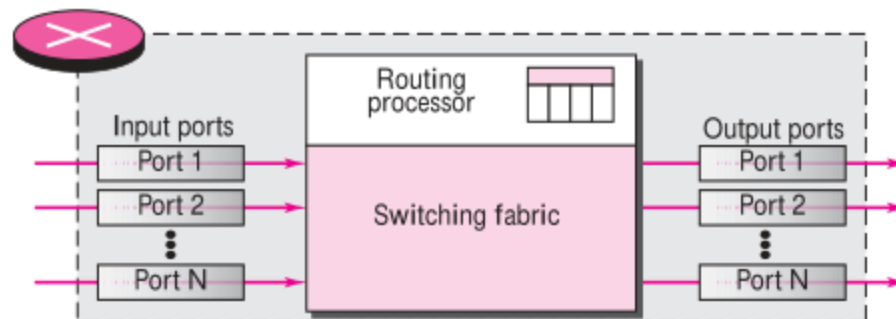
Forwarding – MPLS



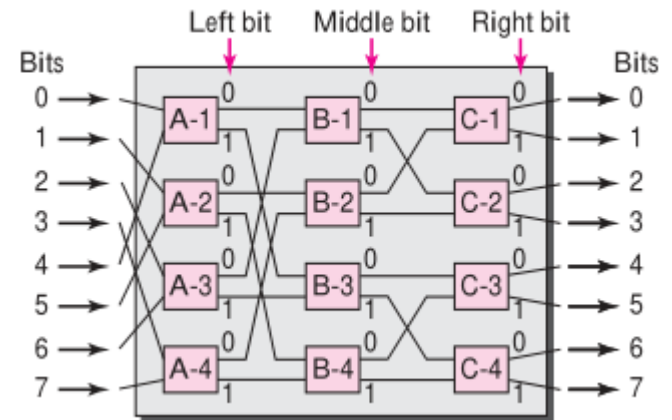
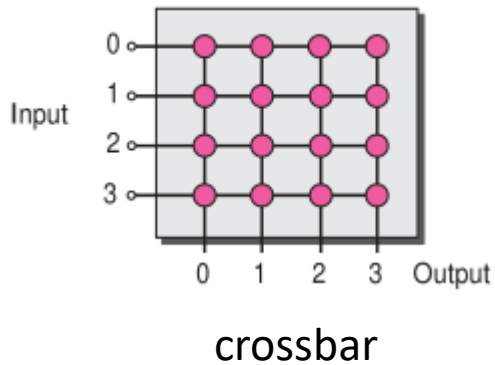
Router



Router

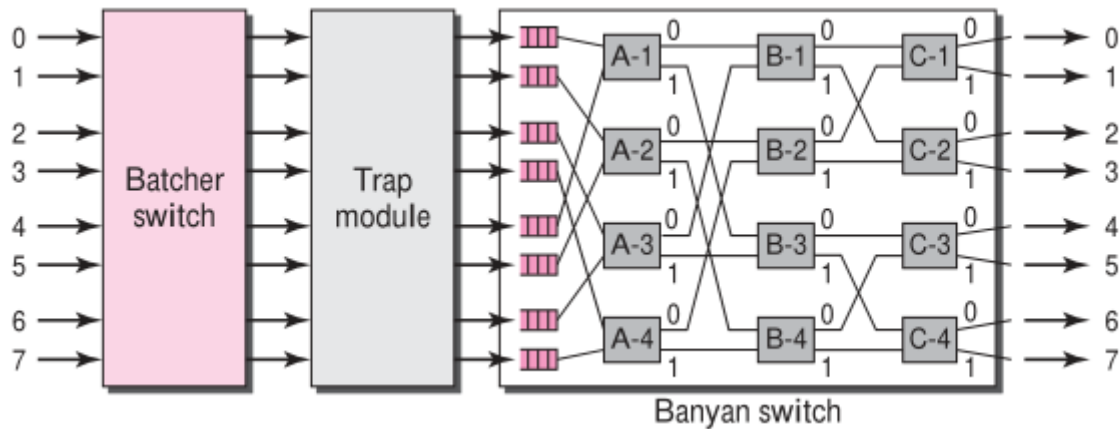
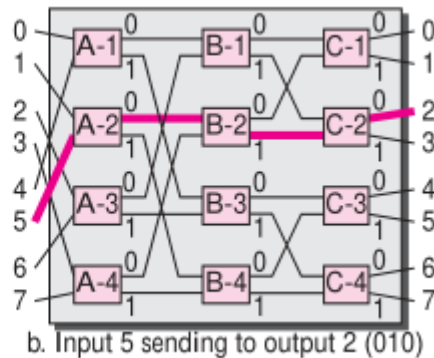
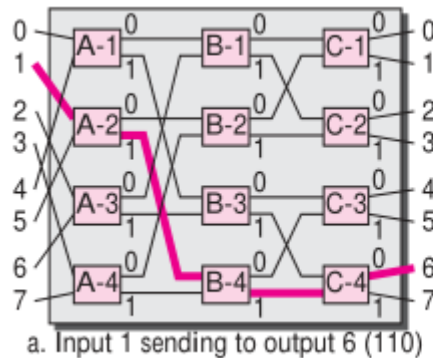


Router / switching



Banyan switch

Router / switching





POLITECHNIKA POZNAŃSKA

Poznań University of Technology



Distributed Systems Group

Sieci komputerowe

Wykład 5: ARP

16.05.2024

Wykład w całości przygotowany na
podstawie książki i slajdów: TCP/IP
Protocol suite 4ed, Behrouz A. Forouzan

Adress resultion protocol

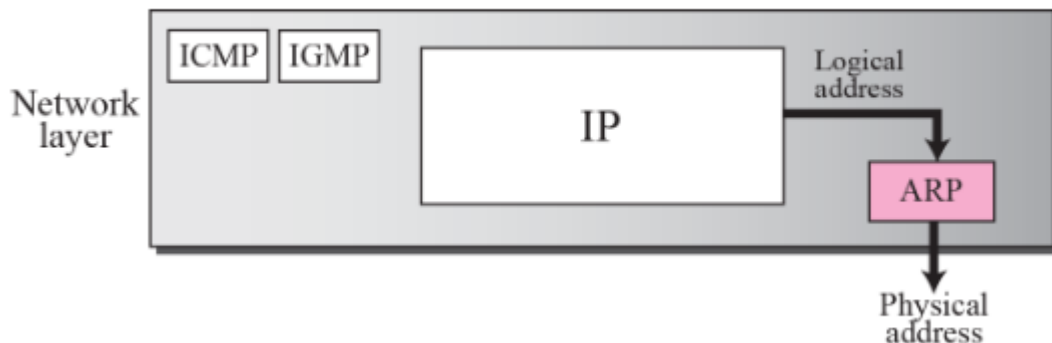


Dostarczenie pakietu wymaga od hosta lub routera dwóch adresów:

