

**Curs 05**

# Protocolul IP



# Objective

- Nivelul rețea
- Protocolul IPv4
- ARP

# Nivelul rețea

- Funcții
- Protocele



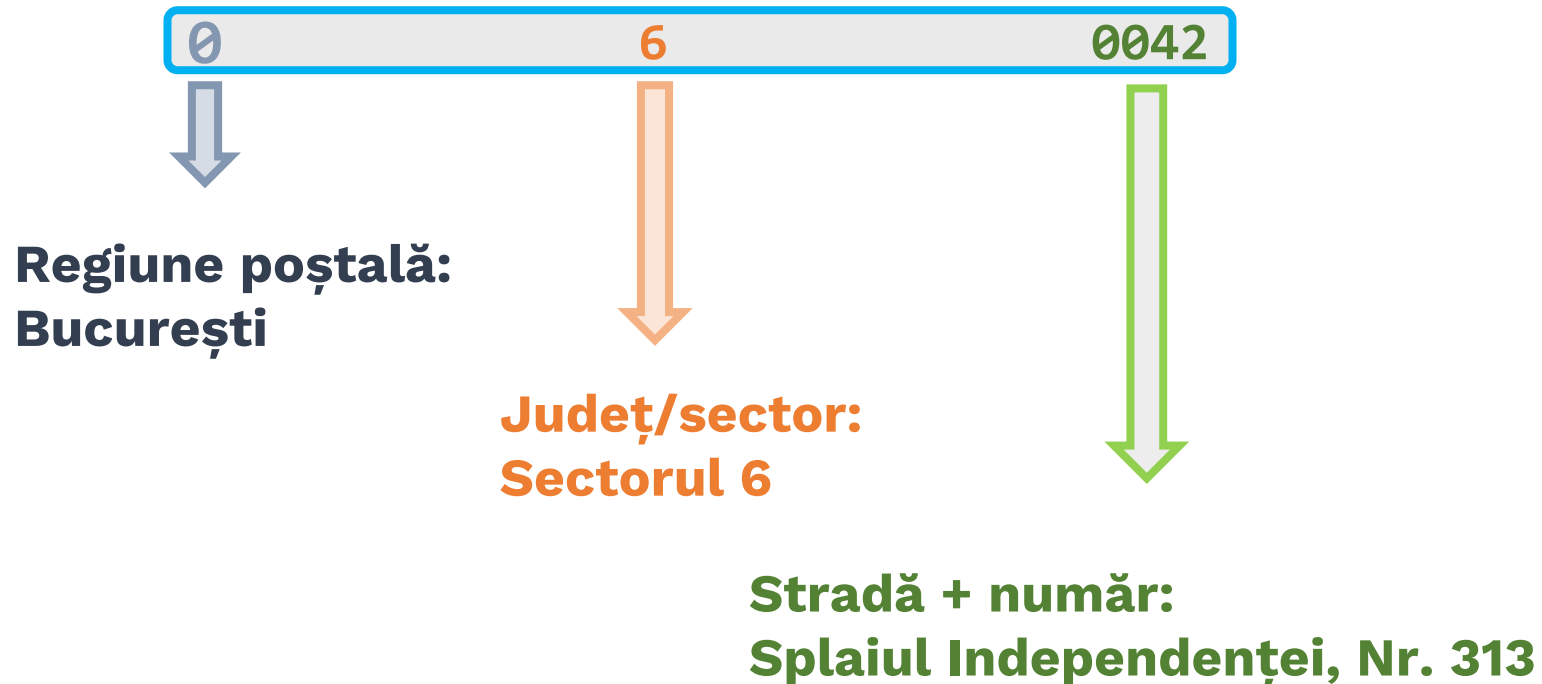
# Necesitatea unei adresări globale



- Problemă: Adresele MAC sunt ineficiente pentru rețele mari:
  - Folosesc o schemă de adresare plată ce nu scalează
- Consecință: adresele MAC sunt folosite doar cu vizibilitate locală (în domeniul local de broadcast)
- Soluție: Este necesară folosirea unui alt set de adrese pentru adresare globală
  - Aceste adrese trebuie să fie organizate ierarhic pentru a putea fi gestionate de echipamentele de rețea

# Exemplu de adresare ierarhică

- Codul poștal:

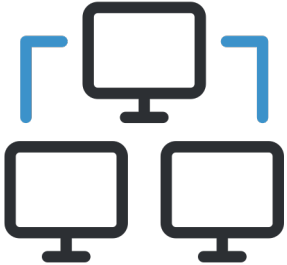


# Funcțiile nivelului rețea



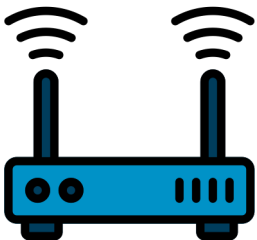
## Adresare globală

- ✓ Introduce un protocol cu adresare ierarhică numit IP (Internet Protocol)
- ✓ Fiecare dispozitiv este identificat în mod unic la nivel global prin adresa sa IP



## Comunicație end-to-end

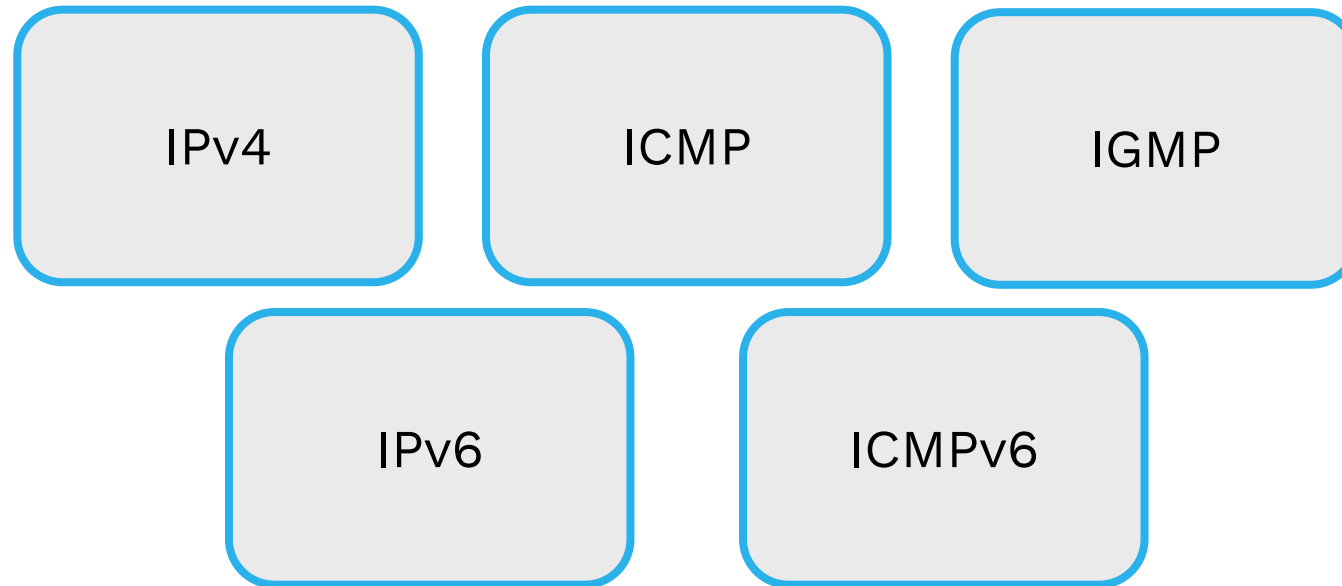
- ✓ Protocoalele nivelului rețea sunt de tip best-effort și nu stabilesc conexiuni
- ✓ Stabilirea conexiunilor este responsabilitatea protocoalelor de nivel superior



## Rutare (dirijare)

- ✓ Dispozitive intermediare numite rutere iau decizii de dirijare a traficului în funcție de destinație

# Protocoloale de nivel rețea



# IPv4

- Funcții
- Antet
- Adresa IPv4
- Procesul de subnetare
- VLSM





# Funcțiile IPv4

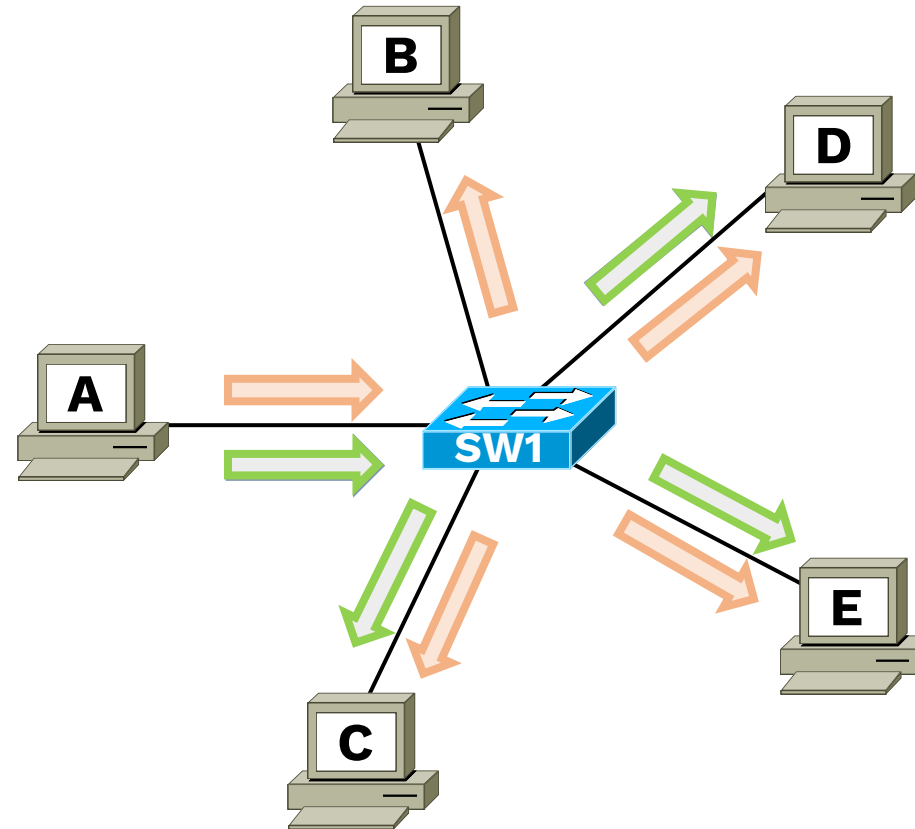
- IPv4: Internet Protocol, versiunea 4
- Definit în **RFC791**, în anul 1981
- IPv4 oferă fiecărui dispozitiv din Internet o adresă unică: **adresa IP**
- IPv4 adaugă informația de adresare prin **încapsulare**
- PDU-ul (Protocol Data Unit) rezultat ca urmare a încapsulării IP poartă numele de **pachet**
- Pe baza informației de adresare conținută în antetul IP se realizează dirijarea traficului în Internet

