



5. 10. 2022

Curs 1

Lab: 40%

Examen: 60% { minim 5 la ambele

↓
grile și moodle

www.cs.ubbcluj.ro ~ buflng

Retele de calculatoare - Tannenbaum

Tipuri de trafic

- Socket-uri : mecanism de comunicare client - server

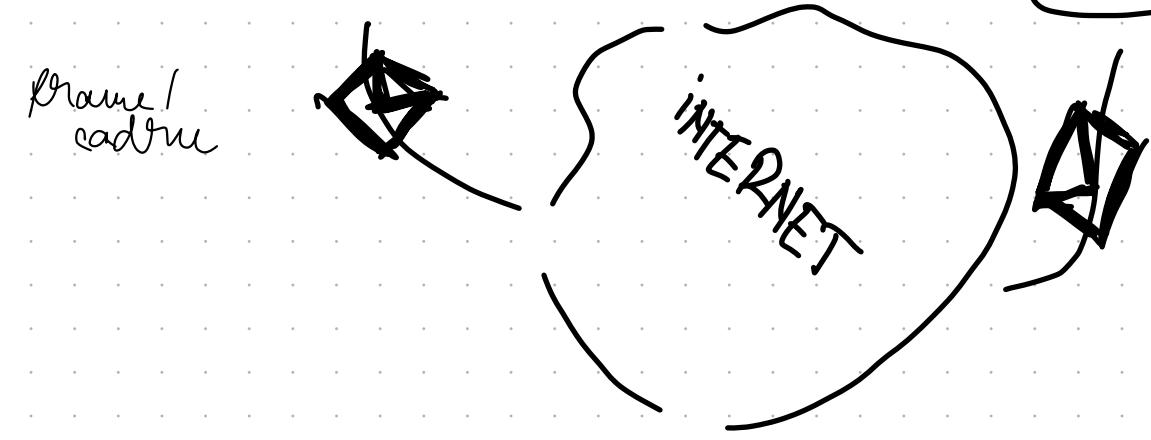
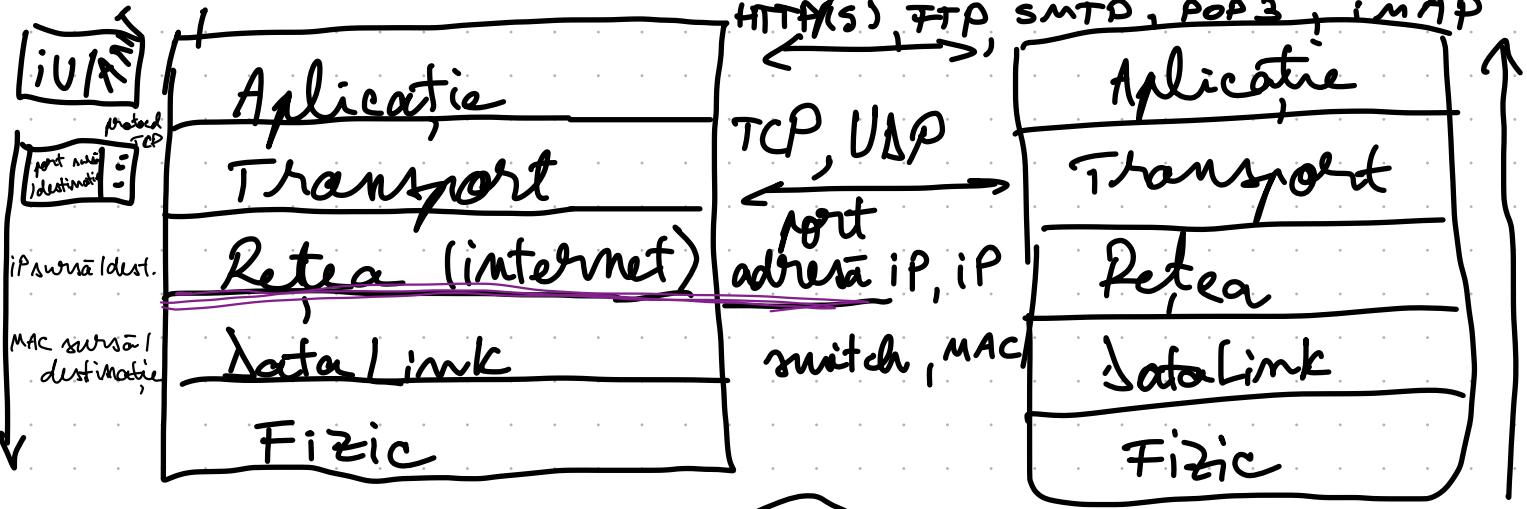
Stiva TCP/IP

- Principii de bază la funcționarea rețelei internet
 - direcțarea (căutarea) pachetelor (ruterare)
 - mecanismul de adresare IP
 - Sistemul Numelor de domenii (DNS)
(+ traducere nume calculator → adresă IP)
 - Stiva de protocole TCP/IP (RFC-uri)

Protocol = set de reguli de comunicare

STIVA TCP/IP





HTTP - 80

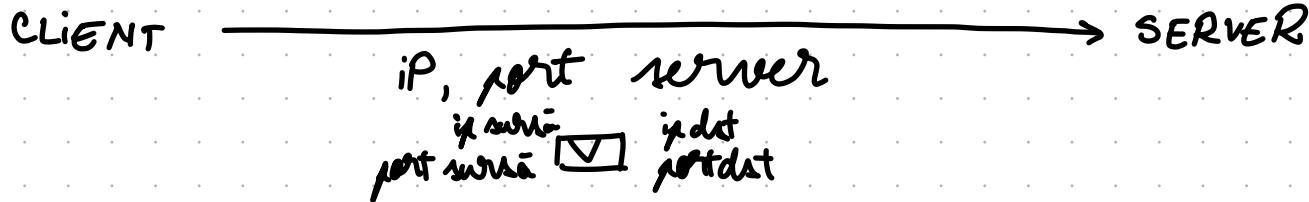
HTTPS - 443

FTP - 21

SSH - 22

Curs 2

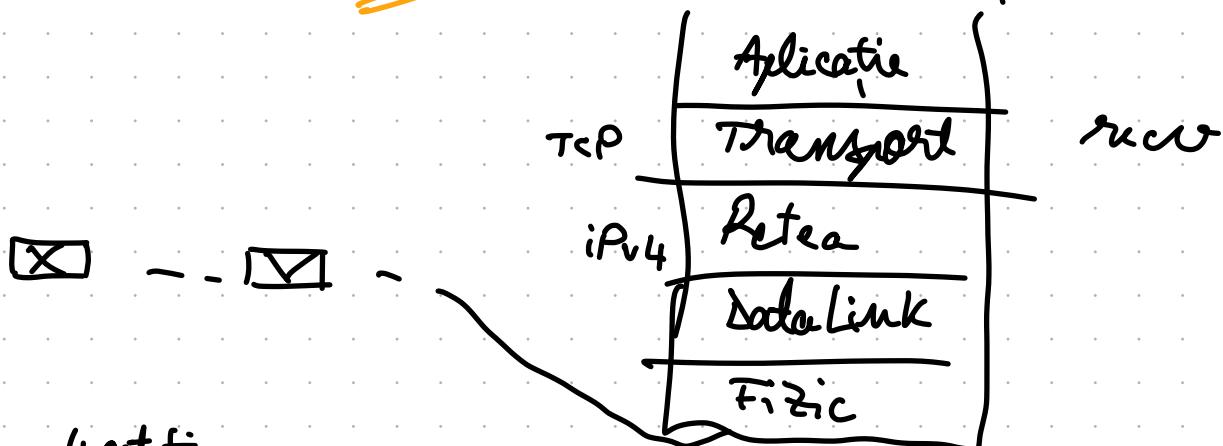
Socket -uri



IP → localizează calculatorul pe care rulează procesul (client sau server)

PORT → localizează procesul care lucrează pe calculatorul cu IP-ul dat

DOAR PROCESE NU POT OCUPA ACELAȘI PORT!!



IPv4 - 4 octeti

IPv6 - 16 octeti

CLIENT

SERVER

c=socket(

1 - socket

if(connect(c, {
 AF_INET
 connect(c, {
 ip server } , size (structura))
 - port server })

⇒ eroare la conectare la server

send (c, &i, sizeof(i), 0) ~ rewrite

recv (c, &j, sizeof(j), MSG_WAITALL) ~ read

close(c)

SERVER

s = socket

oice adresa IP

if (bind (s, { - 0.0.0.0 = INADDR_ANY }, sizeof (structura)))
atras !! la port = 0 => "port already being used"

"192.168.1.17" → 192 · 256³ + 168 · 256² + 1 · 256 + 17 · 256⁰
inet_addr - convertește ip repres. ca string
⇒ adresa ip este .pe 4 octeti

inet_ntoa - network to ascii

netstat → comunicile TCP active în sistem

listen (s, 5) → diferențiază clientul de server

accept ~ connect din client

c' = accept (s, ->client , ->portclient , sizeof (structura))

- server iterative (deserveste clientul rând pe rând)
- server concurrent (deserveste clientul în același timp)

c' - socket de legătură server \leftrightarrow client
recv(c', ...)
send(c', ...)
close(c')

Intre accept - close = deservirea clientului
close(s)

Server concurrent

```
while(1) {  
    c' = accept(1, ...)  
    if(fork() == 0) {  
        recv(c', ...)  
        send(c', ...)  
        close(c')  
        exit(0)  
    }  
}
```

I
close(1)  eventual intr-un handler
pt. SIG-INT (CTRL-C)

info 1

Atom 1

Atom 1

info 1

netstat -n , netstat -o en windows ifconfig
 netstat -luy linux ifconfig

hostnames - i
 ↓
 i more

"Hello" → l = 5 noms long

$\alpha_1 = \text{rand()} \% (l-2)$; entre 0 si $l-2 \rightarrow$ exclusif
 $(0,1,2)$
 $\alpha_2 = \text{rand()} \% (l-\alpha_1-2) + \alpha_1$ autre α_1+1 si $l-1 \rightarrow$ exclusif
 $\alpha_3 = \text{rand()} \% 5 + \alpha_2 + 1$

entre α_2+1 si $l \rightarrow$ exclusif

$\text{rand} \% (l-\alpha_2) + \alpha_2 + 1$

ok => 0

send(ok)