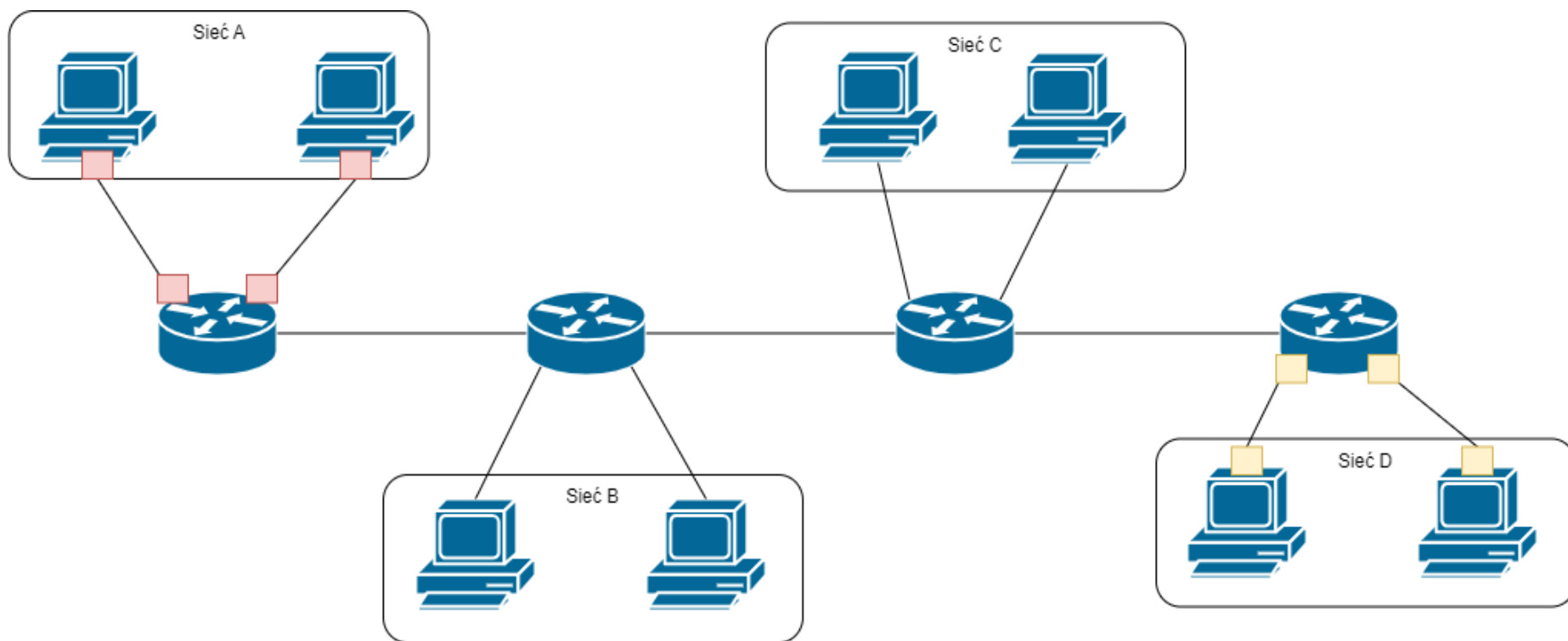
- **adresu fizycznego** np. MAC
- **adresu logicznego** np. IP

Odwzorowanie to może być:

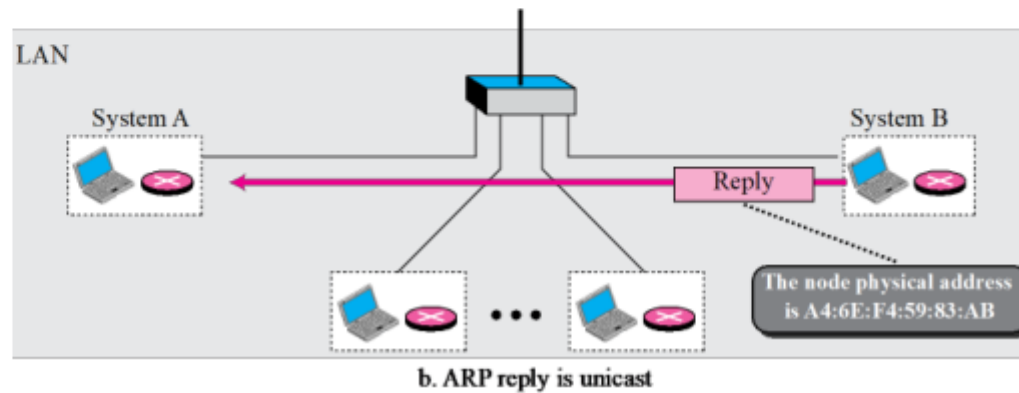
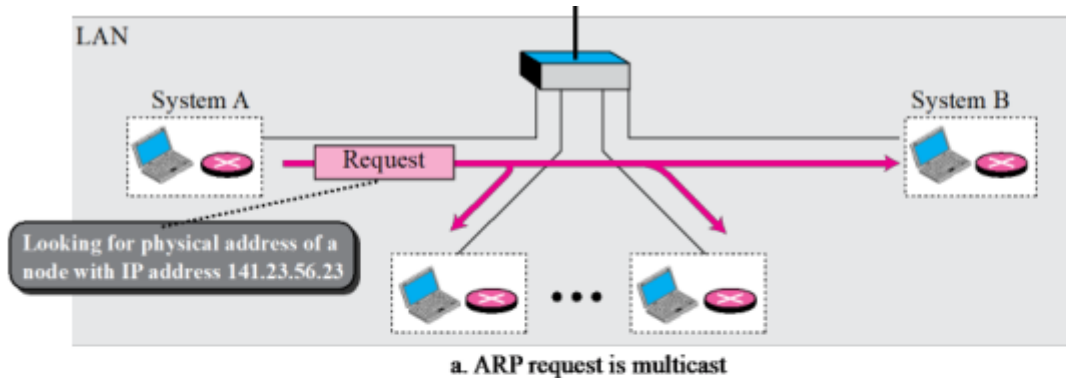
- **Statyczne**
 - Problemy (zmiana)
 - mobilne
- **dynamiczne**
 - ARP
 - RARP