# Transmisii IPv4

Unicast

Broadcast

Multicast



# IANA

ARIN  
American Registry for Internet Numbers

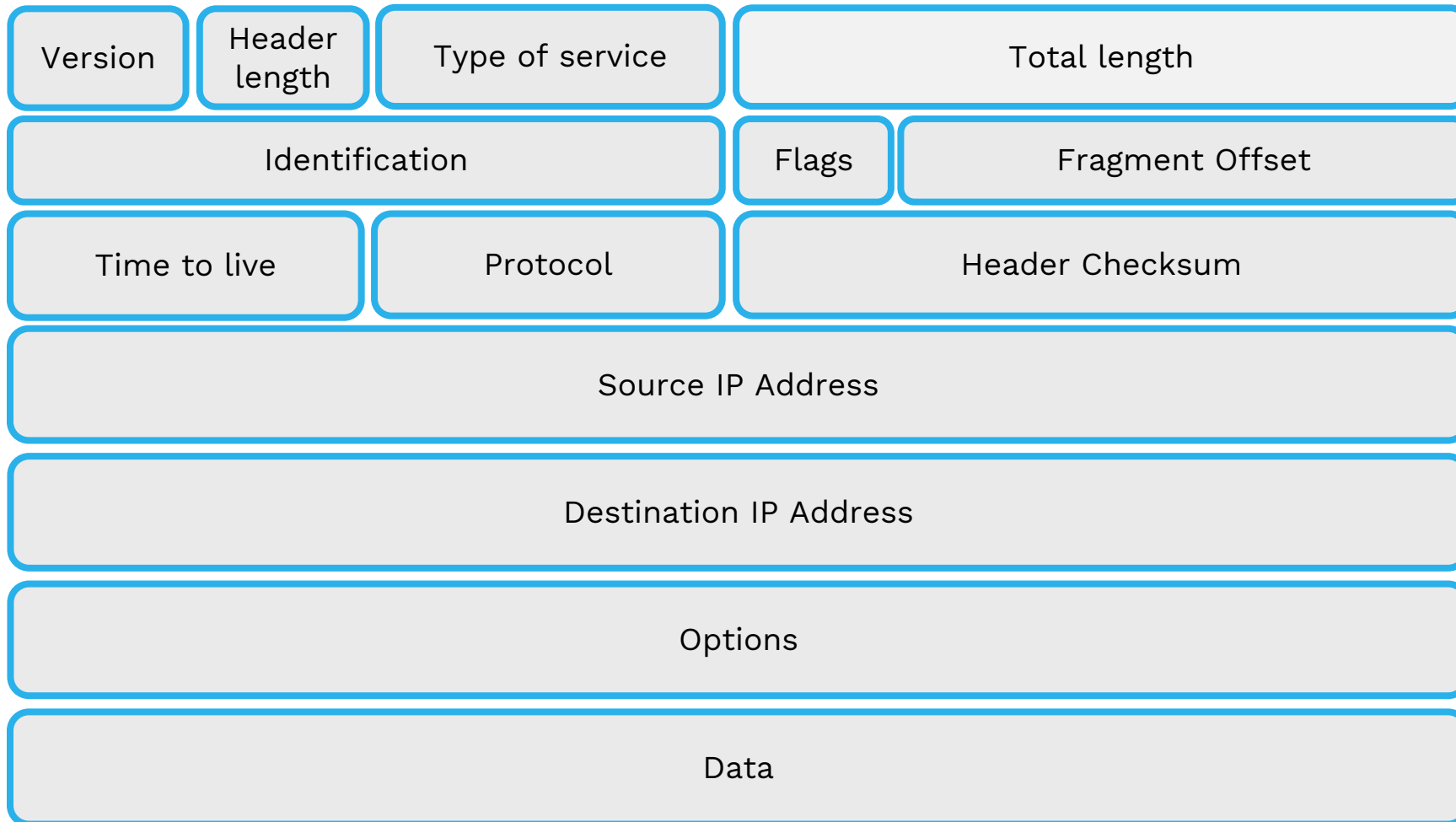
RIPE NCC  
RIPE NETWORK COORDINATION CENTRE

APNIC

lacnic

AFRINIC  
The Internet Number Registry for Africa

# Formatul antetului



# Adresa IPv4

- Adresa IPv4 este formată din 4 octeți
- Formatul cel mai folosit este zecimal cu punct:

141 . 85 . 241 . 139

- Utilă pentru calcule mai este reprezentarea adresei în format **binar**:

10001101 . 01010101 . 11110001 . 10001011

# Transformări binar ↔ zecimal

10000000	128
01000000	64
00100000	32
00010000	16
00001000	8
00000100	4
00000010	2
00000001	1

$$141 = 128 + 8 + 4 + 1$$

$$= 10001101$$

$$85 = 64 + 16 + 4 + 1$$

$$= 01010101$$

$$241 = 128 + 64 + 32 + 16 + 1$$

$$= 11110001$$

$$139 = 128 + 8 + 2 + 1$$

$$= 10001011$$

# Adresa IPv4

- Adresa IPv4 este compusă din două părți:
  - Partea de rețea
  - Partea de host
- Dispozitivele ce au partea de rețea comună sunt situate în aceeași rețea și pot comunica fără să aibă nevoie de un ruter
- Părțile de rețea și de host se determină folosind **masca de rețea (Subnet mask)**
- Masca de rețea este o adresă IP specială ce este formată dintr-un șir continuu de 1 urmat de un șir continuu de 0:

**11111111.11111111.11111111.00000000 = 255.255.255.0**

# Masca de rețea

- Deoarece notația zecimală a unei măști de rețea este dificil de utilizat s-a introdus o notație specială:

**11111111.11111111.11111111.00000000**  $\equiv$  /24

- /24 poartă numele de prefixul rețelei și reprezintă numărul de 1 din masca rețelei
- O reprezentare completă a unui IP de stație împreună cu rețeaua din care face parte devine:

**141.85.241.139/24**



# Adresa de rețea

- Prin aplicarea operației de **AND** pe biți între mască și adresa IP se obține **adresa de rețea**:

Partea de rețea				Partea de host			
141	.	85	.	241	.	139	
10001101	.	01010101	.	11110001	.	10001011	AND
11111111	.	11111111	.	11111111	.	00000000	
10001101	.	01010101	.	11110001	.	00000000	
141	.	85	.	241	.	0	

- Adresele de rețea au toți biții din partea de host setați pe 0
- Adresa de rețea este folosită de stații pentru a determina dacă să trimită direct destinației sau gateway-ului pachetul

# Adresa de broadcast

- Prin aplicarea operației de **OR** pe biți între inversa măștii și adresa IP se obține **adresa de broadcast** a rețelei:

Partea de rețea		Partea de host				
141	.	85	.	241	.	139
10001101	.	01010101	.	11110001	.	10001011
00000000	.	00000000	.	00000000	.	11111111
10001101	.	01010101	.	11110001	.	11111111
141	.	85	.	241	.	255

- Adresele de broadcast au toți biții din partea de host setați pe 1
- Adresa de broadcast este folosită ca adresă destinație în pachete ce vrem să ajungă la toate dispozitivele din respectiva rețea

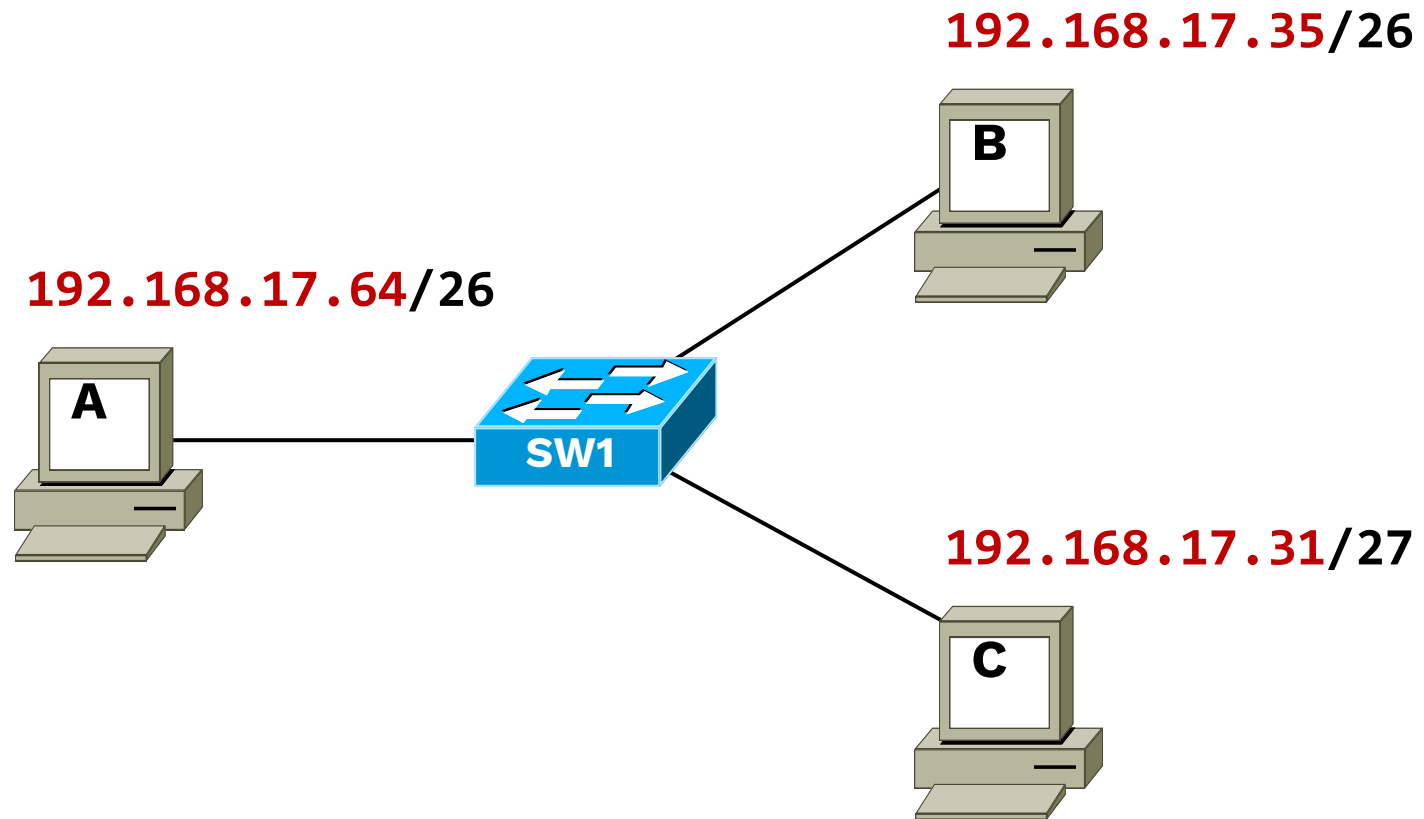
# Adresa de loopback

- O interfață specială a dispozitivelor de rețea este interfața de loopback
- Interfața de loopback este virtuală și nu are asociată vreo interfață fizică
- Interfața de loopback este caracterizată prin adresa IP de loopback:

**127.0.0.1**

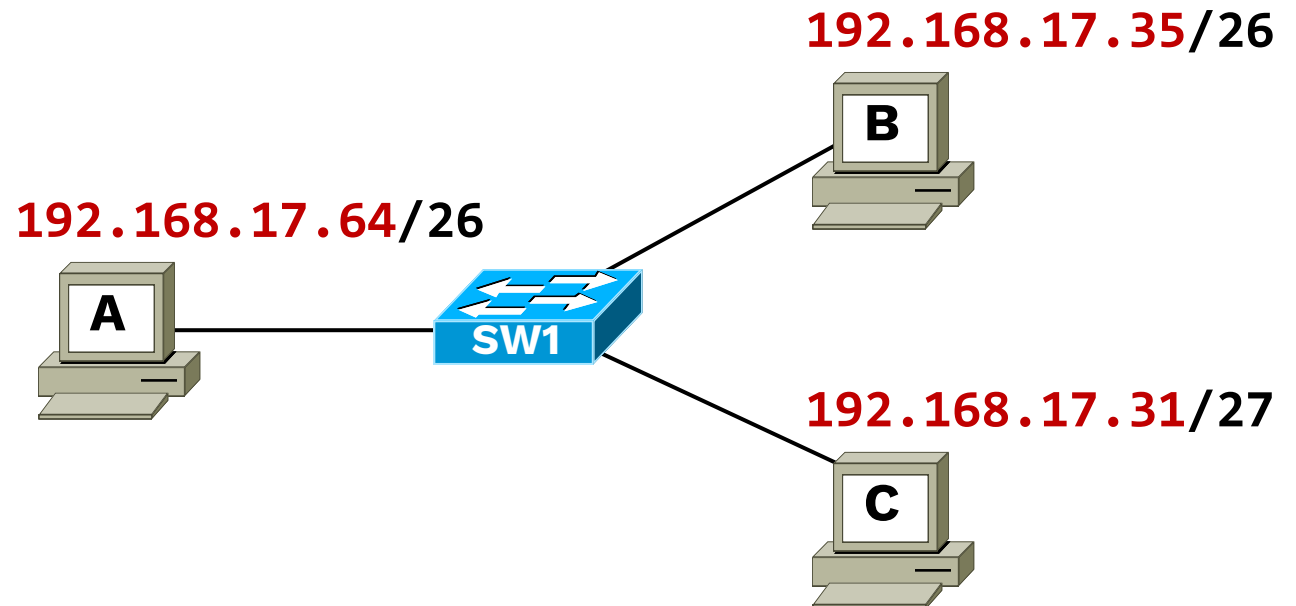
- Prin folosirea acestei interfețe se poate testa integritatea stivei de protocoale de pe un sistem

# Topologia exemplu



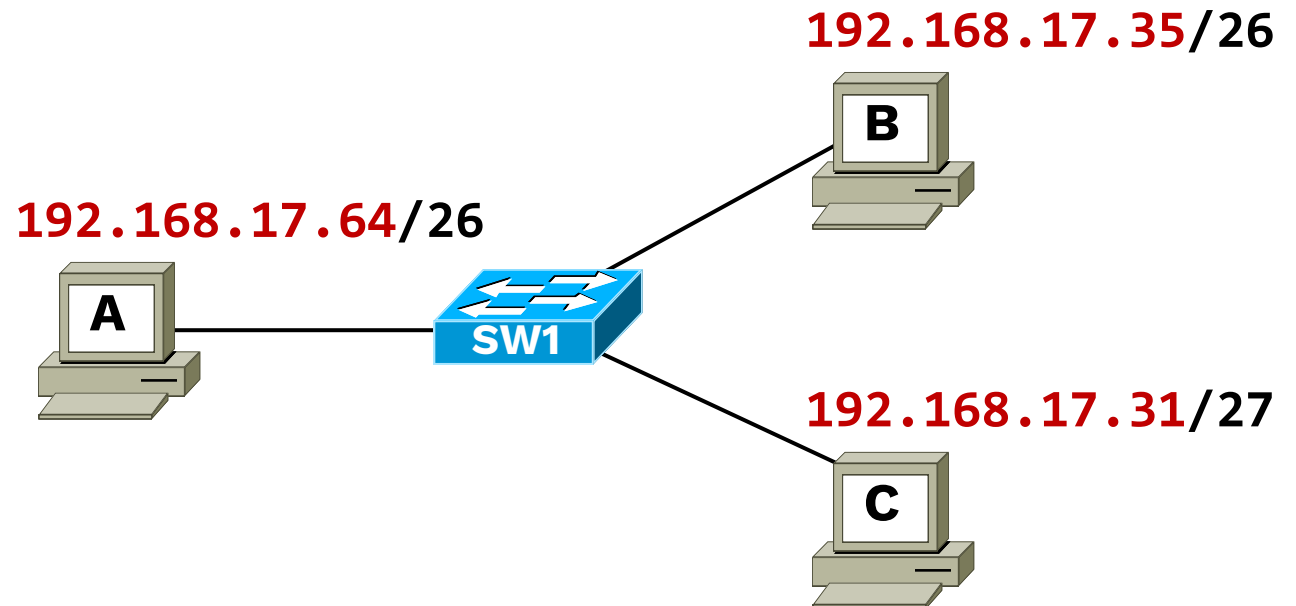
# Exercițiul 1: Verificarea configurației

- Stațiile sunt configurate cu IP-urile și măștile din figură. Există vreo problemă cu această configurație?



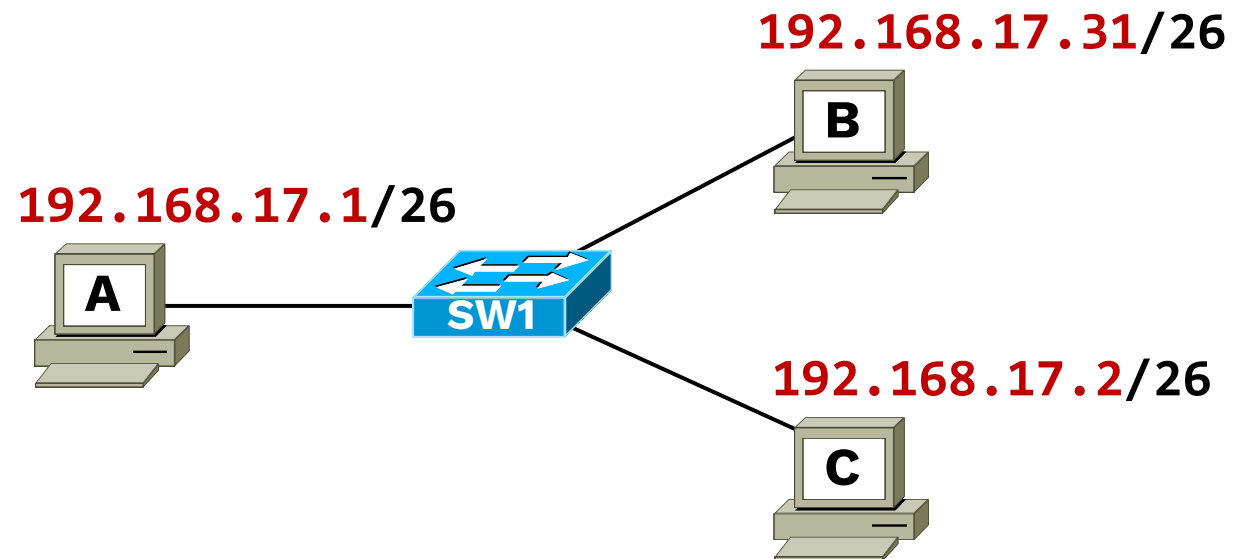
# Exercițiul 1: Verificarea configurației

- Stațiile sunt configurate cu IP-urile și măștile din figură. Există vreo problemă cu această configurație?
  - R: Da; A are configurată o adresă de rețea și C are configurată o adresă de broadcast; în plus, C are o mască de rețea diferită de A și B



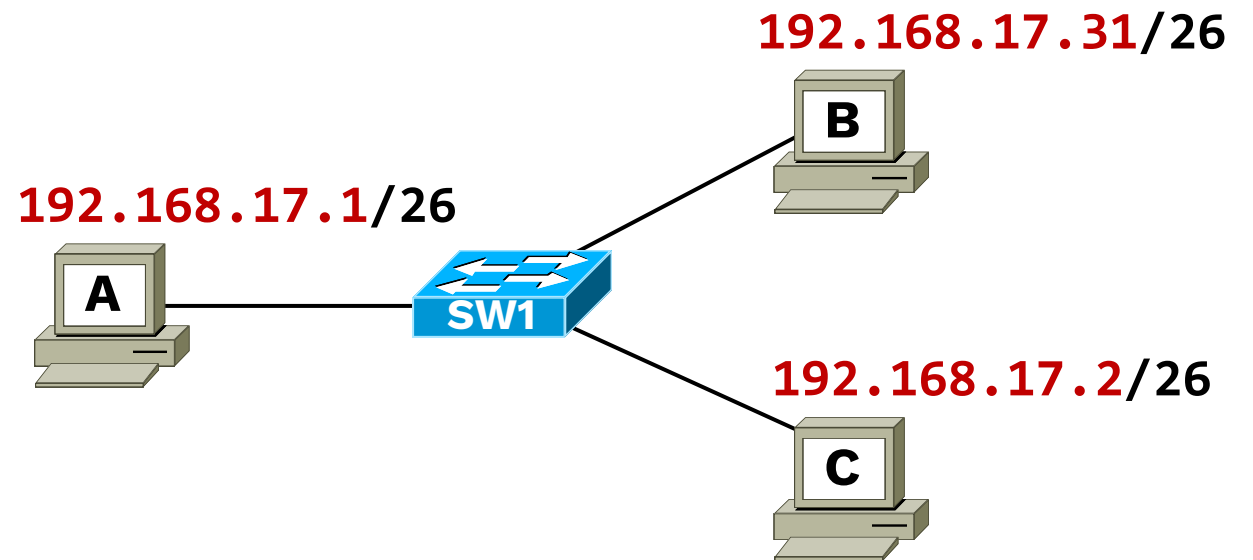
# Exercițiul 2: Broadcast A

- Adresele IP greșite au fost corectate. **A** dă un broadcast. Ce adrese IP sursă și destinație vor fi incluse în antetul IP?
- Care este adresa de rețea a lui **A**?