$\forall i (i=0, i \leq 422, ok; i++)$
 T

recv (suma)

if (suma == sumaT)

ok = 1

send (ok)

nr --

J

recv (ok) nr - 5

while (!ok && nr > 0)

input (suma)

send (suma)

recv (ok)

nr --

J

Cours 3

C → S
 IP : Port
 server

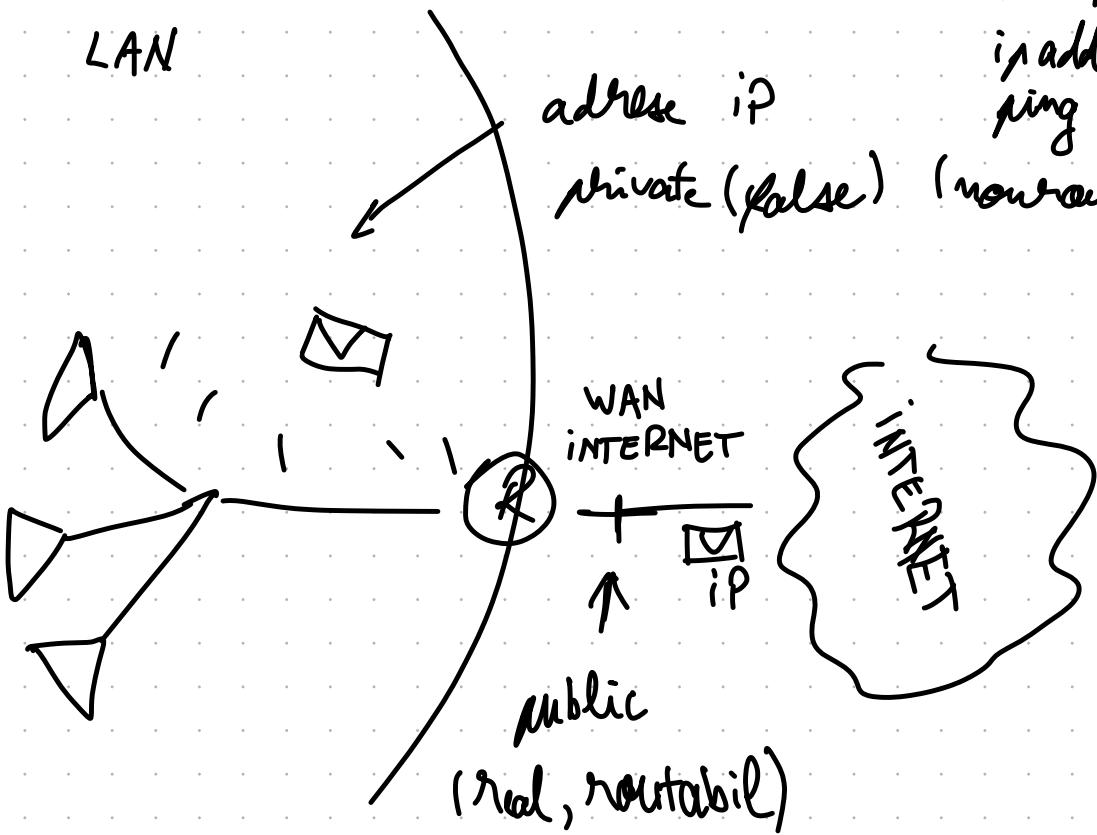
S

netstat

if (select statistics)
 if Linux mai

ifconfig windows
 ifconfig Linux
 ip addr Linux
 ping

address IP
 private (false) (nonroutable)



192.168.1.X

192.168.0.X

NAT (Network Address Translation)

SNAT (Source Network Address Translation)

C

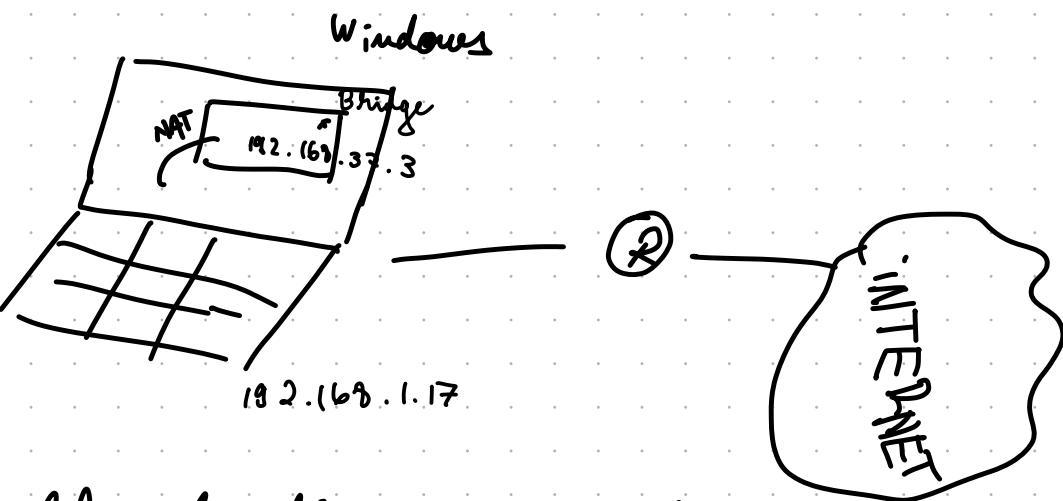
IP Client

192.168.1.17

S

IP Server

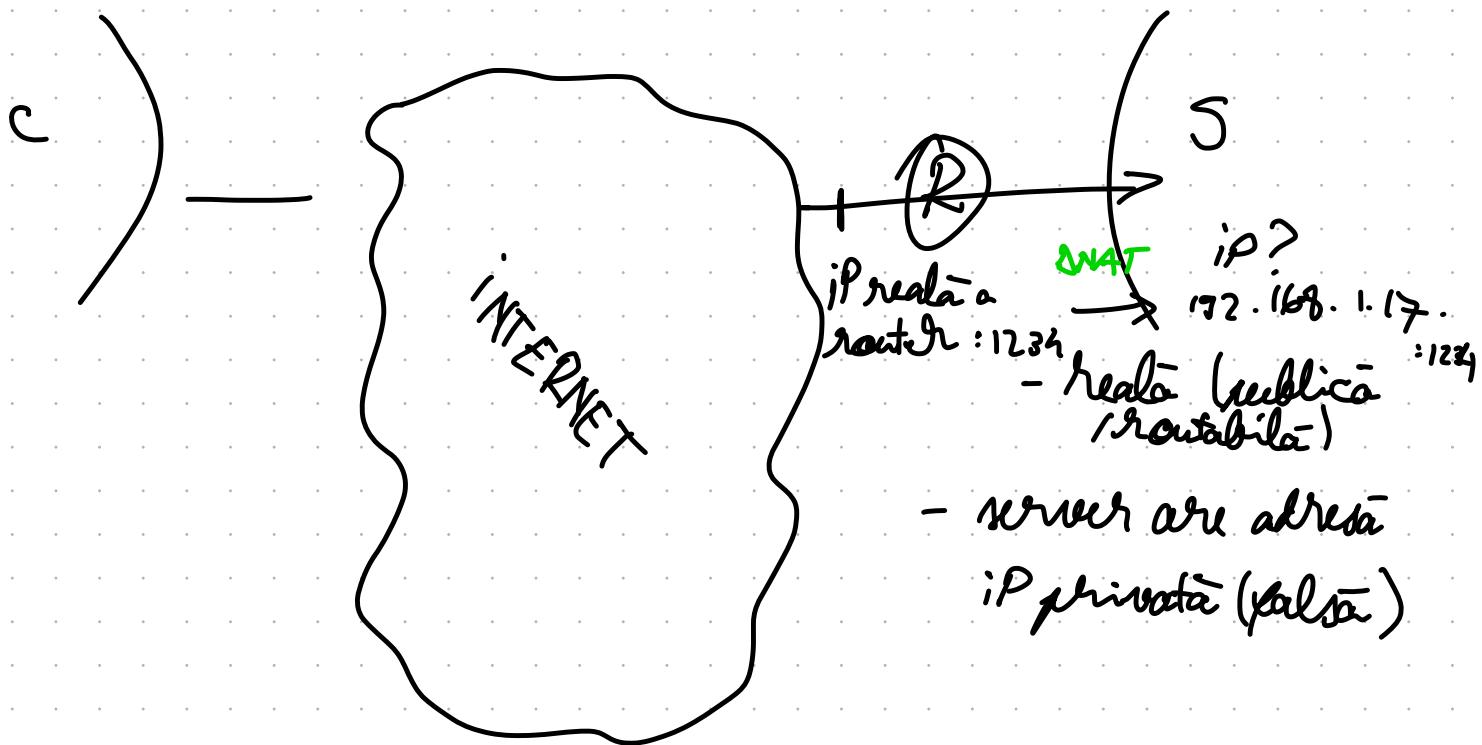
192.168.1.37



Calc. funcționează ca router și mașina virtuală.

Server linux: 172.30.0.4

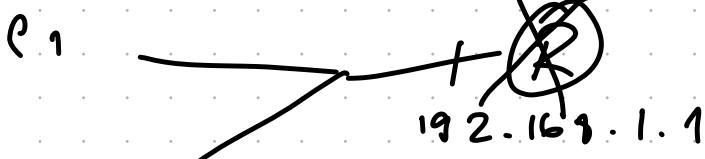
172.30.111.117



DNAT

Port Forwarding
Virtual Servers

192.168.1.2



192.168.1.3

Graedi crescente

uint16_t i, j; facto;

scanf ("%d", &i);

%hu - unsigned short

strace, ltrace

strace . / client

TCP - reliable

Curs 4

Socket-uri UDP

! conexiune ~~X~~ UDP - termen gresit

Emitator / Receptor vs Client / Server la TCP

- nu există siguranță că se primesc toate pachetele trimise
- nu se face deosebire: 1) pachet pierdut pe drum
2) nu există proces care să rețină pachete
3) datele pot ajunge în ordine diferită
- la TCP: pachetele au nr. de secvență, la UDP nu!!
(identificare + în ce ordine)
- avantajele UDP: aplicații streaming (videochat...)
→ se preferă pf. rapiditate vs reliabilitate

UDP nu e serviciu orientat conexiune

"client"

Emitator

c = socket (AF_INET, SOCK_

~~GRAM~~,

~~connect~~

sendto (c, de unde din , căt ,
memorie

"server"

Receptor

s = socket (AF_INET, ~~SOCK-~~

~~GRAM~~

bind(s, 6000)

~~listen~~

~~accept~~

```

flag, structura, sizeof(struct),
    -IP destinator
    -Port destinator

recvfrom( .. )

close(c)

sendto - trimite datele unei partenerilor
        ale cărui date de identificare sunt
        complete de recvfrom anterior