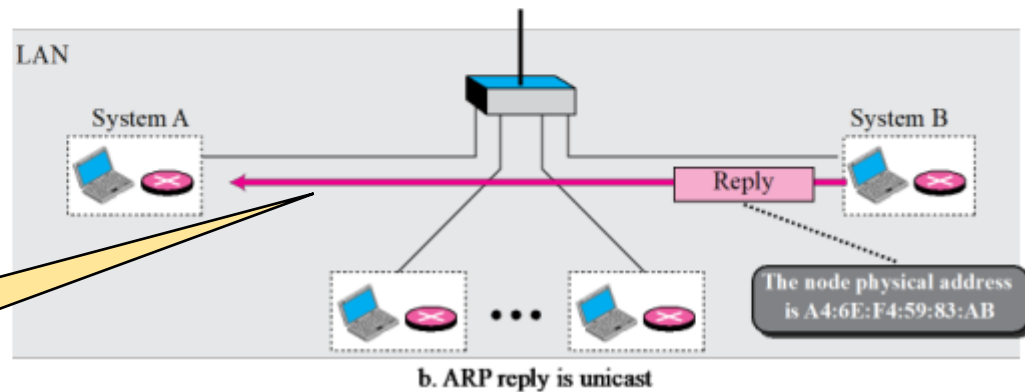
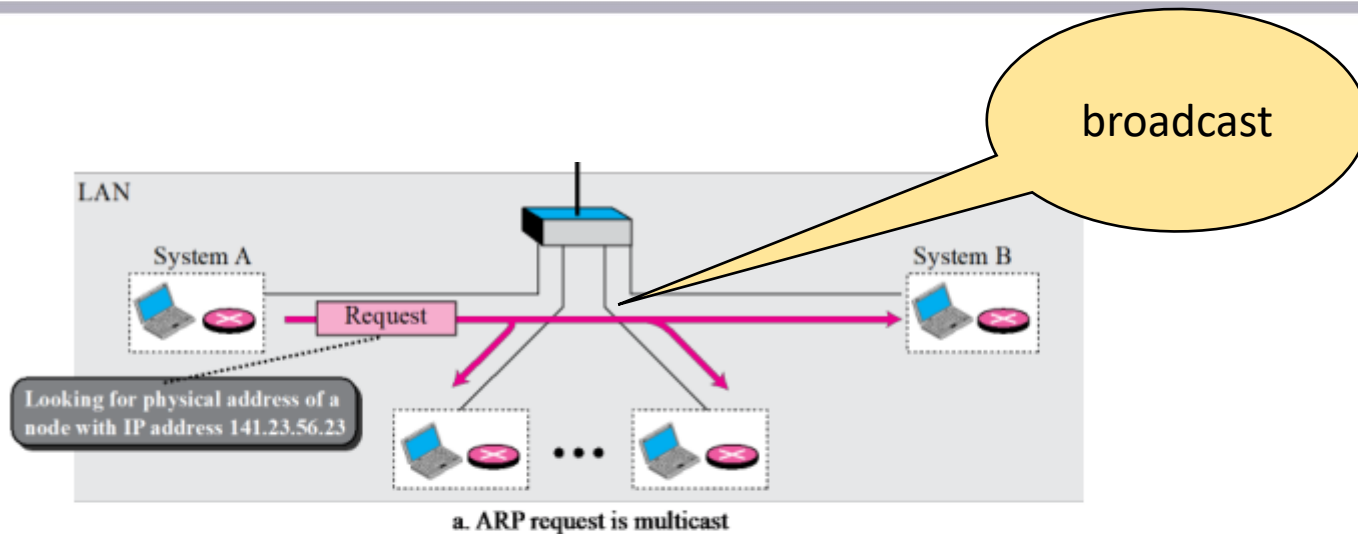
Adress resolution protocol



Address resolution protocol

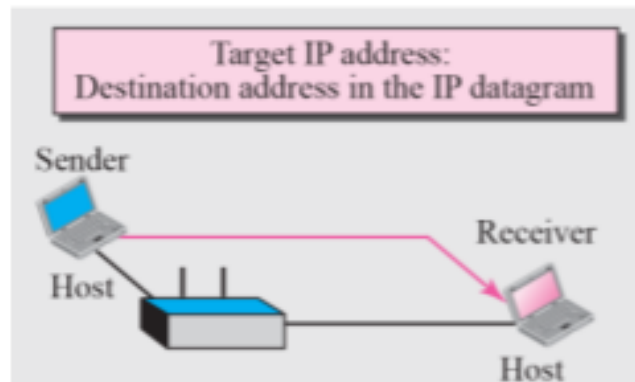


Address resolution protocol

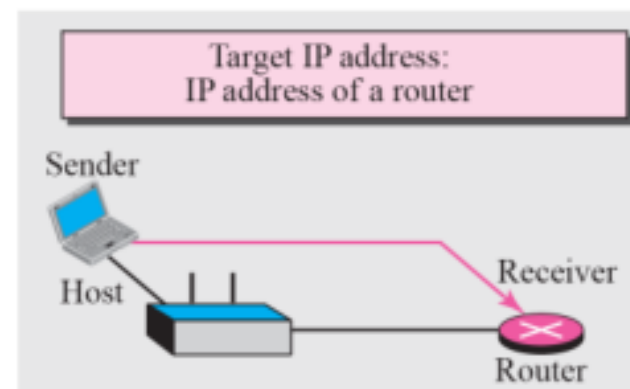


ARP – 4 przypadki wykorzystania

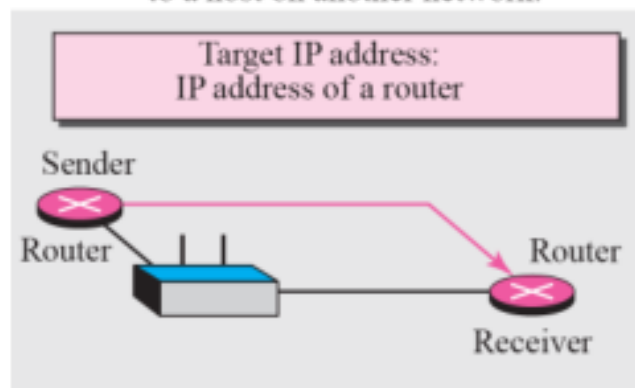
Case 1: A host has a packet to send to a host on the same network.