# Exercițiul 2: Broadcast A

- Adresele IP greșite au fost corectate. **A** dă un broadcast. Ce adrese IP sursă și destinație vor fi incluse în antetul IP?
  - R: Sursă: **192.168.17.1**;  
destinație: **192.168.17.63**
- Care este adresa de rețea a lui **A**?
  - R: **192.168.17.0**





# Clase de adrese

- Adresele IP au fost istoric clasificate în 5 clase de adrese (**A**, **B**, **C**, **D** și **E**), fiecare cu o mască specifică
- Inițial dispozitivele luau în considerare aceste clase pentru a determina masca rețelei
- IANA atribuia unei organizații un întreg bloc classful de adrese, însă cele de clasa **A** erau deseori prea mari și cele de clasa **C** prea mici
- În rețelele moderne clasele de adrese nu mai sunt relevante

# Clase de adrese

- Clasele sunt identificate după primii biți ai primului octet

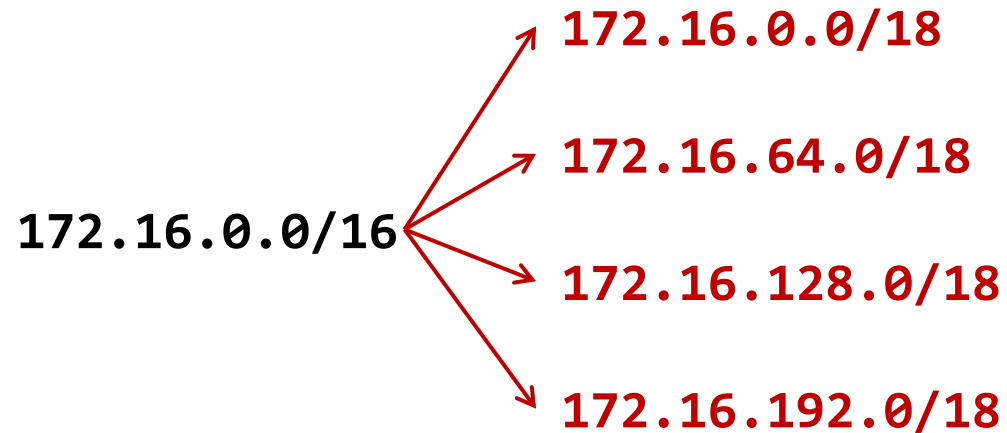
Clasă	Primul octet	Gama de adrese	Mască	Scop
A	0...	0.0.0.0 – 127.255.255.255	/8	-
B	10...	128.0.0.0 – 191.255.255.255	/16	-
C	110...	192.0.0.0 – 223.255.255.255	/24	-
D	1110...	224.0.0.0 – 239.255.255.255	-	Multicast
E	1111...	240.0.0.0 – 255.255.255.255	-	Experimental

# Adrese publice și private

- Pentru a economisi adrese, RFC1918 a alocat trei spații de adrese pentru rețele private:
  - **10.0.0.0/8 - 10.255.255.255/8**
  - **172.16.0.0/12 - 172.31.255.255/12**
  - **192.168.0.0/16 - 192.168.255.255/16**
- Adresele private nu pot fi atribuite unei organizații și nu pot fi folosite în Internet
- Pentru a conecta o stație cu adresă privată la Internet aceasta trebuie translatată la o adresă publică, proces numit **NAT** (Network Address Translation)

# Subnetare

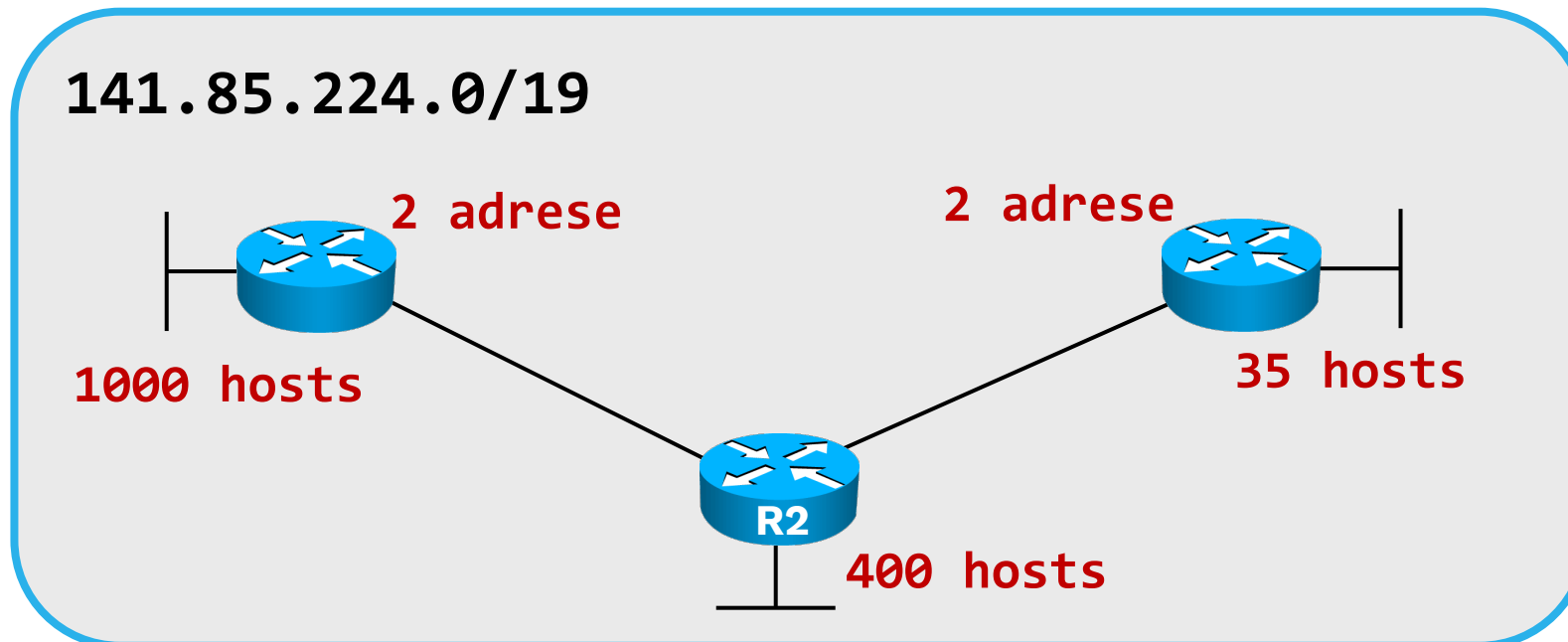
- Istoric, un **subnet** reprezenta o rețea obținută prin deplasarea la dreapta a unei măști de rețea clasf:



- Retelele actuale au abandonat ideea de rețele classful și folosesc **VLSM** (Variable Length Subnet Mask); în acestea un subnet nu este cu nimic diferit de o rețea.

# Subnetare

- O definiție actuală pentru subnet ar putea fi orice rețea ce face parte din spațiul de adresă a unei rețele mai mari
- Procesul de subnetare (**subnetting**) constă în a împărți o rețea mai mare în mai multe rețele ce respectă un set de cerințe



# Subnetare

- Înțelegerea procesului de subnetare ne ajută să răspundem la întrebările:
  - Este blocul de adrese cumpărat suficient pentru cerințele organizației?
  - Putem organiza rețelele astfel încât să fim pregătiți pentru extinderea numărului de stații?
  - Este necesară o atribuire optimă a spațiilor de adresă sau este suficientă împărțirea egală între departamente?
  - Putem optimiza tabelele de rutare dacă avem o rețea mare?
- Există două tipuri de subnetare:
  - În subnet-uri egale
  - Optimă (cu pierdere minimă de adrese)

# Subnetare

- Exemplu: Să se subneteze spațiul de adrese **192.168.10.0/24** pentru a acomoda trei rețele având **60**, **30** respectiv **15** stații. Subrețelele obținute să fie egale ca dimensiune.
  - Avem nevoie de 3 subrețele deci trebuie împrumutați **2 biți** pentru partea de subrețea a adresei IP

**Rețeaua de subnetat:**    192   .   168   .   10   .   0   /24

**Primul subnet:** 11000000 . 10101000 . 00001010 . 00000000 /26

**Al doilea subnet:** 11000000 . 10101000 . 00001010 . 01000000 /26

**Al treilea subnet:** 11000000 . 10101000 . 00001010 . 10000000 /26

# Subnetare

Rețeaua de subnetat: 192 . 168 . 10 . 0 /24

Primul subnet: 11000000 . 10101000 . 00001010 . 00000000 /26

Al doilea subnet: 11000000 . 10101000 . 00001010 . 01000000 /26

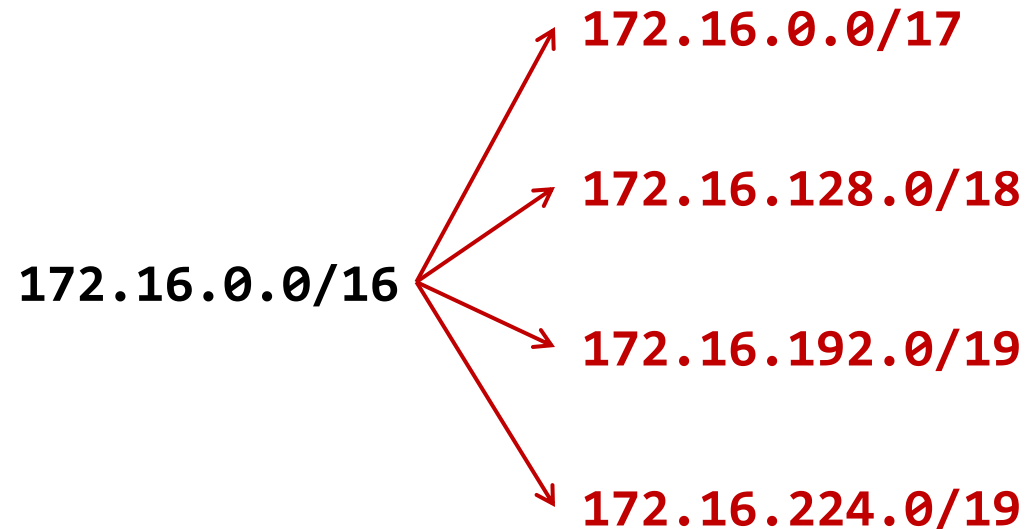
Al treilea subnet: 11000000 . 10101000 . 00001010 . 10000000 /26