```

- Problema - e se întâmplă dacă trimitem către server din partea
mai multor procese (emittori)?
- în server nu mai există acel set de identifică clientii
prin descriptor

Soluție : în receptor se face un proces fin
care generează un port (random) și îl trimite
maiori emittorului
În fork() va avea loc deservirea emittorului

```

sendto(c, 6000)
recvfrom(c, ..  $\xleftarrow{6001}$ )

```

```
sendto(c, 6001)
```

```

bind(1, 6000)
recv(1, .. 6000)
if(fork() == 0)
    bind(1, 6001)
    recv
    sendto(1, 6001)
}

```

La TCP :

connect \Leftrightarrow accept

SYN \rightarrow

"3 way handshake"

Packet TCP

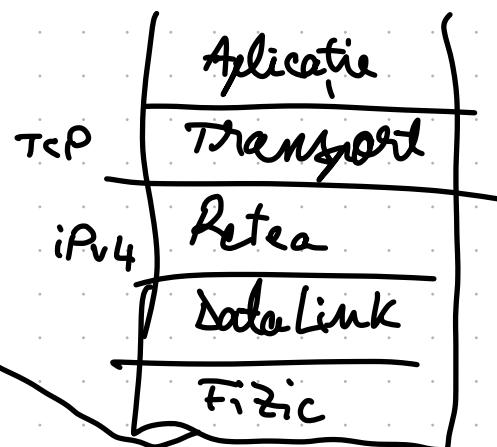
SYN, ACK

ACK



close

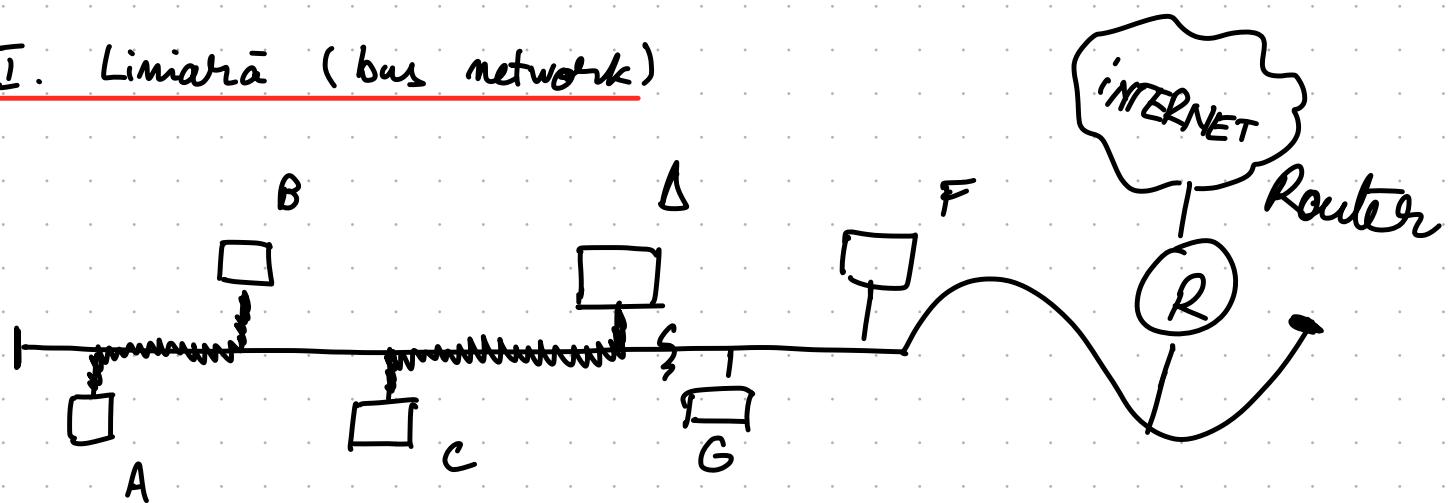
\rightarrow FIN bit finalize



Curs 5

Topologii de rețele locale

I. Liniară (bus network)



10 Mbps

BNC

Coliziuni

5

10 Mbps



"Broadcast"

Rețele locale cu difuzare → Broadcast domain

- dacă trimite A la B - mesajul se propagă pe totă lungimea rețelei! **Dezavantaj** → se reduce și viteza de comunicare

Deci: nici fiabil, nici rapid :-(

nici sigur - un utilizator malicios care poate să atrice conexiunea pt. toți userii

1 kilobit = 1000 biti

1 Megabit = 1000 Kilobiti

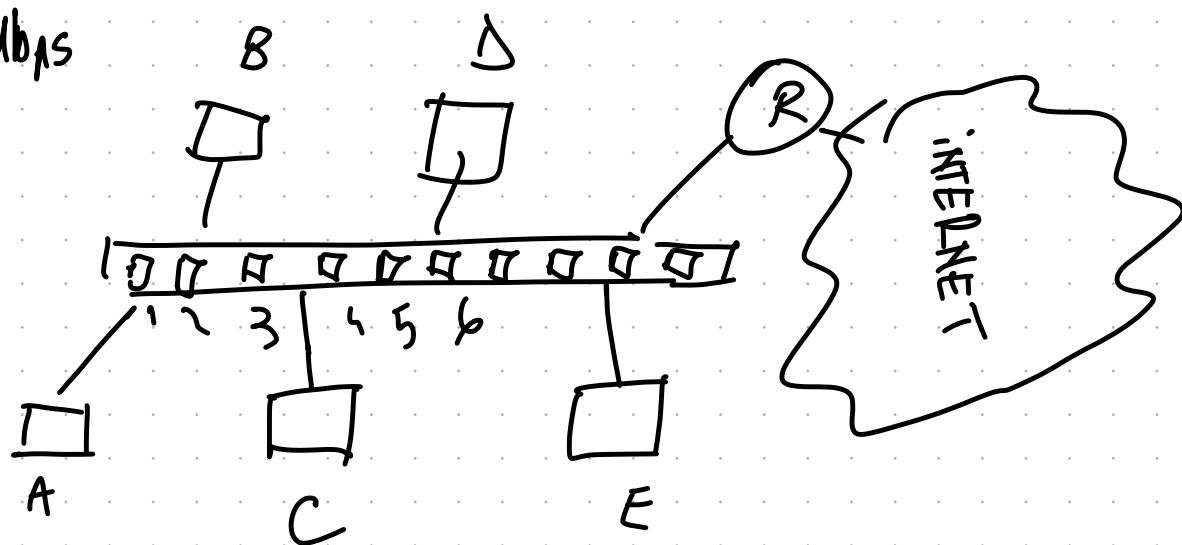
1 Gigabit = 1000 Megabiti

1 kilobyte = 10^24 bytes

1 Megabyte = 10^24 kilobytes

II Rețile cu topologie stea

Hub Switch
↳ 10Mbps



rezolvă problema fiabilității, dar se păstrează problema vitezei!

Frame: MAC surse și MAC destinație

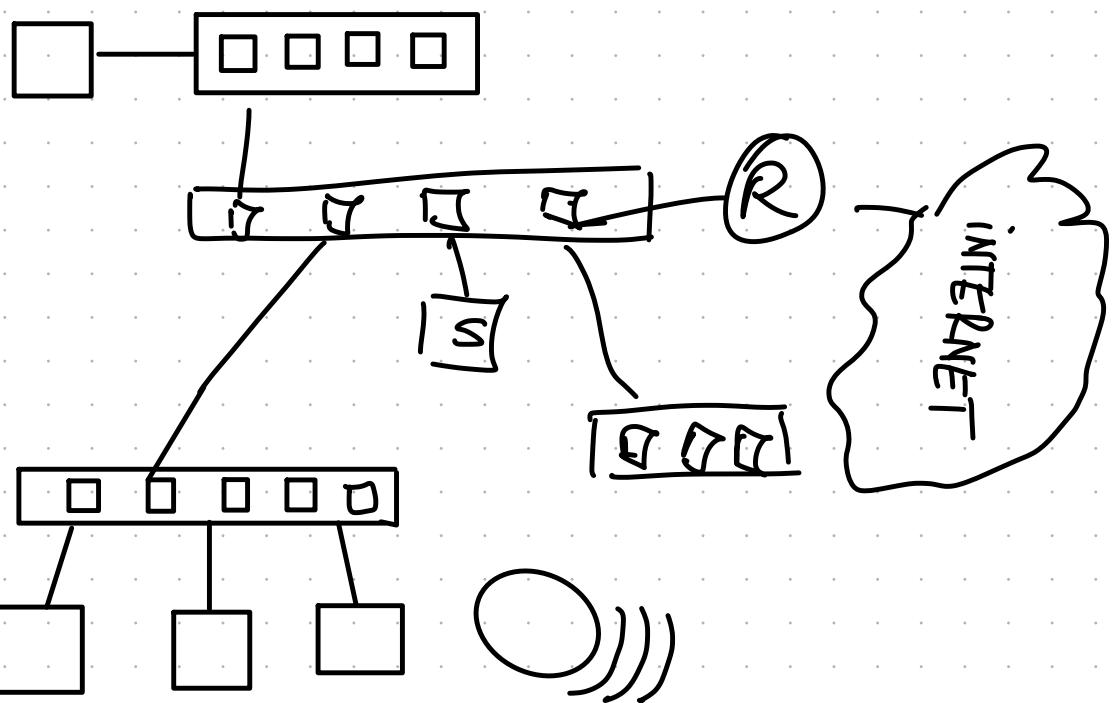
Hub vs Switch

- au procesor și memorie

- pot decide ruta pe care să trimite un frame \Rightarrow se reduce problema coliziunilor

III. Rețele cu topologie stea extinsă (extended star)

core switches



|||||||

Adrese MAC (acă Ethernet / hardware / physical)

AB : FC : 2D : SE : 14 : 7D
 hardcoded

↓
 00
 FF

Prinii 3 octeti identifică manufacturer

Ultimii 3 octeti – particular card of manufacturer

FF : FF : FF : FF : FF : FF MAC de Broadcast

ARP Address Resolution Protocol → Afflare MAC pe baza de IP

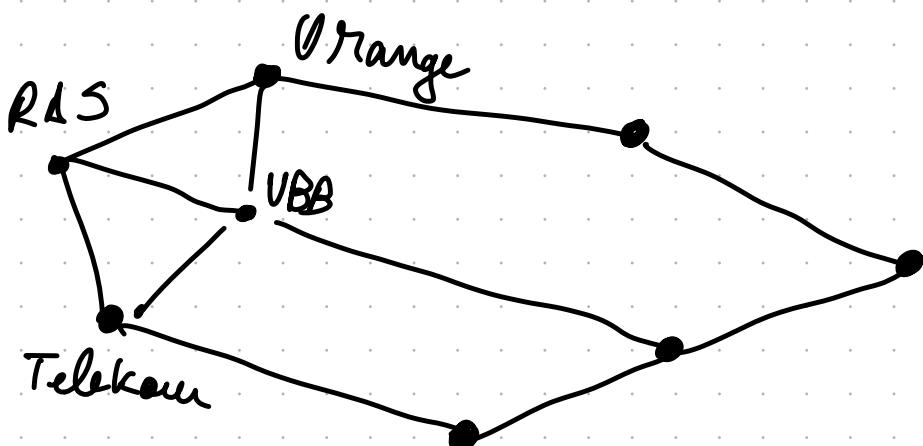
RARP Reverse ARP → Afflare IP pe baza de MAC