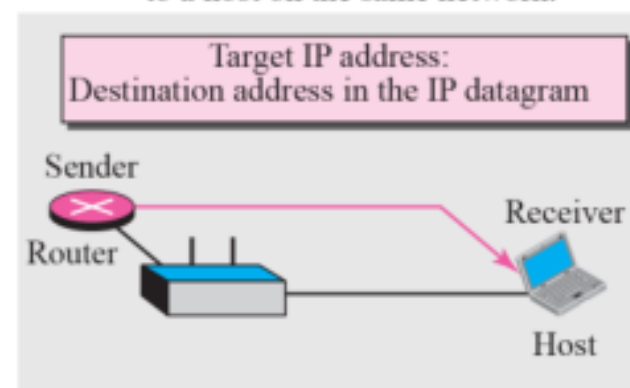
Case 2: A host has a packet to send to a host on another network.



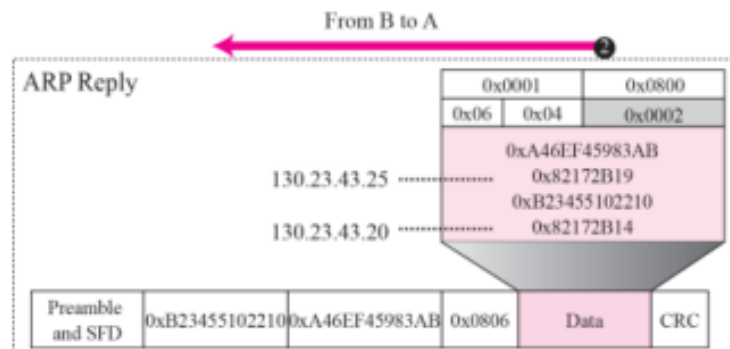
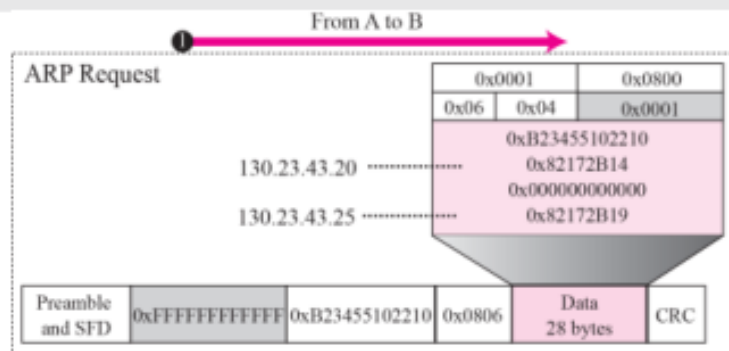
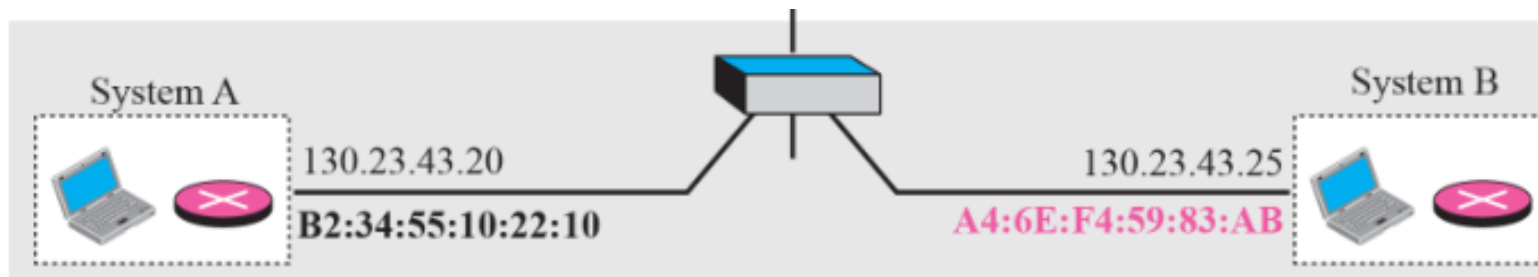
Case 3: A router has a packet to send to a host on another network.



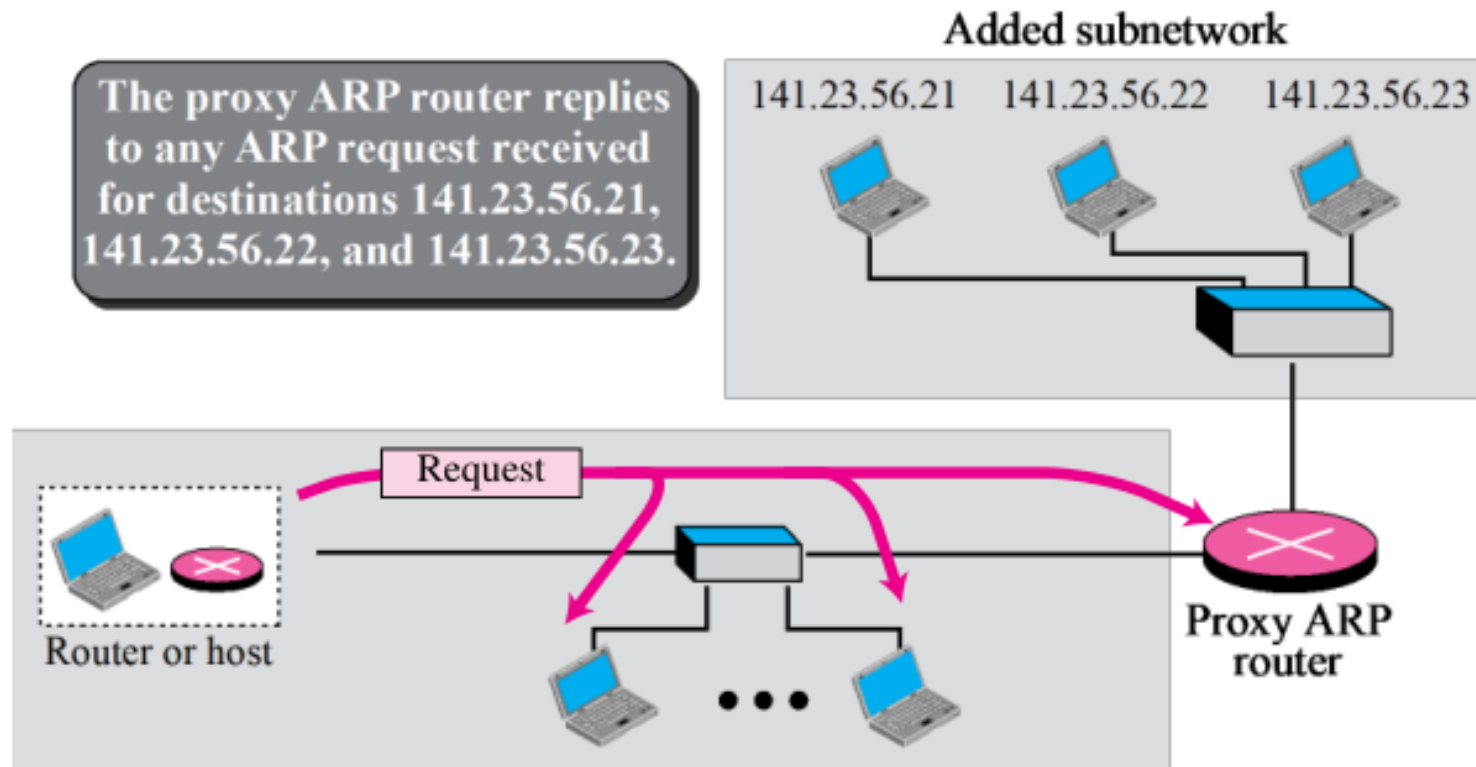
Case 4: A router has a packet to send to a host on the same network.