- Cerințele de subrețele erau de **60**, **30** și **15** stații. Sunt suficient de mari subrețelele obținute?
  - R: Da. Necesarul este de **6**, **5**, respectiv **5** biți de stație. De ce sunt 5 biți necesari pentru ultima subrețea?
- Cât de multe adrese IP de stații au fost pierdute?
  - R: **62 - 60 = 2**; **62 - 30 = 32**; **62 - 15 = 47**; Total: **81**



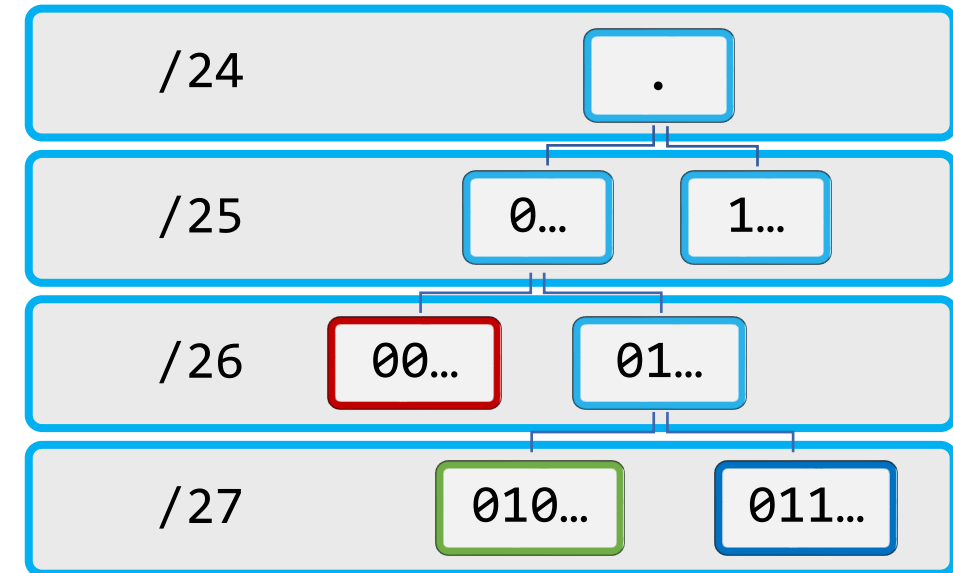
# VLSM

- Putem reduce pierderea de adrese folosind subnetare bazată pe **VLSM**
- VLSM permite crearea de subnet-uri ce nu mai au măști de aceeași lungime

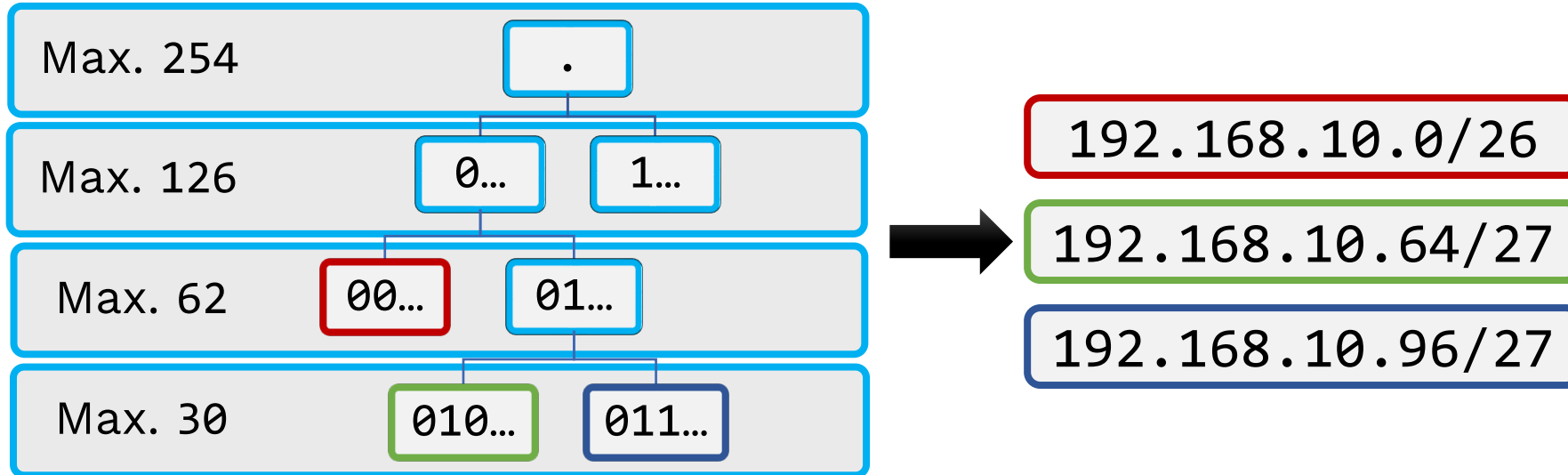


# Subnetare

- Reluăm exemplul anterior: Să se subneteze spațiul de adrese **192.168.10.0/24** pentru a acomoda trei rețele având **60**, **30** respectiv **15** stații. Subnetarea să risipească un număr minim de adrese.
  - Se observă că pentru cele trei rețele avem nevoie de **6**, **5** respectiv **5** biți de host
  - Putem reprezenta arborescent divizarea ierarhică a ultimului octet:



# Subnetare



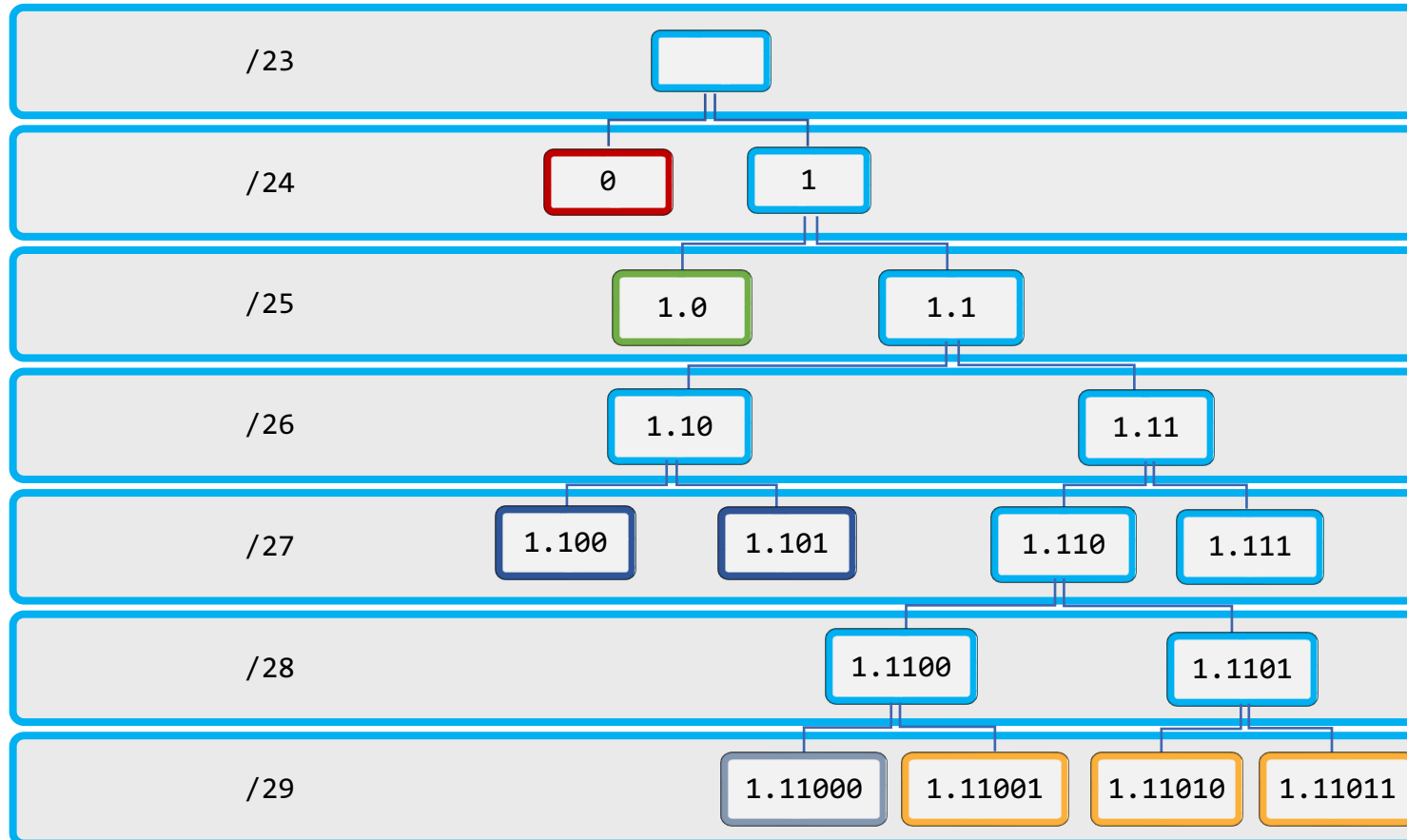
- Cât de multe adrese IP de stații au fost risipite?
  - R:  $62 - 60 = 2$ ;  $30 - 30 = 0$ ;  $30 - 15 = 15$ ; Total: 17

# Exercițiu

- Să se subneteze optim spațiul de adrese **172.18.240.0/23** astfel încât să fie acomodate cerințele:
  - O rețea cu **200** de host-uri
  - O rețea cu **90** de host-uri
  - Două rețele cu **20** de host-uri
  - O rețea cu **6** host-uri
  - Trei rețele cu **4** host-uri