IV Topologie de graph incomplet

LAN - local area network

MAN - metropolitan

WAN - wide area network



STP standing trip protocol

„server“

1234

5999

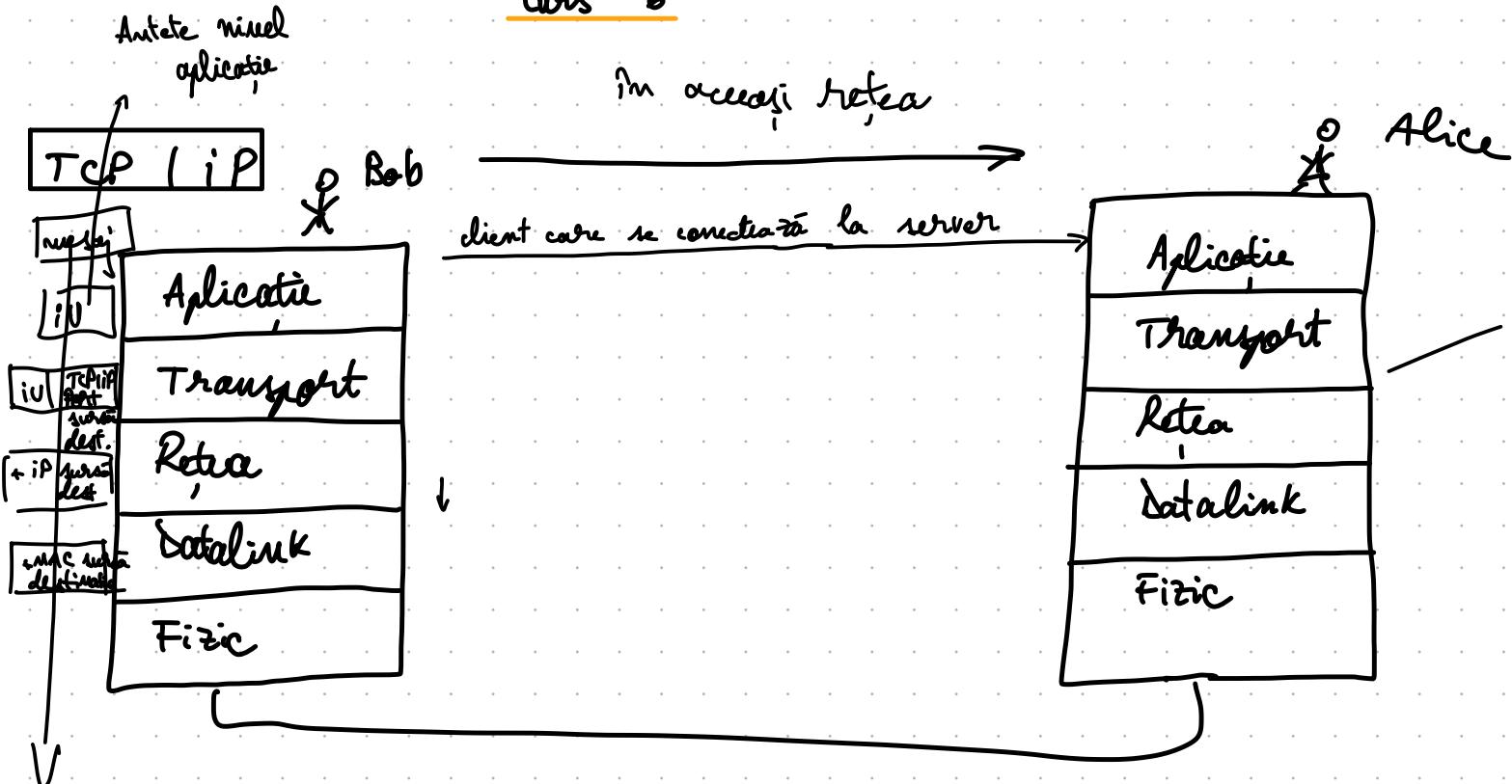
„client“

?

new = 5999

send ne 5999

Curs 6



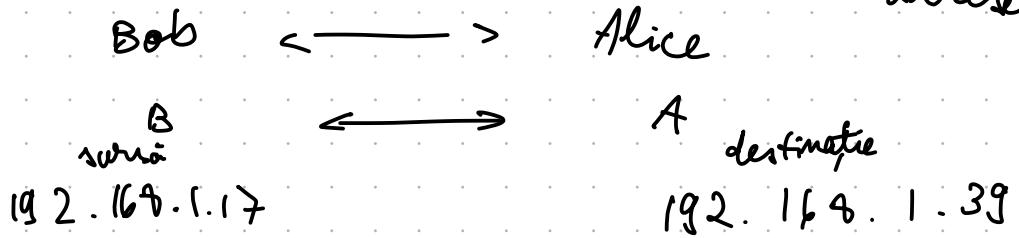
- ip
 - netmask
 - Default Gateway (adresa IP a routerului)
 - DNS (servere care translataza nume calculator → IP)
- are minima 2 adrese IP (operator + provider)

Pe Linux - în /etc/services - legătura între protocol ↔ port
(de ex. https ↔ 443 sau http ↔ 80)

- Adresele MAC sunt vizibile doar la nivelul retelei locale !!

Frame = pacet (transformat) de la nivelul Retea la nivelul Datalink

ARP (address resolution protocol) - află adresa MAC pe baza adreselor IP



ARP Request

mac dest
mac sursă = mac B
FFF : FF : FF : FF : FF

ARP reply

mac sursă = mac A

mac dst = mac B

ping - testează conexiunea între calc. din același rețea

Pe Windows arp -a și arată leg. (IP, MAC) pe care le știe

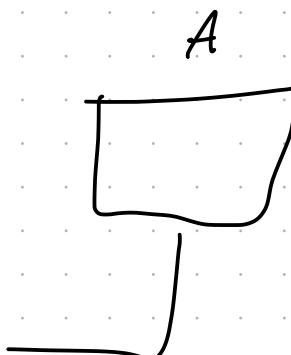
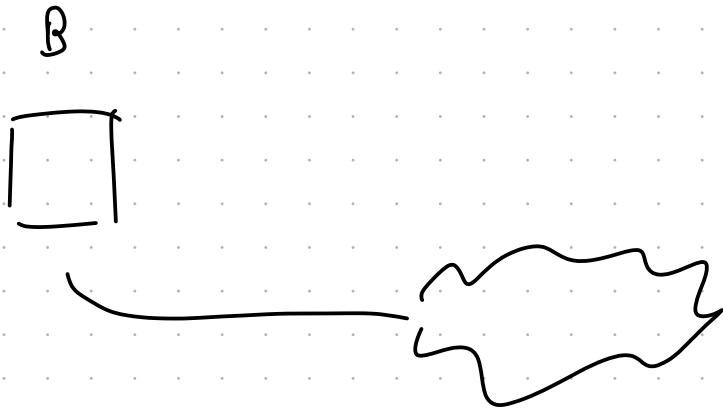
Linux arp -n calculătorul

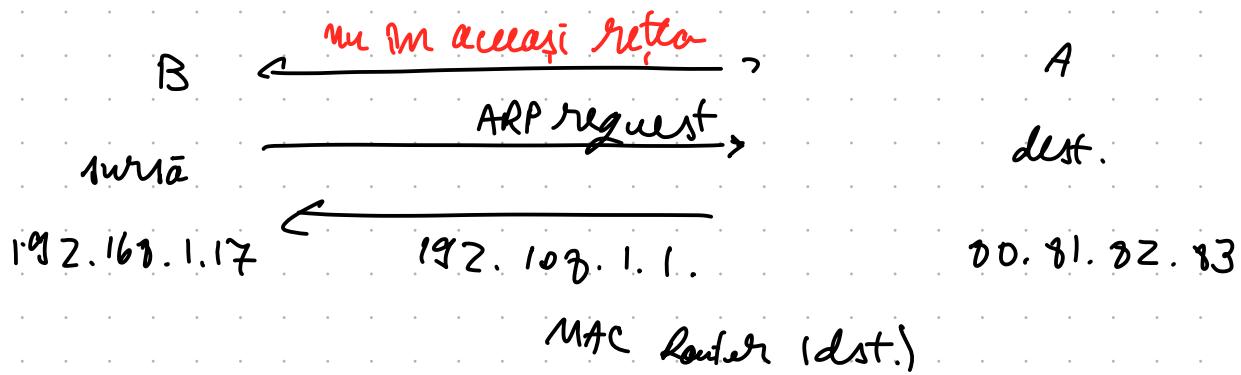
tcpdump (faza finală de pe rețea)

De la Router la Datalink - pachetul trebuie încapsulat în FRAME
(MAC sursă, MAC destinație)

Sursa nu știe MAC ip-ului destinație \rightarrow ARP request mac broadcast
(adică la toți ~~cu ip-ul 192.168.1.39~~ de la nivelul rețelei locale), înțelege doar că ce
trebuie - cu ip-ul 192...)