ARP – przykład struktury



Proxy ARP

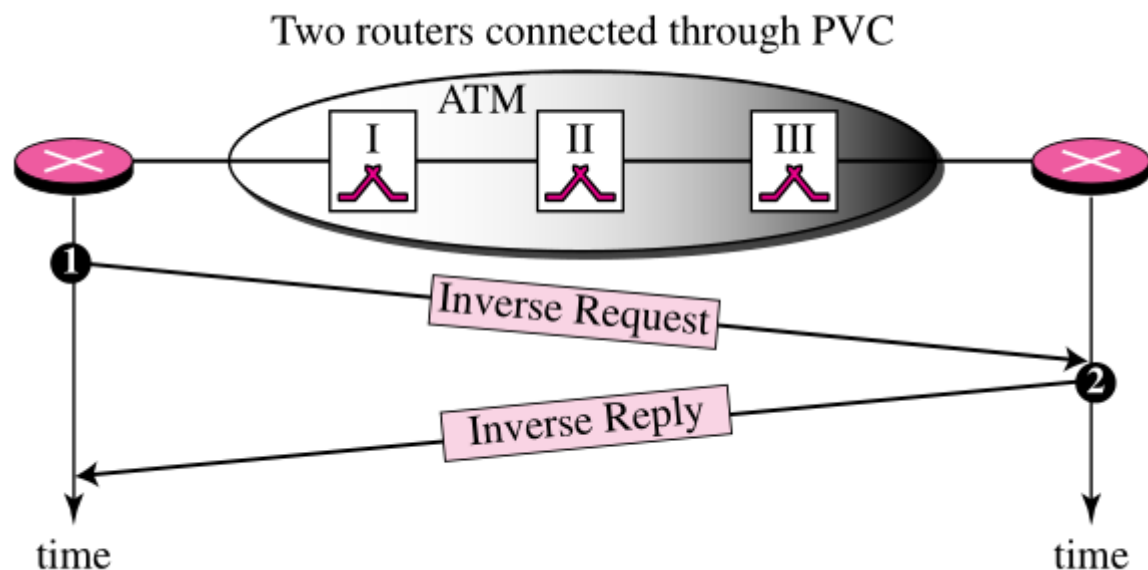


ATM nie ma rozgłaszania.

LAN ma rozgłaszanie.

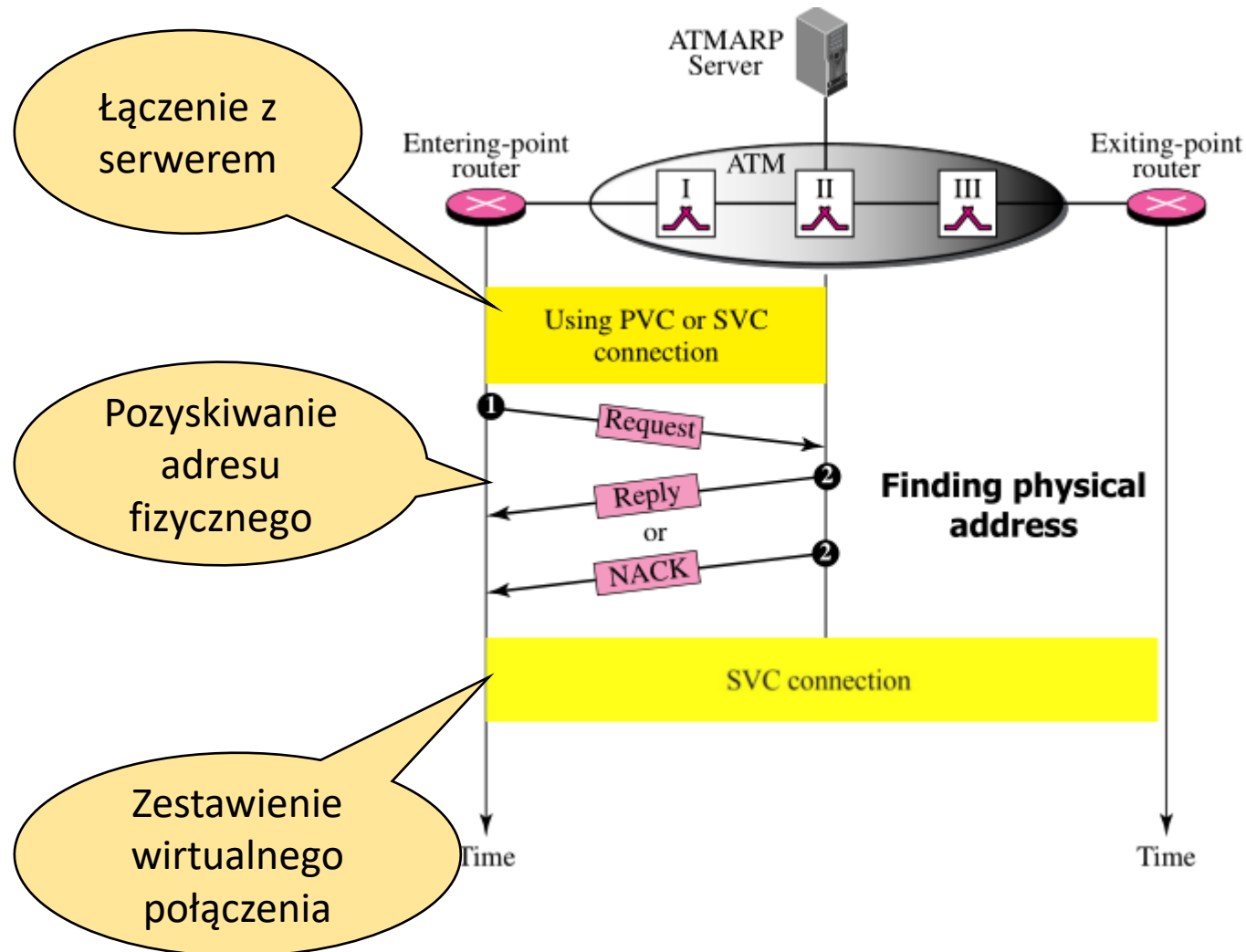
Sposoby połączenia dwóch routerów w sieciach ATM:

- Stałe wirtualne połączenie (PVC)
- Zmienne wirtualne połączenie (SVC)

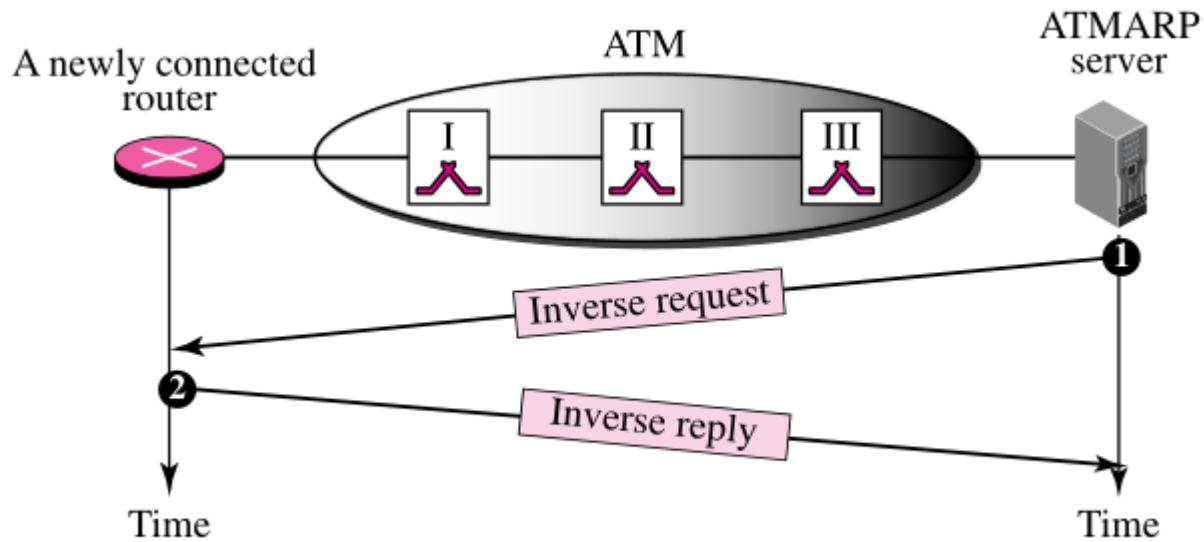


Adres fizyczny zbindowany z nowo połączonym PVC po wymianie inverse request i inverse replay.