# Exercițiu

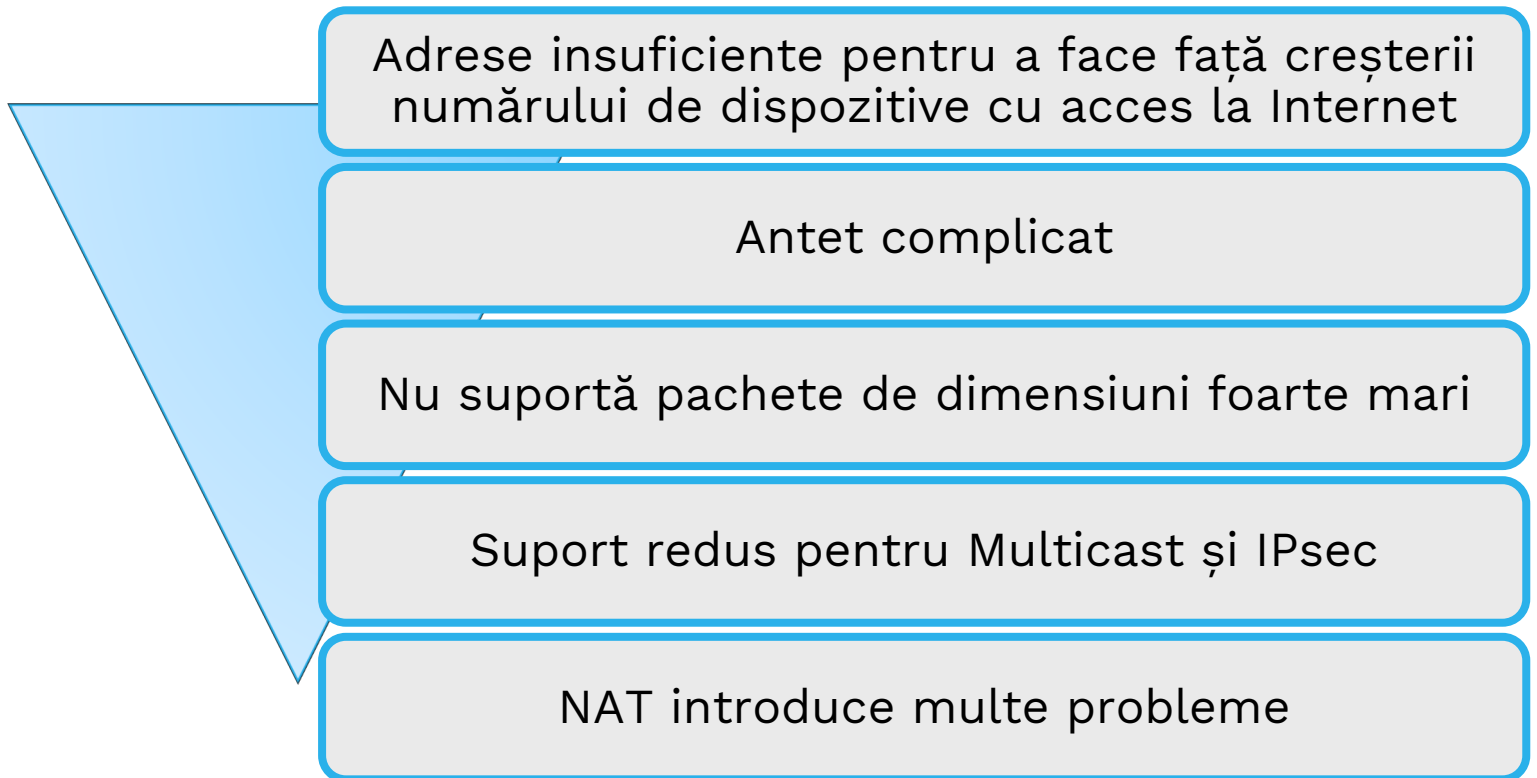
- Cerințe: 200; 90; 20; 20; 6; 4; 4; 4



# Exercițiu

- R:
  - **172.18.240.0/24**
  - **172.18.241.0/25**
  - **172.18.241.128/27**
  - **172.18.241.160/27**
  - **172.18.241.192/29**
  - **172.18.241.200/29**
  - **172.18.241.208/29**
  - **172.18.241.216/29**

# Dezavantaje IPv4



Adrese insuficiente pentru a face față creșterii numărului de dispozitive cu acces la Internet

Antet complicat

Nu suportă pachete de dimensiuni foarte mari

Suport redus pentru Multicast și IPsec

NAT introduce multe probleme

# ARP

Antet

Proxy ARP



**RL**

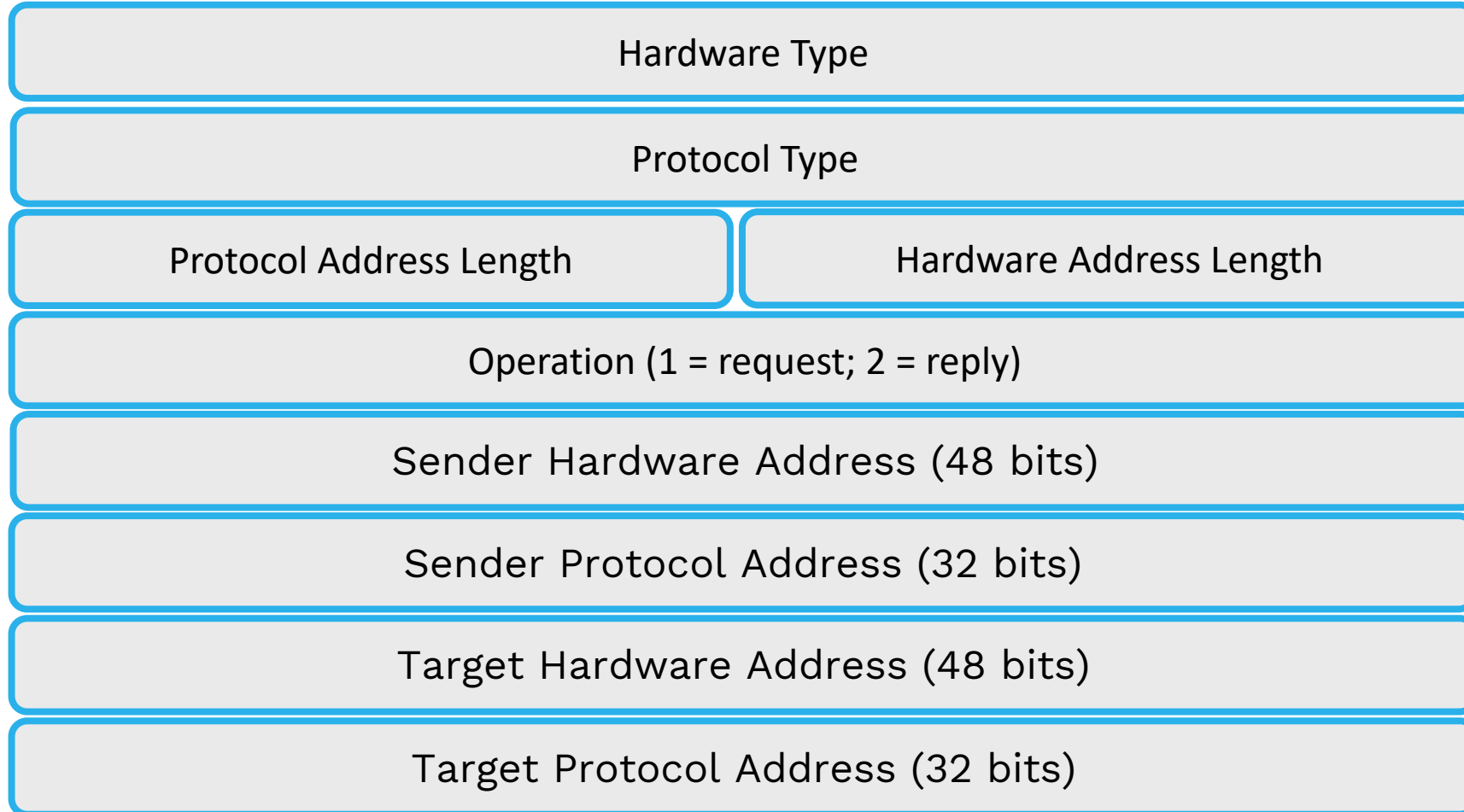
crunch it connected



# ARP

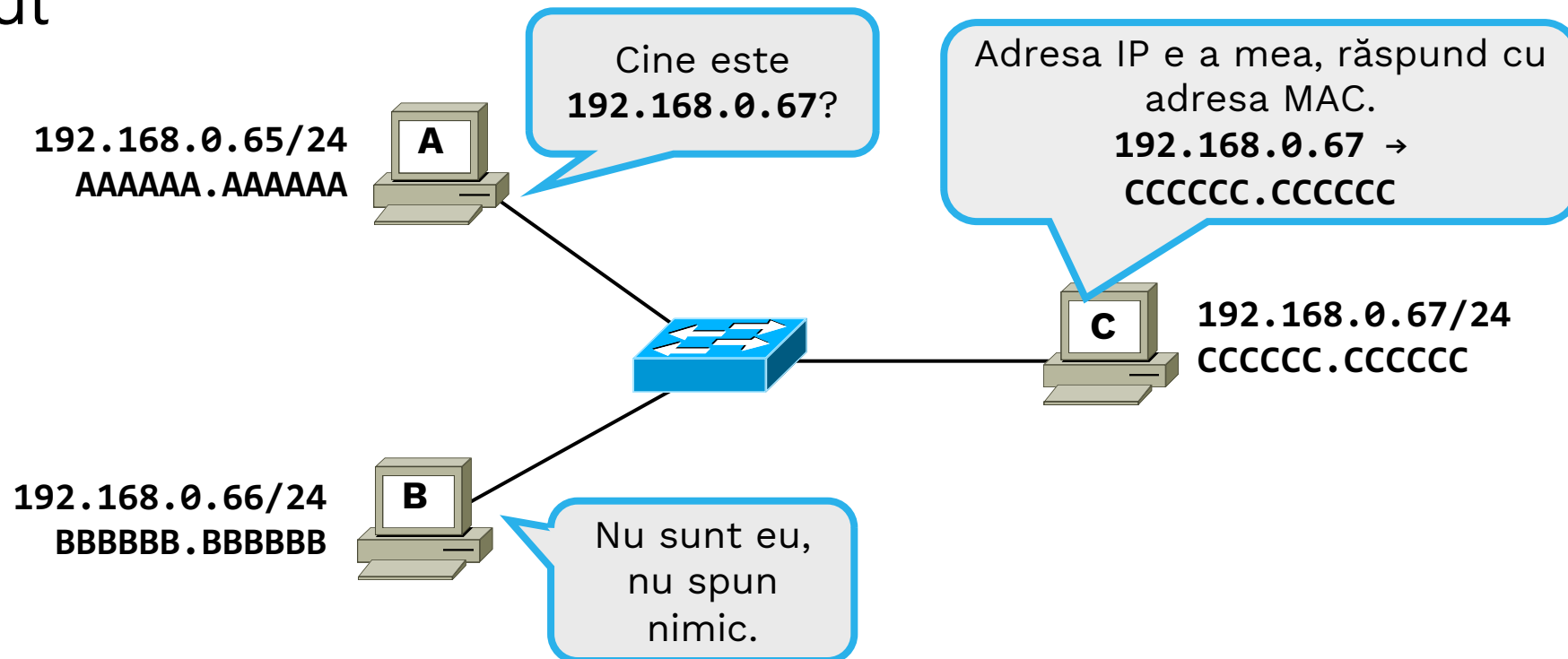
- Când o stație vrea să trimită un pachet într-o rețea Ethernet, aceasta dispune de adresa IP dar nu și de adresa MAC
- Pentru a putea transmite cadrul și a fi acceptat la destinație este necesară determinarea acestei adrese
- Protocolul care determină adresa MAC pornind de la adresa IP poartă numele de ARP (Address Resolution Protocol)