! MAC requests nu au sens decât în același rețea!! Nu funcționează pt. interne și dest. în rețele diferențiate





Ruter - rețea locală \longleftrightarrow rețea provider

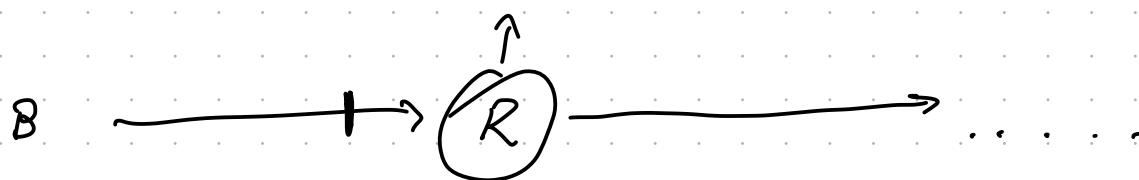
Soluție: se face ARP request la ruter

Când se dorește trimisarea de pachete în afara rețelei, ele se trimis ruterului (Default Gateway) \rightarrow avem nevoie de MAC Router

Ruter, îl călău prim ARP request

La nivel rețea: IP dest: 80.81.82.83

La nivel datalink: MAC dest: (192.168.1.1)



până când ajungem la A
(din Router în Router)

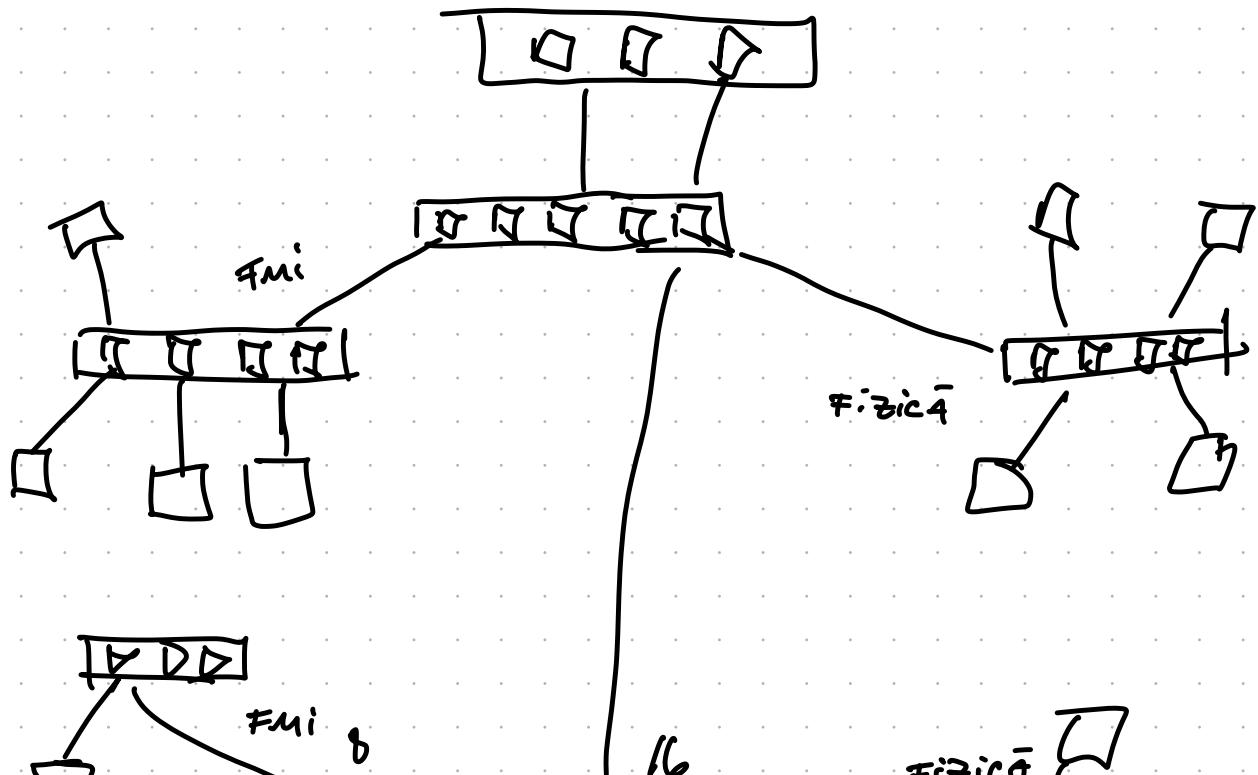
- RARP (Reverse Address Resolution Protocol) - det. adresa IP pe baza MAC

- DHCP (Dynamic Host Configuration Protocol) - obținem IP, netmask, Default Gateway, DNS ...

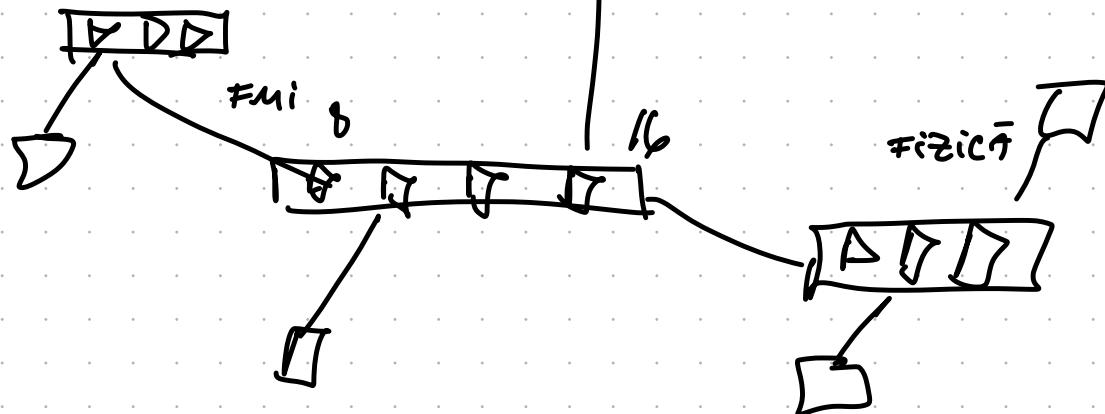
Curs 7

VLAN - virtual local area network \neq VPN (virtual private network)

Extended star topology

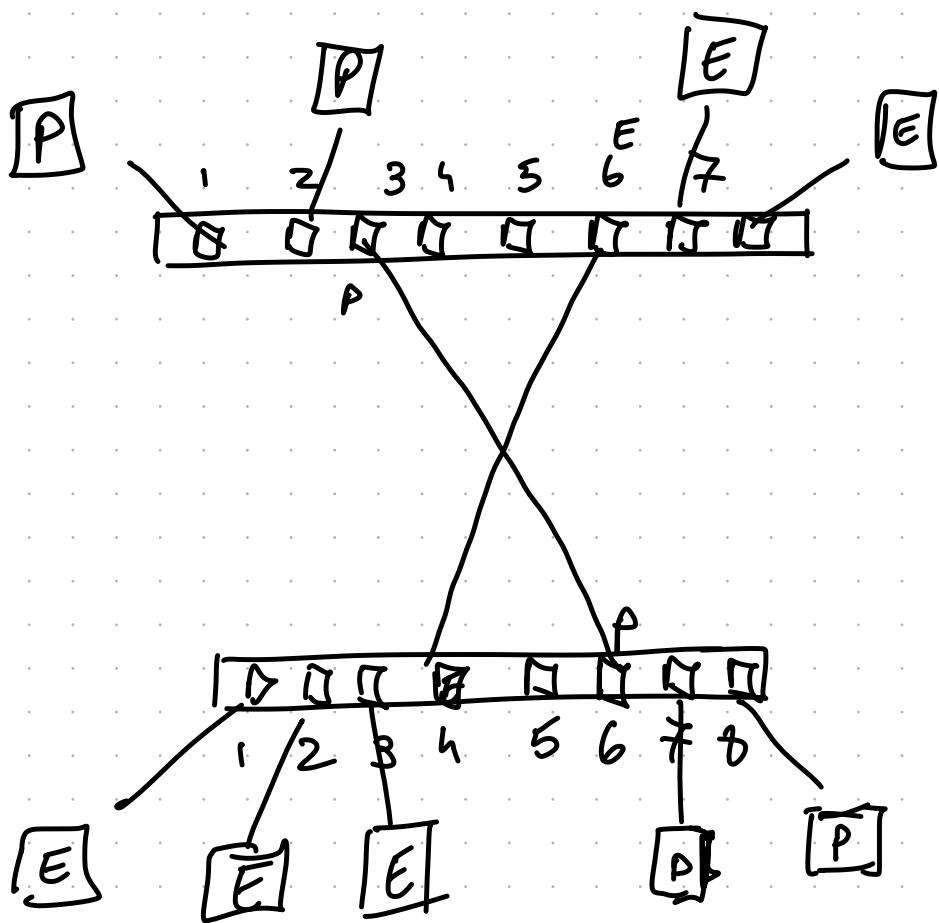


Et. 1.