ARP w SVC



ARP w ATM – nowy router

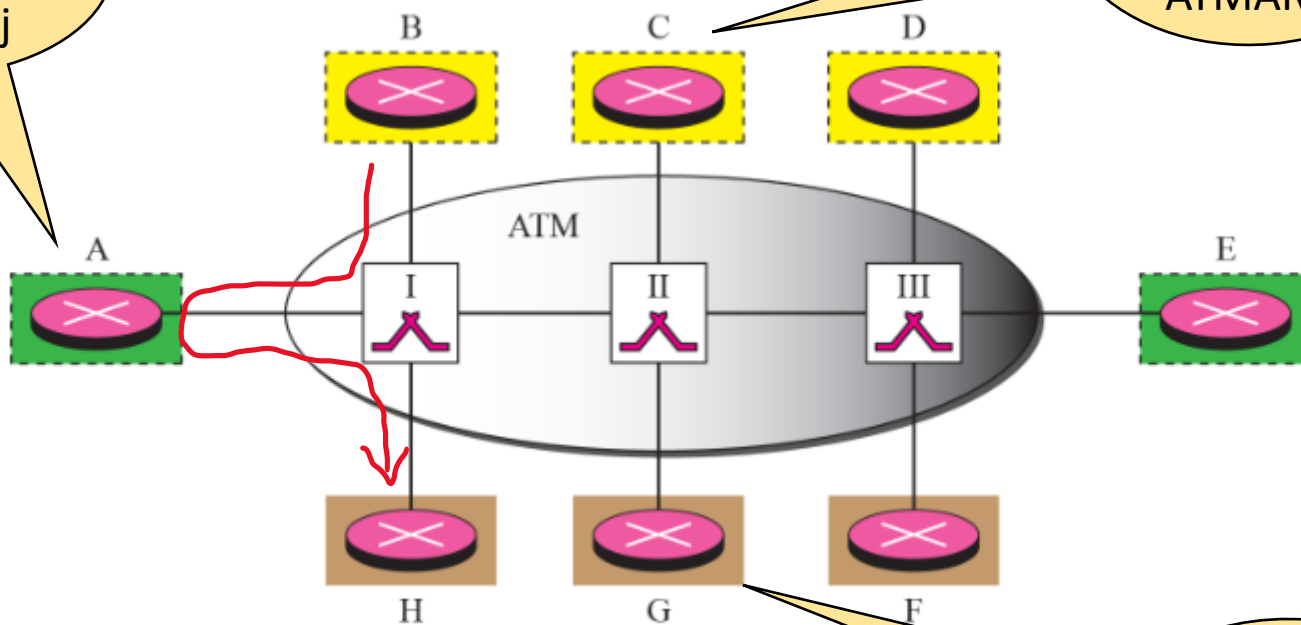


ARP w ATM – wiele logicznych sieci



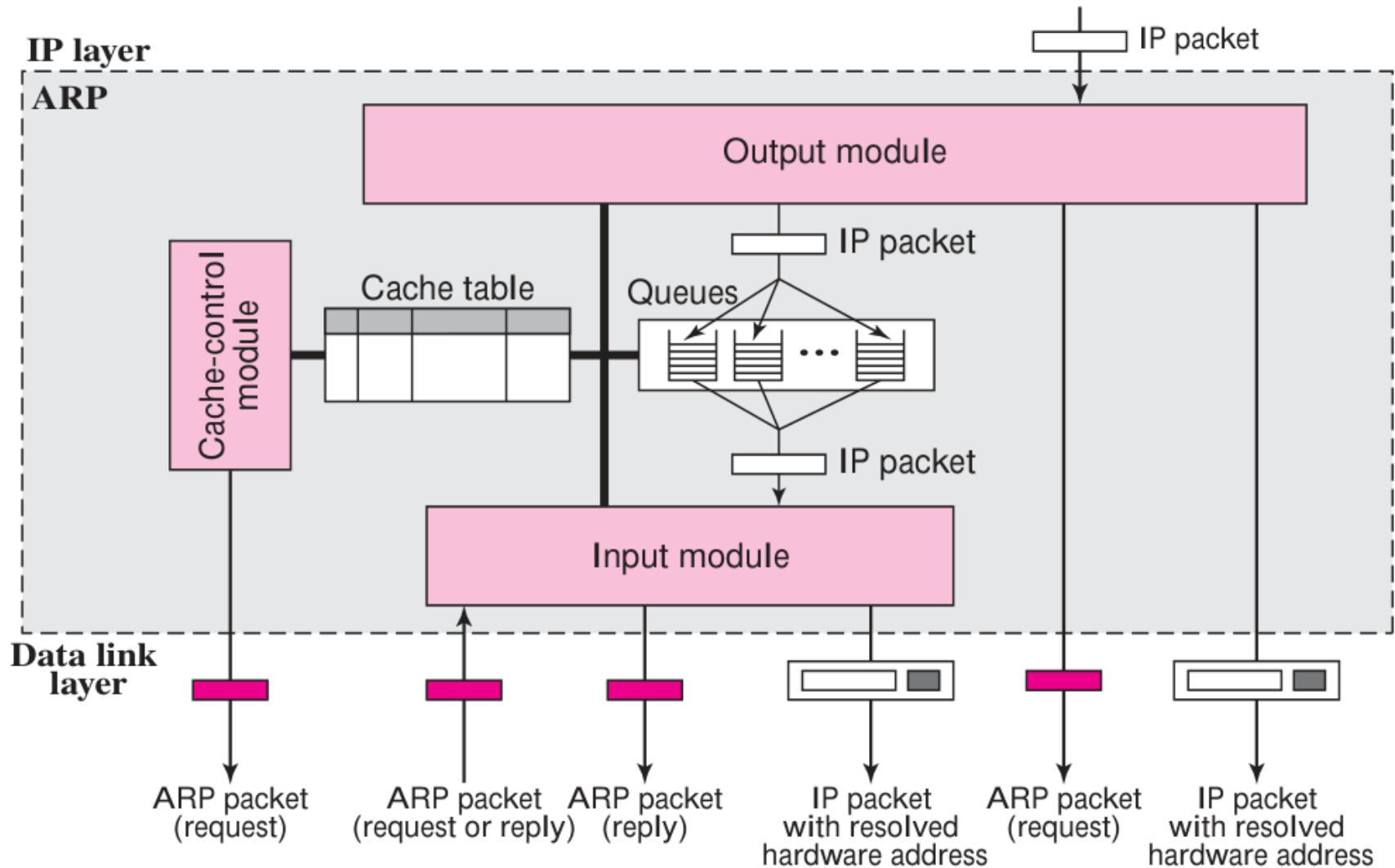
Jest w sieci
żółtej i
brązowej

Musi mieć
swoją
serwer
ATMARP



Musi mieć
swoją
serwer
ATMARP

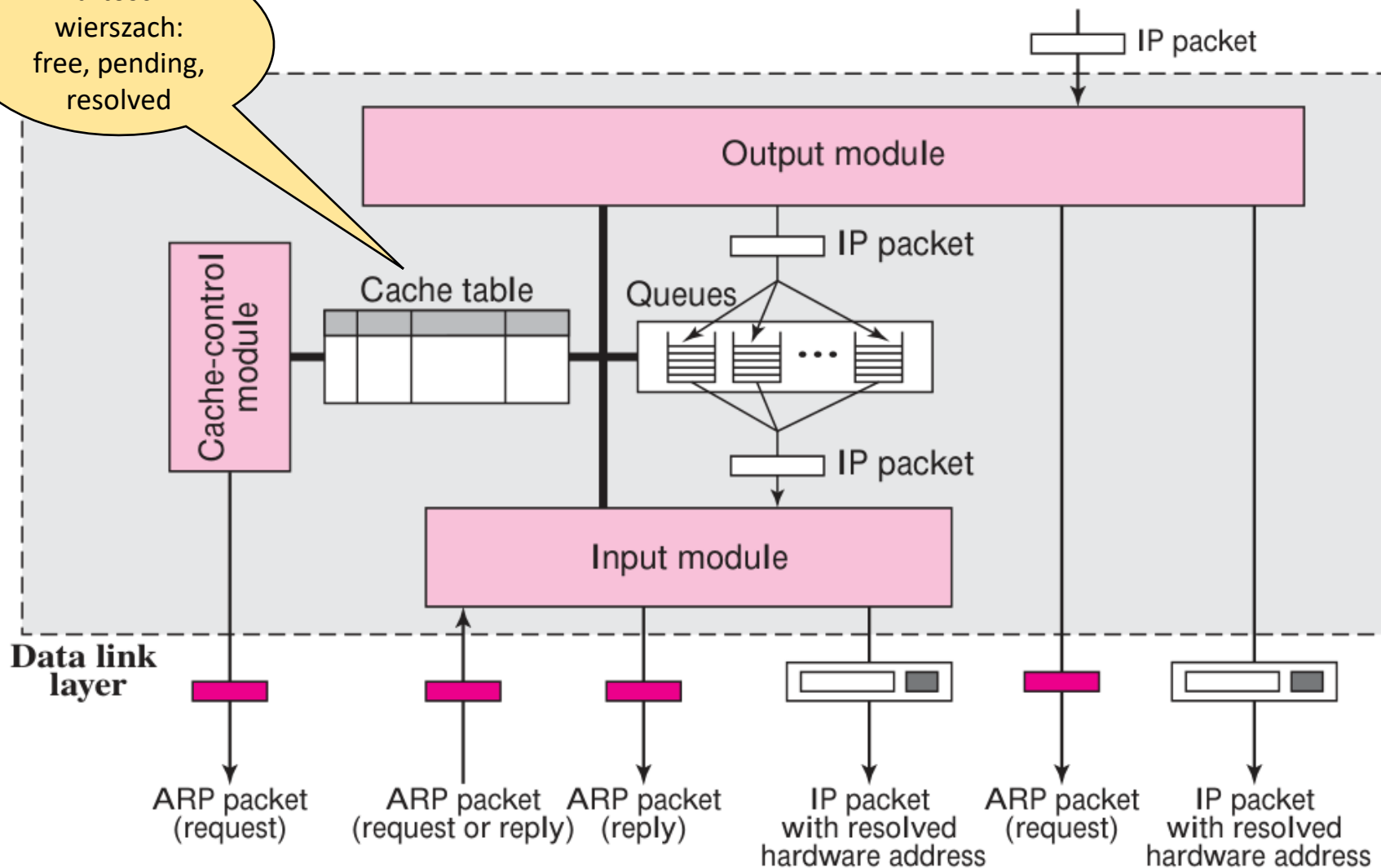
Moduł ARP



Moduł ARP



Wartości w wierszach:
free, pending,
resolved



Moduł ARP



```
1  ARP_Output_Module ( )
2  {
3      Sleep until an IP packet is received from IP software.
4      Check cache table for an entry corresponding to the
5          destination of IP packet.
6      If (entry is found)
7      {
8          If (the state is RESOLVED)
9          {
10             Extract the value of the hardware address from the entry.
11             Send the packet and the hardware address to data
12                 link layer.
13             Return
14         } // end if
15         If (the state is PENDING)
16         {
17             Enqueue the packet to the corresponding queue.
18             Return
19         } //end if
20     } //end if
21     If (entry is not found)
22     {
23         Create a cache entry with state set to PENDING and
24             ATTEMPTS set to 1.
25         Create a queue.
26         Enqueue the packet.
27         Send an ARP request.
28         Return
29     } //end if
30 } //end module
```

Moduł ARP



```
1  ARP_Input_Module ( )
2  {
3      Sleep until an ARP packet (request or reply) arrives.
4      Check the cache table to find the corresponding entry.
5      If (found)
6      {
7          Update the entry.
8          If (the state is PENDING)
9          {
10             While (the queue is not empty)
11             {
12                 Dequeue one packet.
13                 Send the packet and the hardware address.
14             }//end if
15         }//end if
16     }//end if
17     If (not found)
18     {
19         Create an entry.
20         Add the entry to the table.
21     }//end if
22     If (the packet is a request)
23     {
24         Send an ARP reply.
25     }//end if
26     Return
27 }//end module
```

Moduł ARP



```
1 ARP_Cache_Control_Module ( )
2 {
3     Sleep until the periodic timer matures.
4     Repeat for every entry in the cache table
5     {
6         If (the state is FREE)
7         {
8             Continue.
9         } //end if
10        If (the state is PENDING)
11        {
```

```
12        Increment the value of attempts by 1.
13        If (attempts greater than maximum)
14        {
15            Change the state to FREE.
16            Destroy the corresponding queue.
17        } // end if
18        else
19        {
20            Send an ARP request.
21        } //end else
22        continue.
23    } //end if
24    If (the state is RESOLVED)
25    {
26        Decrement the value of time-out.
27        If (time-out less than or equal 0)
28        {
29            Change the state to FREE.
30            Destroy the corresponding queue.
31        } //end if
32    } //end if
33 } //end repeat
34 Return.