# Formatul cadrului



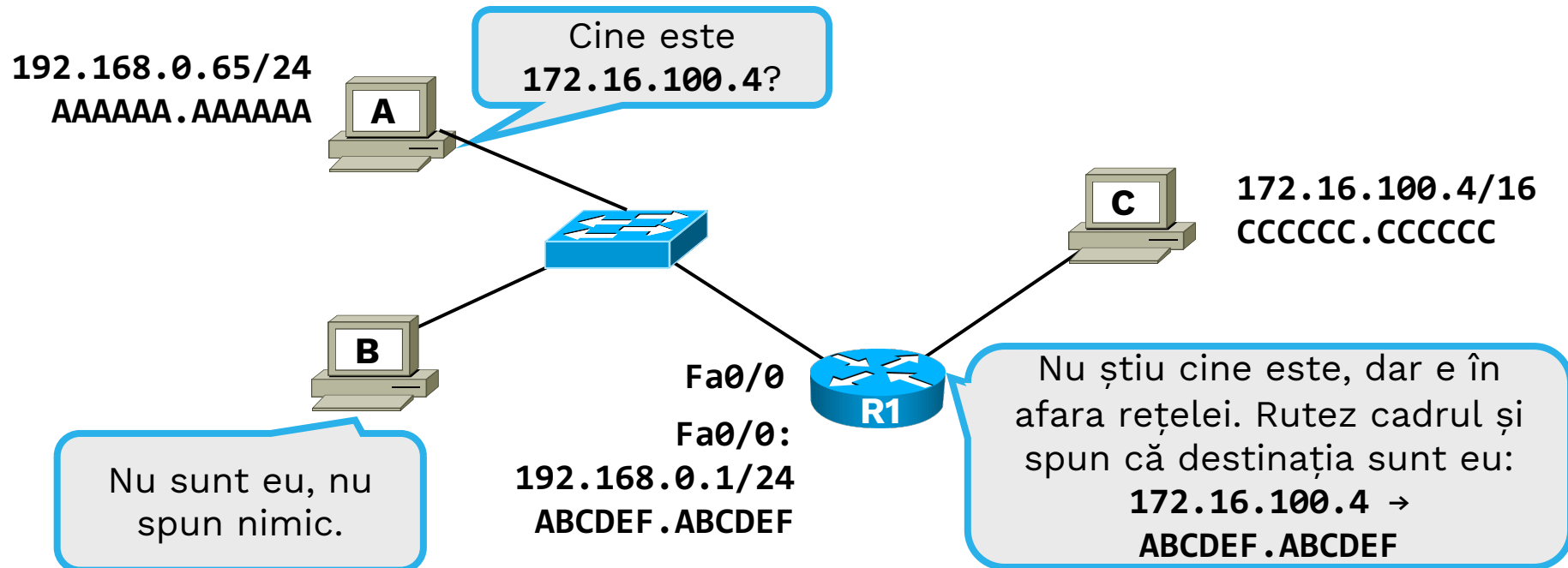
# Exemplu ARP

- Inițial emițătorul dă un mesaj la adresa MAC **FFFFFF.FFFFFFFF** și adresa IP a destinației în care cere adresa MAC
- Doar stația cu IP-ul respectiv va răspunde, restul vor ignora mesajul



# Proxy ARP

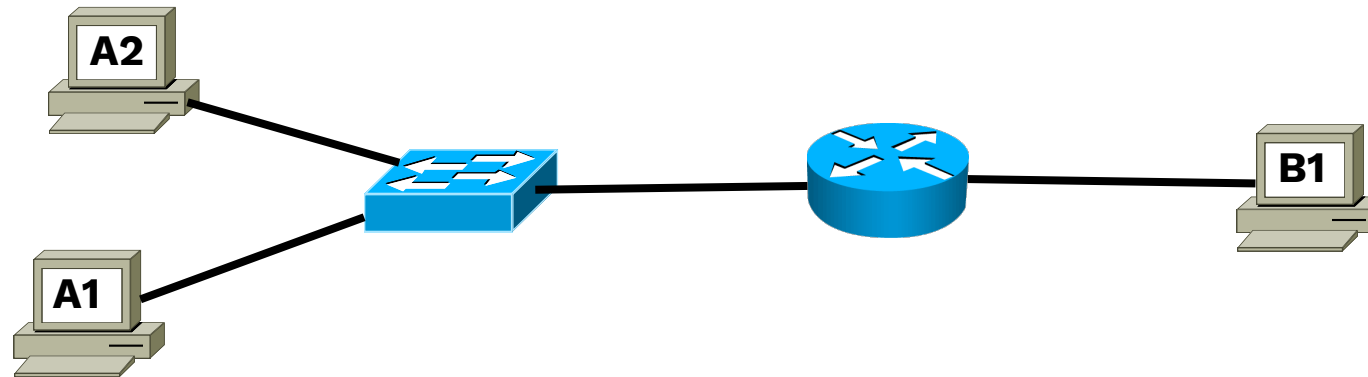
- Tehnică de ARP
- Ruterul răspunde cu propria sa adresă MAC pentru o adresă IP aflată în afara rețelei emițătorului



# Default gateway

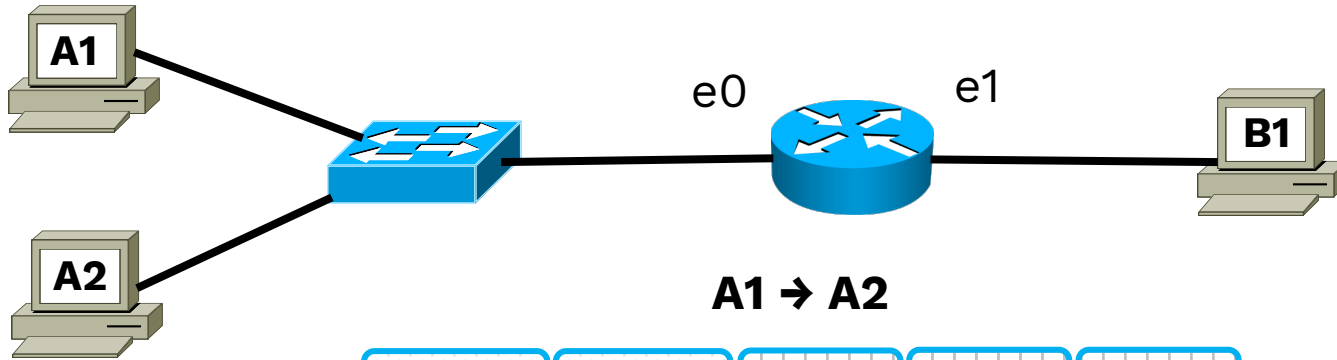
- Stația sursă verifică dacă destinația se află în aceeași rețea
- Dacă da, cererea ARP va conține adresa IP destinație
- Dacă nu, cererea ARP va conține adresa IP a default gateway-ului
- Ce se întâmplă în cazul în care sursa nu știe adresa default gateway-ului?
  - R: Va trimite un cadrul ARP de broadcast la care îi va răspunde ruter-ul de la ieșirea din rețea doar dacă are serviciul de proxy ARP activat.

# Tabele ARP

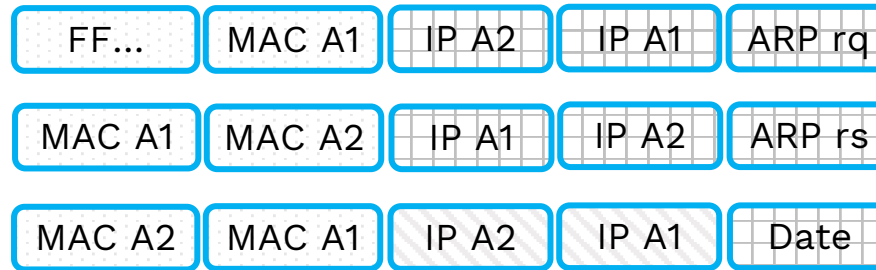


- Un ruter va avea câte o tabelă ARP pe fiecare interfață multiacces activă.
- Câte tabele ARP are un switch? De ce?
  - R: 0