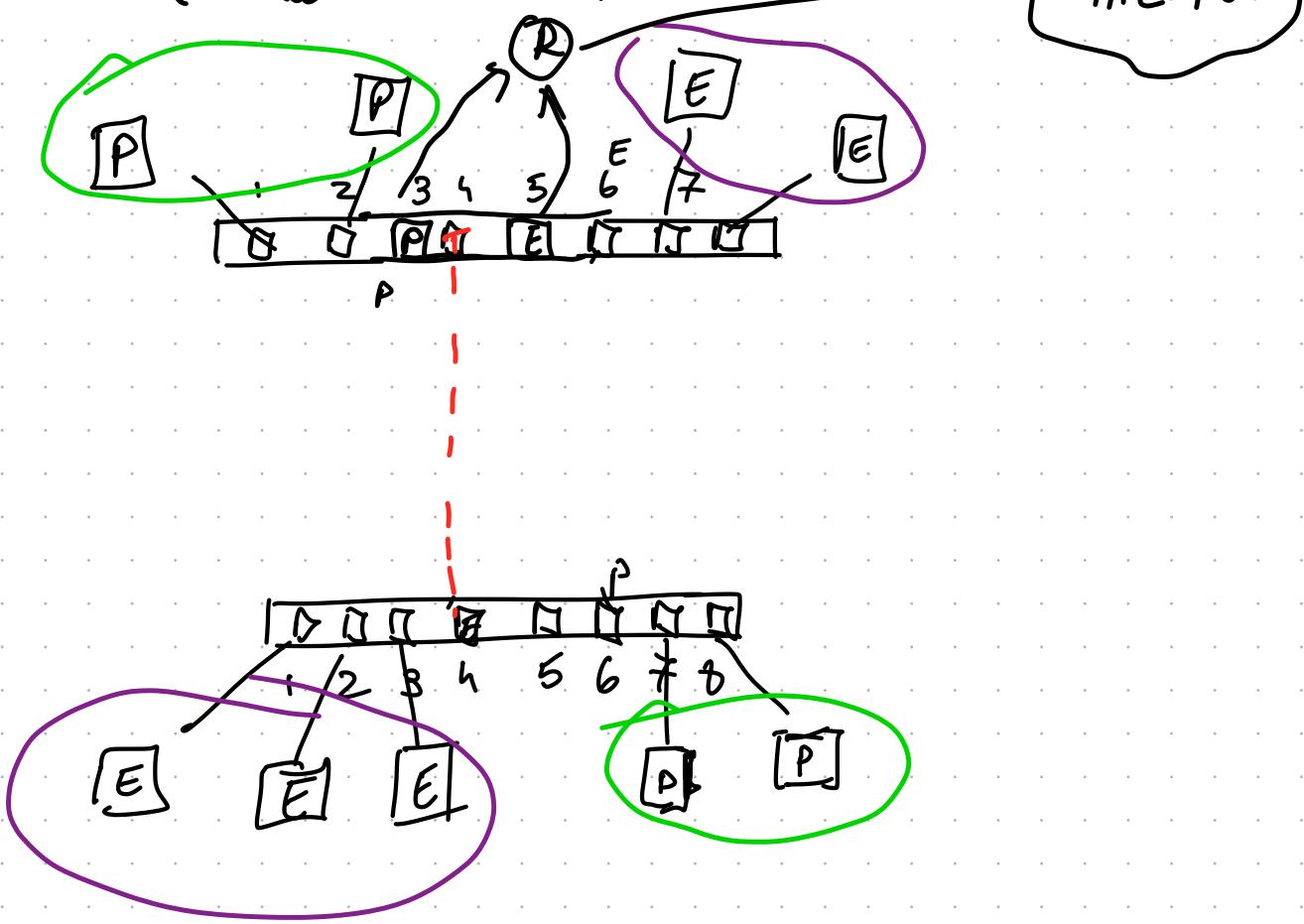


VLAN id

- se poate asocia unui port din switch un id de VLAN



Un port de switch ém mode Trunk = peut être associé avec plusieurs VLAN - une

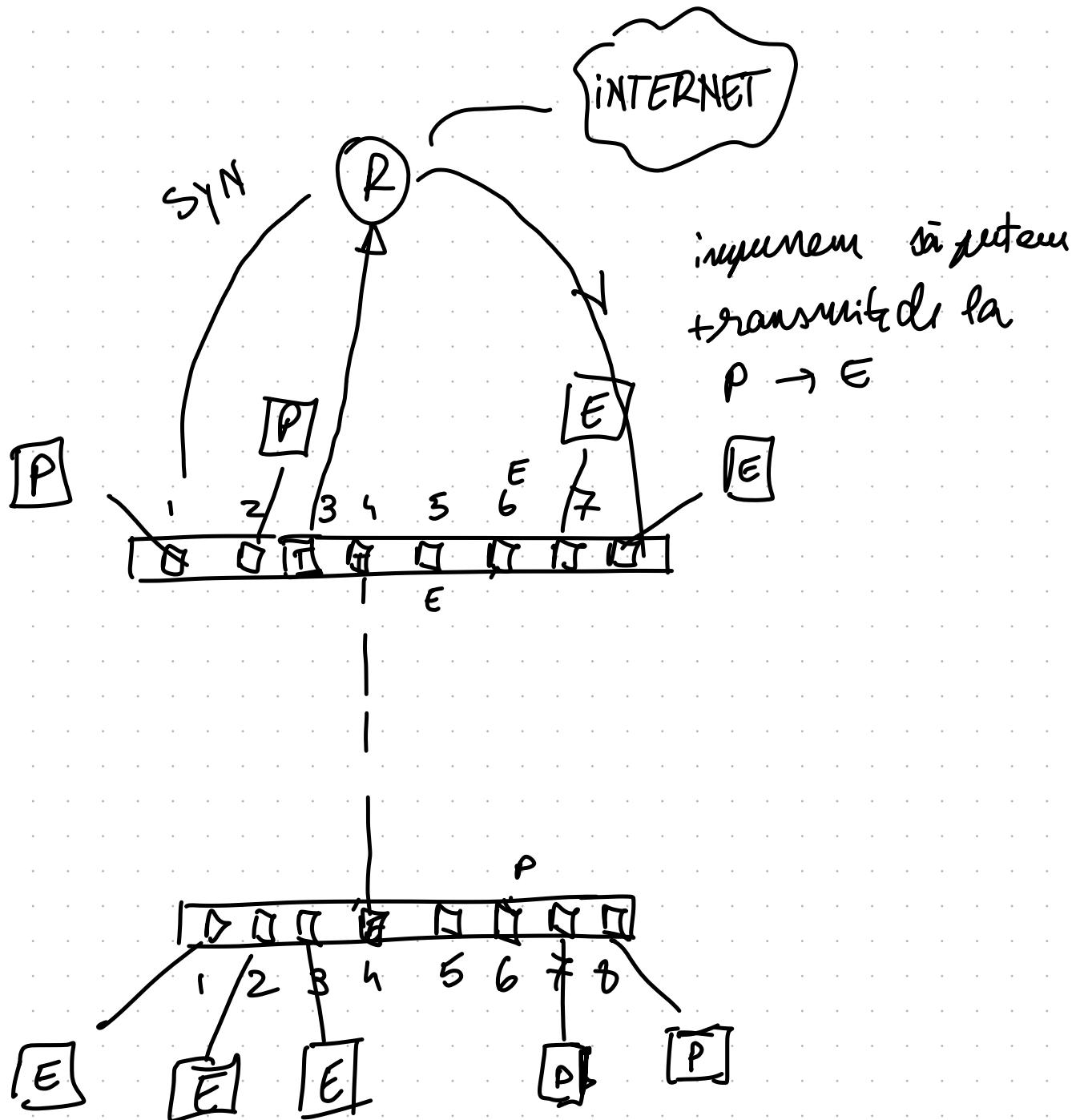


E se vede cu E

P se vede cu P

E NU se vede cu P !

Ca să interconectăm 2 rețele \Rightarrow avem nevoie de router



Clase de adresa

192.168.1.x ⁰
 primii 3 octeti
 255
 adresă IP - 4 octeti

x e octet

Adresele IP au primii 3 octeti în comun, ultimul diferă

192.168.1.x
 adresă rețelei

adresa calcul (device-ului în cadrul rețelei)

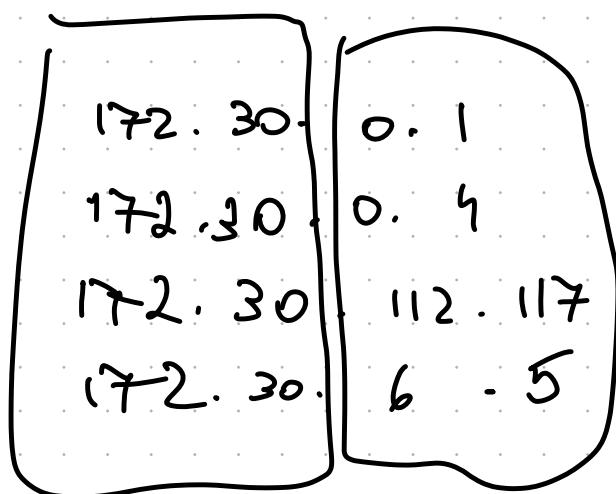
Campuri:

172.30.0.0

172.30.0.0

172.30.255.255

2¹⁶



adresa rețelei

adresa calc. din cadrul rețelei

Subnet mask (Netmask)

Cum putem decide dacă 2 adrese IP fac parte din aceeași rețea?

2^8

255.255.255.0

primii 3 octetii (2^3 biti)

192.168.1.17

rest. adresa rețelei

192.168.1.135

(bitii de 1 din netmask)

-adr.-calc. date de
bitii 0 din netmask

Netmask : 1111 ... 0000

nu pot fi biti 0 și 1 intercalati

ATENȚIE LA EXAMEN

10.1.2.3.

→ sunt în aceeași
rețea dacă

10.17.253.242

255.0.0.0



2^{24}

adică au primul octet în comun

$$\begin{array}{r}
 192.168.1.17 \\
 225.225.225.0 8 \\
 \hline
 192.168.1.0
 \end{array}
 \longrightarrow
 \begin{array}{r}
 192.168.1.135 \\
 255.255.255.0 \\
 \hline
 192.168.1.0
 \end{array}$$

Dacă sunt în același rețea - potem trimite pachete către adresa de rețea (prima adresă din clasa respectivă) fără router
 \Rightarrow se poate face ARP request ca să aflăm MAC dest.

$$192.168.1.17 \longrightarrow, 80.81.82.83$$

Aveam nevoie de router !!

$$\begin{array}{r}
 192.168.1.17 \\
 255.255.255.0 \\
 \hline
 192.168.1.0
 \end{array}
 \neq
 \begin{array}{r}
 80.81.82.83 \\
 255.255.255.0 \\
 \hline
 80.81.82.0
 \end{array}$$

192.168.1.1 \rightarrow prima adresă folosibilă (de
 dică a routerului)

192.168.1.255 \rightarrow adresă de broadcast (nu
 se poate folosi)
 când vrem să trimitem
 pachete la toate calc. din
 rețea

Indiferent de adresa IP dintr-o clasă, dacă facem și (2) între adr. rezp. și netmask se obține adresa de rețea.

Adresa de broadcast (ultima din clasă) se obține dacă și să se între adresa IP și netmask-ului.

ex. 192.169.1.135 //

0 . 0 . 0 . 255

192.169.1.255 = adresa de broadcast

fărăabilă: de la 2 la 254 (af. 2⁸)

172.30.x.y

255.255.0.0

172.30.0.0 = adresa de rețea

172.30.255.255 = adresa de broadcast

172.30.0.1 - prima adr. fărăabilă

10.17.23.45

255.0.0.0 => 10.0.0.0 adr. rețea

10.255.255.255 broadcast

172. 30. 17. 49

172. 30. 115. 231

Nu putem decide dacă sunt în același retea sau
netmask !!

192. 168. 1. X

192. 168. 1. 0 . . . 192. 168. 1. 255

192. 168. 1. 0 / 255. 255. 255. 0

unde începe

rețeaua

124 - nr. biti 1 din
netmask

"cat de măre"
 $2^8 = 256$

172. 30. 0. 0 / 255. 255. 0. 0 - Categorie

172. 30. 0. 0 / 16

10. 0. 0. 0 / 8

Sunt. viitoare - Clase ABC

Curs 8

192.168.1.0 | 2 55.255.255.0

clasa C

/
adresă de rețea sau 124 ↴
 (ultima adresă folosibilă) ↴ netmask
 nr. biti 1 din
 netmask

...
192.168.1.255

/
adresă de broadcast

2^{16} adrese

172.30.0.0 / 255.255.0.0

clasa B

172.30.255.255 sau 116

10.0.0.0 - 1255.0.0.0
/ 18

clasa A

10.255.255.255

! Clase false (private)

{ 0.0.0.0
 255.255.255.255
 2^{32} | "spatiu de adrese"

INADDR_ANY - 0.0.0.0

127.0.0.1

223.255.255.255 - ultima adresă IP folosibilă
 > 224 - scoperi experimentale

Clase de Adrese A, B și C

CLASE A → primul bit din primul octet = 0

10000 . 0000 .