35 } //end module
```

Moduł ARP - przykład



<i>State</i>	<i>Queue</i>	<i>Attempt</i>	<i>Time-Out</i>	<i>Protocol Addr.</i>	<i>Hardware Addr.</i>
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

Moduł ARP - przykład



Moduł wyjściowy ARP odbiera datagram IP (z warstwy sieciowej) z adresem docelowym **114.5.7.89**. Sprawdza cache i stwierdza, że istnieje wpis dla tego miejsca docelowego ze stanem ROZWIĄZANY (R w tabeli). Wyodrębnia adres fizyczny, który wynosi **457342ACAE32** i wysyła pakiet oraz adres do warstwy łączy danych w celu wykonania transmisji. **Tabela pamięci podręcznej pozostaje taka sama.**

Moduł ARP - przykład



Dwadzieścia sekund później moduł wyjściowy ARP otrzymuje datagram (z warstwy sieciowej) z adresem docelowym 116.1.7.22. Sprawdza tabelę pamięci podręcznej i nie znajduje w niej odpowiedniego wpisu. Moduł dodaje wpis do tabeli ze stanem PENDING i wartością Próby 1. Tworzy nową kolejkę dla tego adresu docelowego i umieszcza pakiet w kolejce. Następnie wysyła żądanie ARP do warstwy łącza danych dla tego adresu docelowego. Nowa tabela pamięci podręcznej jest pokazana w tabeli 8.6.

Moduł ARP - przykład



<i>State</i>	<i>Queue</i>	<i>Attempt</i>	<i>Time-Out</i>	<i>Protocol Addr.</i>	<i>Hardware Addr.</i>
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

Moduł ARP - przykład



Piętnaście sekund później moduł wejściowy ARP odbiera komunikat ARP z adresem protokołu docelowego (IP) 188.11.8.71. Moduł sprawdza tabelę i znajduje ten adres. Zmienia stan wpisu na ROZWIĄZANY i ustawia wartość limitu czasu na 900. Następnie moduł dodaje docelowy adres sprzętowy (E34573242ACA) do wpisu. Teraz uzyskuje dostęp do kolejki 18 i wysyła wszystkie pakiety z tej kolejki, jeden po drugim, do warstwy łącza danych.

Moduł ARP - przykład



<i>State</i>	<i>Queue</i>	<i>Attempt</i>	<i>Time-Out</i>	<i>Protocol Addr.</i>	<i>Hardware Addr.</i>
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
R	18		900	188.11.8.71	E34573242ACA

Moduł ARP - przykład



Dwadzieścia pięć sekund później następuje aktualizacja modułu kontroli pamięci podręcznej dla każdego wpisu. Wartości limitów czasu dla pierwszych trzech wpisów w stanie RESOLVED zmniejszanych zostaje o 60. Wartość time-out dla ostatniego wpisu w stanie RESOLVED jest zmniejszany o 25. Stan przedostatniego wpisu zostaje ustawiony na FREE bo time-out spadł na 0. Wartość prób dla każdego z trzech oczekujących wpisów zwiększa się o jeden. Po inkrementacji próby wartość dla jednego wpisu (tego o adresie IP 201.11.56.7) wynosi więcej niż maksimum; stan zostanie zmieniony na FREE, a kolejka zostanie usunięta, a do oryginalnego adresu docelowego zostanie wysłany odpowiedni komunikat ICMP

Moduł ARP - przykład



<i>State</i>	<i>Queue</i>	<i>Attempt</i>	<i>Time-Out</i>	<i>Protocol Addr.</i>	<i>Hardware Addr.</i>
R	5		840	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
F					
R	8		390	114.5.7.89	457342ACAE32
P	12	2		220.55.5.7	
P	23	2		116.1.7.22	
F					
R	18		875	188.11.8.71	E34573242ACA

Dziękuję za uwagę. Pytania?