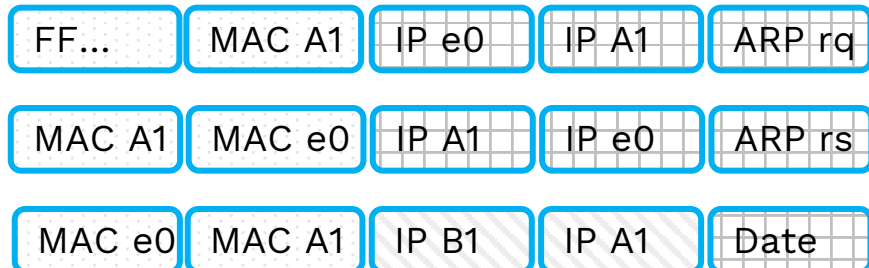
# ARP - Exemflu



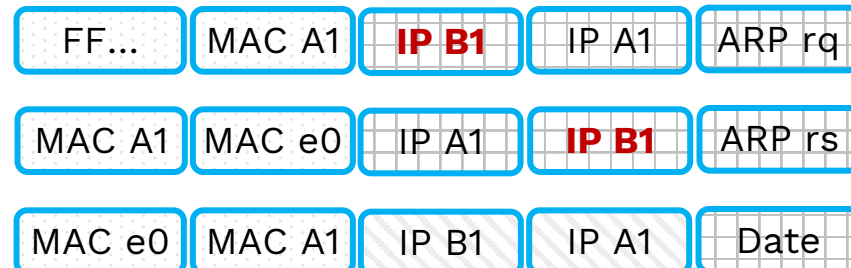
**A1 → A2**



**A1 → B1 (default gateway)**



**A1 → B1 (proxy ARP)**



# DHCP

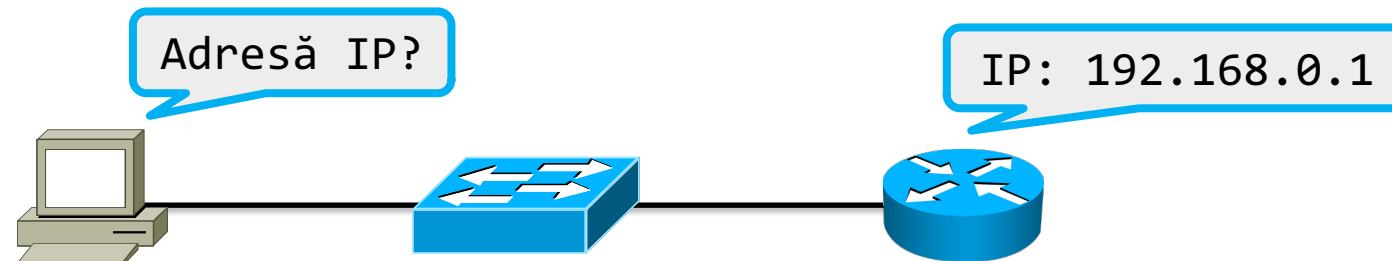
- Funcționare
- DHCP Relay



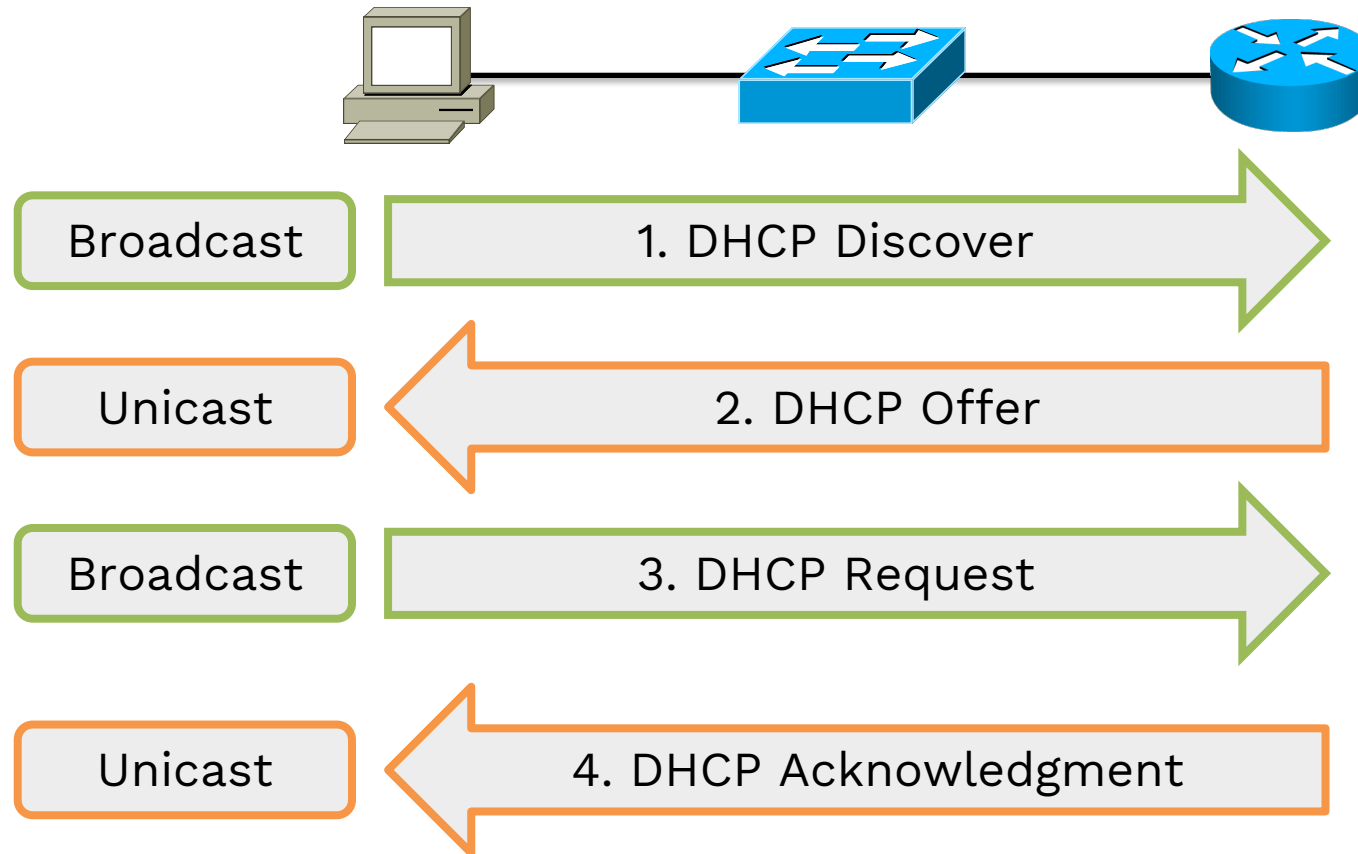


# DHCP

- Dynamic Host Configuration Protocol
- Folosit de o stație pentru a-și determina automat adresa IP
- Este necesar un server DHCP
  - Acesta poate fi un ruter sau un calculator dedicat din rețea
- De ce este util DHCP?



# DHCP



# DHCP – 1. Discovery

1. Discovery

2. Offer

3. Request

4. Ack

- Clientul trimite un broadcast UDP pe rețeaua locală
- Serverele DHCP din rețea au configurate **DHCP pools** care reprezintă de fapt seturi de adrese ce pot fi asignate clienților
- La primirea unui **DHCP discover**, fiecare server rezervă pentru clientul respectiv o adresă IP
- Pe un server pot fi configurate mai multe **DHCP pools**; rețeaua din care va fi alocată adresa este aleasă în funcție de IP-ul interfeței pe care s-a primit cererea

# DHCP – 2. Offer

1. Discovery

2. Offer

3. Request

4. Ack

- După rezervarea IP-ului, serverul trimite un răspuns unicast clientului
- Răspunsul trebuie să conțină următoarele câmpuri:
  - Adresa MAC a clientului
  - Adresa oferită de server
  - Masca de rețea a adresei
  - Durata lease-ului
  - Adresa serverului de DHCP
- **Lease-ul** reprezintă durata de timp pentru care adresa IP este rezervată clientului

# DHCP – 3. Request

1. Discovery

2. Offer

3. Request

4. Ack

- Clientul trimite un broadcast pentru a spune dacă oferta este acceptată
- Clientul știe adresa IP a serverului. De ce este necesar un mesaj de broadcast?
  - R: Pot exista multiple servere DHCP în rețea. Toate trebuie informate de alegerea clientului pentru a putea elibera adresele rezervate în primele două faze.

# DHCP – 4. Acknowledgment

1. Discovery

2. Offer

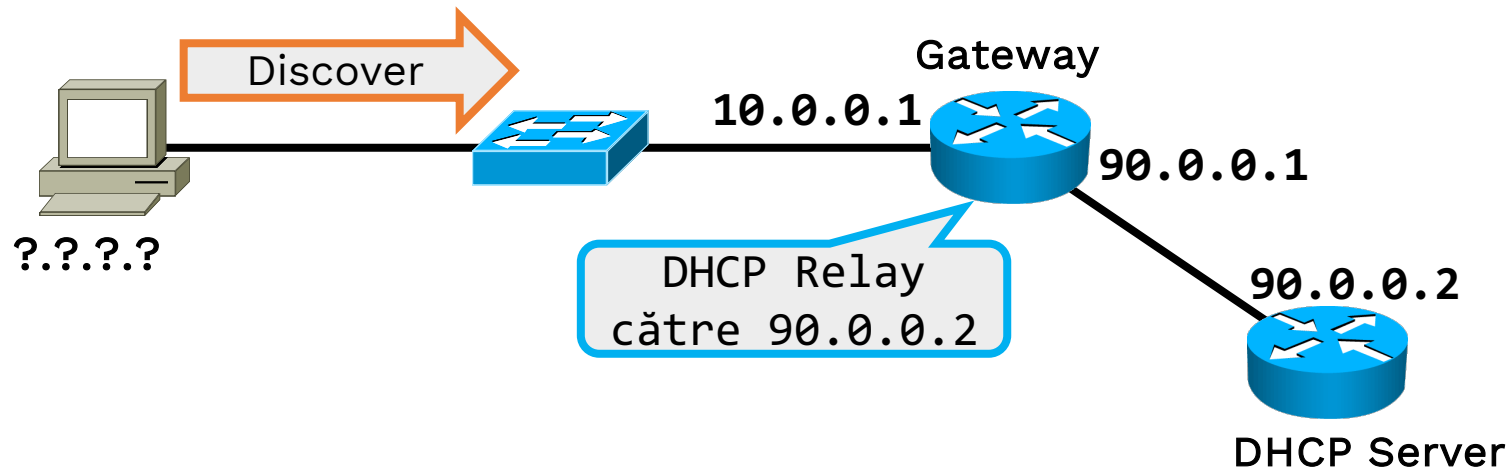
3. Request

4. Ack

- Serverul îi transmite clientului că procesul s-a încheiat și adresa i-a fost atribuită pe durata lease-ului
- Dacă lease-ul se apropie de expirare, clientul poate cere o prelungire
- Există posibilitatea ca la expirare clientul să ceară adresa pe care a avut-o înainte
  - De ce este utilă păstrarea adresei?
- În Ack pot fi trimise și alte informații cerute de client:
  - Default gateway
  - Servere DNS

# DHCP relay

- Există situații în care serverul DHCP nu este în rețeaua locală
- Deoarece mesajul este un broadcast către 255.255.255.255 acesta nu poate fi transmis în alte rețele
- Redirecțarea unei cereri DHCP se poate face prin configurarea DHCP Relay pe ruterul din rețeaua locală
- Cererea DHCP va fi redirecțată către IP-ul serverului de DHCP din altă rețea



# Sumar