10111 1111 .

0. { → Clase de tip A
127.

~~Retea~~ / ~~nu se folosește~~
0.0.0.0 / 255.0.0.0

netmask = 255 0.0.0

0.X.Y.Z

0. 255.255.255

Bcast

1. 0.0.0 / 255.0.0.0 1.X.Y.Z

1.255.255.255

Bcast

2. 0.0.0 / 255.0.0.0 2.X.Y.Z

2.255.255.255

Bcast

:

126. 0.0.0 / 8

127. 0.0.0 / 8 ~~nu se folosește~~

CLASE B → al doilea bit din primul octet = 0

10 000 000
10 111 111

128 ...

191 ...

128. 17. 0. 0 / 255. 255. 0. 0

140. 255. 0. 0 / 16

172. 30. 0. 0 / 16

(128) . 17. 39. 0 / 255. 255. 255. 0

128. 17. 39. X

128. 17. 39. 255 Broadcast

clasa de tip C,
daca adresa IP
apartine unei
 clase B

CLASE C \rightarrow bitul 3 din primul octet 0

11 000 000 192. ...

11111111 223. 255. 255. 255

finalul spatului de adrese

≥ 254 clase D \rightarrow pe al 4-lea bit din primul octet 0

1110 .

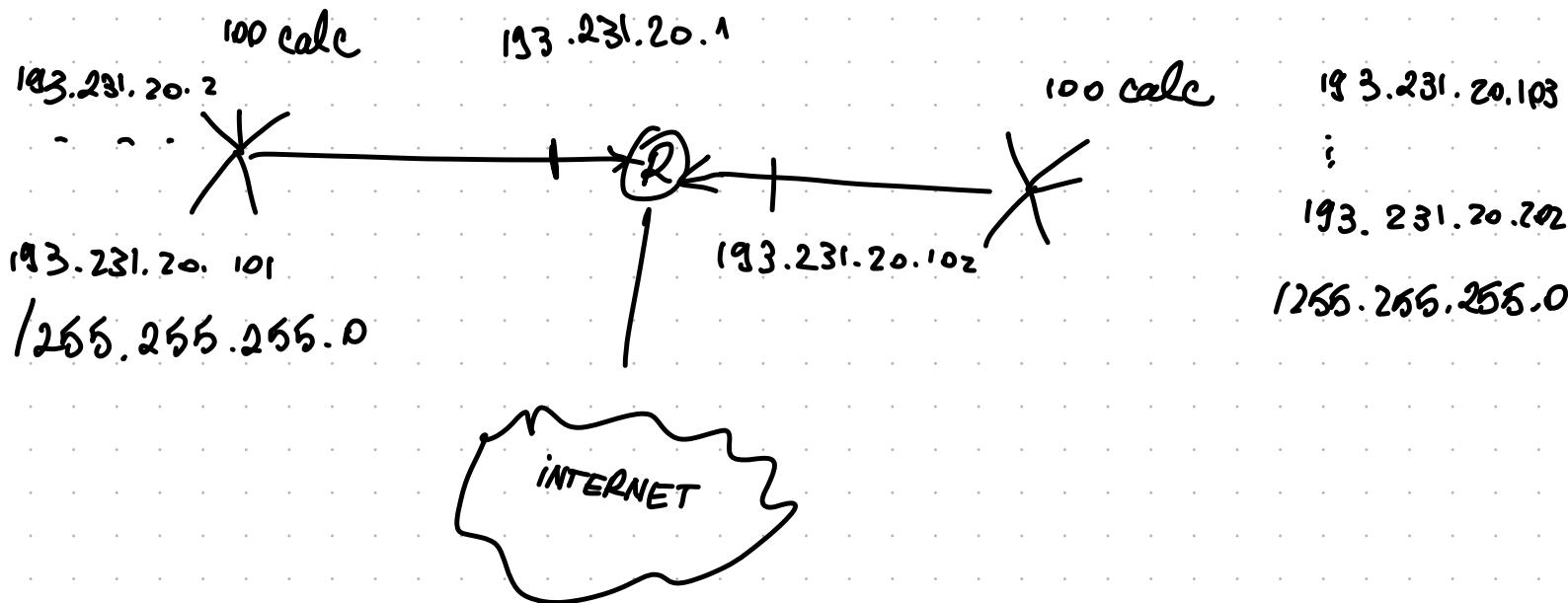
1
239. 255. 255. 0

clase E \rightarrow pe al 5-lea bit din primul octet 0

193.231.20.0 /24

: 255.255.255.0
255

GRESIT



S

D

193.231.20.2 →

193.231.20.110

255.255.255.0

255.255.255.0

193.231.20.0

193.231.20.0

Ar rezulta că S, C sunt în acensiunea rețea deși NV sunt!

Selectie : Împărțim spațiul de 256 adrese în 2 clase:

128

193.231.20.101 rețea
0000 0000

:
193.231.20.127 bcast
0111 1111

128

193.231.20.128 rețea
0000 0000

:
193.231.20.255 bcast
1111 1111

255.255.255.1,000 0000

125 sau 255.255.255.128

255.255.255.1000 0000
.128

125

netmask nu prin care indicău

că adresa le arăta comun primii

25 biti

193.231.20.0 /24



193.231.231.20.0

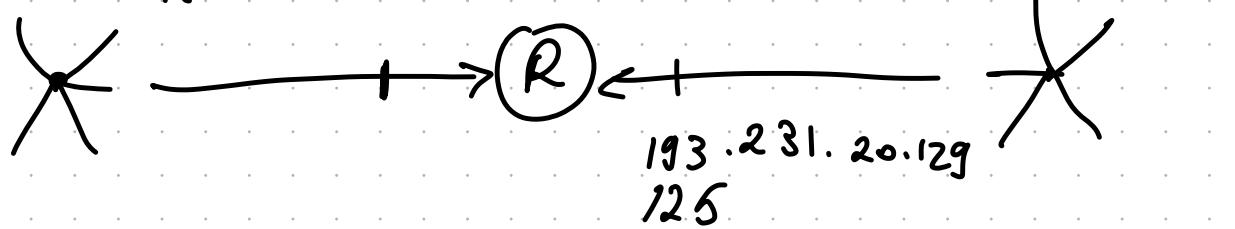
193.231.20.128

1255.255.255.128

1255.255.255.128

193.231.20.1 /25

193. ...



193.231.20.2

→ 193.231.20.130

255.255.255.128

193.231.20.0

≠

255.255.255.128

193.231.20.128

1000 0010

1000 0000

193. 231. 20. 0

0
:
;
127
128
. .
; ;
, 125
;
255

1255. 255. 255. 128

125

1255. 255. 255. 128

125

0
:
;

63

64
:
;

127

128
:
;
191

192

255

$$32 - 26 = 6$$

$$\begin{array}{r} 2^6 = 64 \\ \hline \end{array}$$

1255. 255. 255. 192

11000000

126

⁰
:
31 } 32

32
:
63 } 32

64
:
95 } 32

96
:
127 } 32

1255.255.255.224
127

1110 0000

-
-

Subclase de 4

0 retra 1255.255.255.1111 1100

3 broadcast .252

4 retra 130

7 broadcast

8
11

12
15

16
:

! Adresa de rețea e multiplu de dimensiunea clasei!!

Puteți săparti în clase de 3!!

193.231.20.0

0
127

125
1255.255.255.128

128

128
191

126
1255.255.255.192

64

192
255

126
255.255.255.192

64

256 LA EXAMEN

a) 1, 2, 3, 4, 5

b) 1, 2, 4, 8, 16

c) 1, 2, 4, 6, 8, 255

255.255.255.128

$$\underline{\underline{256 - 128 = 128}}$$

Pt. clasa C (256 adrese ip sau mai mici, urmări deducem dimensiunea unei clase rezănd din 256 ultimul octet din rețea)



B 25 calc

(32)

(16)

10 calc

A



50 calc

(64)

193.231.20.0 /24

A: 193.231.20.0 Retra

:

:

:

20.15 Broadcast

1255.255.255.210

/28

clasa de 16 adrese

$$= 2^4$$

$$\Rightarrow 32 - 4 = 28 \text{ rețea}$$

$$256 - 16 = 240$$

B. 193.231.20. ~~16~~ 32 1255.255.255.224

:

127

20. ~~47~~

greșit!!

63

clasa de 32 trebuie să fie
la multiplu de 32!

ex: 193.231.20.17



193.231.20.33

255.255.255.224

255.255.255.224

0001 0001

1110 0000

—————
1000 0000

0010 0000 1

1110 0000 0

—————

0010 0000
= 32

C: 193. 231. 20. 64 / 126

Curs 9

Agregarea claselor de adrese

Recap Curs 8

193. 231. 20. 0 / 255. 255. 255. 0

Subnet mask

→ Clasa de adrese = adresa rețea + subnet mask

!!

193. 231. 20. 0 / 255. 255. 255. 128

125

193. 231. 20. 128 / 255. 255. 255. 128

125

193. 231. 20. 0 / 255. 255. 255. 192

126

193. 231. 20. 64 / 26

193. 231. 20. 128 / 26

193. 231. 20. 192 / 26

- - -

193.231.20.0 / 255.255.255.252

130

1111 1100

193.231.20.4 | 130

193.231.20.8 | 130

:

- cea mai mică clasă are 4 adrese !!

193.231.20.0 → rețea

193.231.20.1

193.231.20.2

193.231.20.3 → broadcast

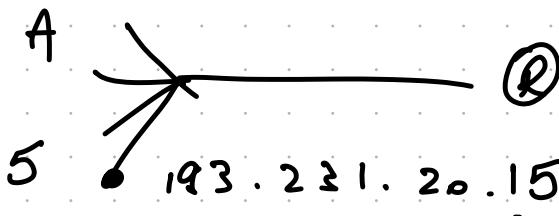
0000 0000
0 000 0001
0 000 0010
<u>0 000 0011</u>

Au în comun primii 30 de biti

→ Dimensiunea unei clase e putere a lui 2 !

→ Adresa de rețea e multiplu de dimensiunea clasei !

ex.



mijlocul clasei de 32

193.231.20.0 / 255.255.255.0
124

2 clase de 256

193.231.21.0 / 255.255.255.0
124

193.231.20.0 rețea

193.231.21.255 bcast

Rețea agregată

193.231.21.0 / 23
 $512 = 2^9$
 \downarrow
32-9

sau 255.255.254.0
11111110 0000 0000

4 clase de 256 \Rightarrow 1024 adrese 2^{10}

193.231.20.0 / 24
193.231.21.0 / 24
193.231.22.0 / 24
193.231.23.0 / 24

\Rightarrow 193.231.20.0 / 22

sau 255.255.252.0

193.231.20.⑥⁸
1024 $\downarrow +②$

la final sunt exact 10 biți de
0 \Rightarrow se poate negați

ex. Să se adauge clasele

→ maxim de 1024!! nu mai mult!!

193.231.21.0 124 }
193.231.22.0 124 }

? clasa de 512

⇒ 193.231.21.0 /23

193.231.21.0

gresit!!

NU SE POATE

193.231.22.255

adresa de rețea nu e multiplu
de 512

193.231.21.00000000

100000000

$$512 = 2^9 = 100000000$$

(2) nu e par! nu se împarte la
512

$$\begin{matrix} x \cdot y \cdot z \cdot 0 \\ \downarrow \\ 2^8 \end{matrix}$$

"Adresa de rețea trebuie să fie multiplu de
dimensiunea clasei"

193.231.21.255

→ nu suntem siguri că e adresa locast fără netmask

ex. 20.0.0.0 /8 clasa A
 1255.0.0.0
 2^{2^4} adrese

20.0.0.0 20.0.255.255

20.0.0.0/16

1255.255.0.0

20.1.0.0/16

20.2.0.0/16

20.3.0.0/16

⋮
⋮

20.254.0.0/16

20.255.0.0/16

O clasa A se poate reprezenta în
cate clase de tip B?

$$\frac{256}{=}$$

$$2^{2^4} = 2^x \cdot 2^{16}$$

$$x=8 \Rightarrow 2^8 = 256$$

Pe 20.3.0.0/16 se reprezintă 2 subclase

$\underbrace{20.3.}_{00\dots} 0.0$

$\underbrace{20.3.}_{128.0} \dots$

20.3.127.255 /17

dimensiunea 2^{15}

Netmask : 255.255.1200 0000.0

255.255.128.0

{ 0.0.0.0
255.255.255.255 → 0.0.0.0 / 0.0.0.0
10

x.y.z.t
0.0.0.0
0.0.0.0 clasa neadivizată

20. 21. 24. x ?
20. 21. 25. x {
20. 21. 26. x
⋮
20. 21. 31. x

8 clase de 256 adrese
 $\Rightarrow 2^3 \cdot 2^8 = 2^{11}$
 $32 - 11 = 21$

$32 \times 256 = 2^5 \cdot 2^8 = 2^{13} = 32$ clase C

20. 21. 63. 255
20. 21. 127. 255

Problema → Pt spatiul de adresa:

[20.21.22.48 192.161.17.95]

1a - l scriem ca trei numere de clase de adresa

unde 20.21.22.48 - adresa rețea

192.161.17.95 - broadcast

(lungime clasele maxime)

• clasă de 16

20.21.22.48 | 255.255.255.240
| 28

• clasă de 64

20.21.22.64 | 255.255.255.192

• clasă de 128 | 256
 (2^7) $32 - 7 = 25$

20.21.22.128 | 255.255.255.128

• de 256 | 25
20.21.23.0 | 24

de 2^8 = 8 clase de 256 (8 clase C)

20.21.24.0 | 255.255.248.0

20.21.32.0 | 19 $5 + 8 = 13$ $2^{13} = 8192$ clase C

20.21.64.0 | 18 = 64 clase C ($2^6 \cdot 2^8$)

20.21.128.0 | 17 ($2^{7+8} = 128$ clase C)
(jumătate de clasă B)

20.22.0.0 | 15

6

↓

↓

2

↓

↓

4

• multiplu de 2

2 clase B

$2^1 \cdot 2^{16}$

$$20.24.0.0 \quad | \quad 13 \quad 2^3 \cdot 2^{16}$$

8 clase B

$$8 \times 2^{16} = 2^3 \cdot 2^{16} = 2^{19} \Rightarrow 13$$

$$20.32.0.0 \quad | \quad 11$$

$$32 \text{ clase B} = 2^5 \cdot 2^{16} = 2^{21} \quad \therefore 20.63.255.255$$

$$20.64.0.0 \quad | \quad 10$$

$$2^6 \cdot 2^{16}$$

$$\underline{256-64} \quad | \quad 255.192.0.0$$

$$20.128.0.0 \quad | \quad 9 \quad \text{jumătate de clasă A}$$

$$21.0.0.0 \quad | \quad 8$$

$$22.0.0.0 \quad | \quad 7$$

$$\frac{1}{2} \text{ clase A} = 2^{25}$$

$$24.0.0.0 \quad | \quad 5$$

$$8 \text{ clase A} = 2^3 \cdot 2^{24} = 2^{27}$$

$$\underline{248.0.0.0} \quad (256 - 8)$$

$$32.0.0.0 \quad | \quad 3$$

$$32 \cdot 2^{24} = 2^{29}$$

64.0.0.0 / 2

64 clase A

/ 192.0.0.0

128.0.0.0 / 2

ladrón de 64 (caja no dejan
192...)

192.0.0.0 / 9

192.127.0.0

$$128 \cdot 2^{16} = 2^7 \cdot 2^{16} = 2^{23}$$

192.128.0.0 / 11

$$32 \text{ clase B} = 2^5 \cdot 2^{16} = 2^{21} \quad \therefore 192.159.255.255$$

192.160.0.0 / 13

$$0 \text{ clase B} = 2^3 \cdot 2^{16} = 2^{19} \quad \therefore 192.167.255.255$$

1255.248.0.0

192.169.0.0 / 20

$$16 \text{ clase C} = 2^4 \cdot 2^0 = 2^{12} \Rightarrow 32 - 12 = 20$$

192.168.16.0 / 24

192.168.17.0 / 26

192.168.17.64 / 24 \rightarrow clasa de 32

1255.255.255.224

\therefore 192.168.17.95

192.168.17.96 / 32

1255.255.255.255

\downarrow

clasa de o singură adresă
IP

ex. de mărescere ~~pt.~~ examen

Precizați o clăsă ce conține ambele adrese:

80.81.82.83 \rightarrow cea mai mică

82.83.84.85

$$\begin{array}{l} 80 = \boxed{0101\ 00} 00. 81. 82. 83 \\ 82 = \boxed{0101\ 00} 10. 83. 84. 85 \end{array}$$

Netmask: 1111100 . 0 . 0 . 0

252 . 0 . 0 . 0

16

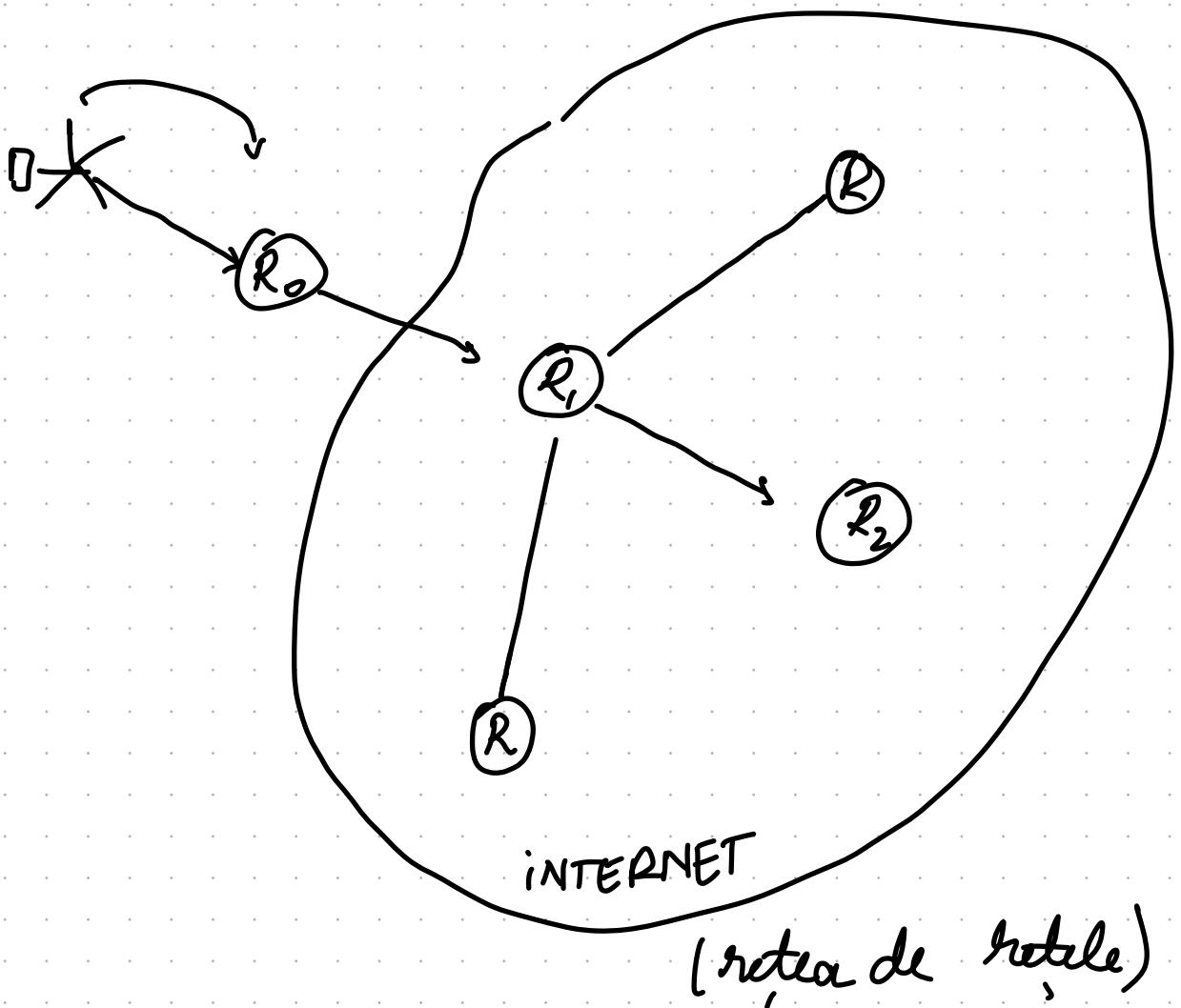
adr. retea: 80 . 0 . 0 . 0 \leftarrow și se lipesc
83. 255. 255. 255

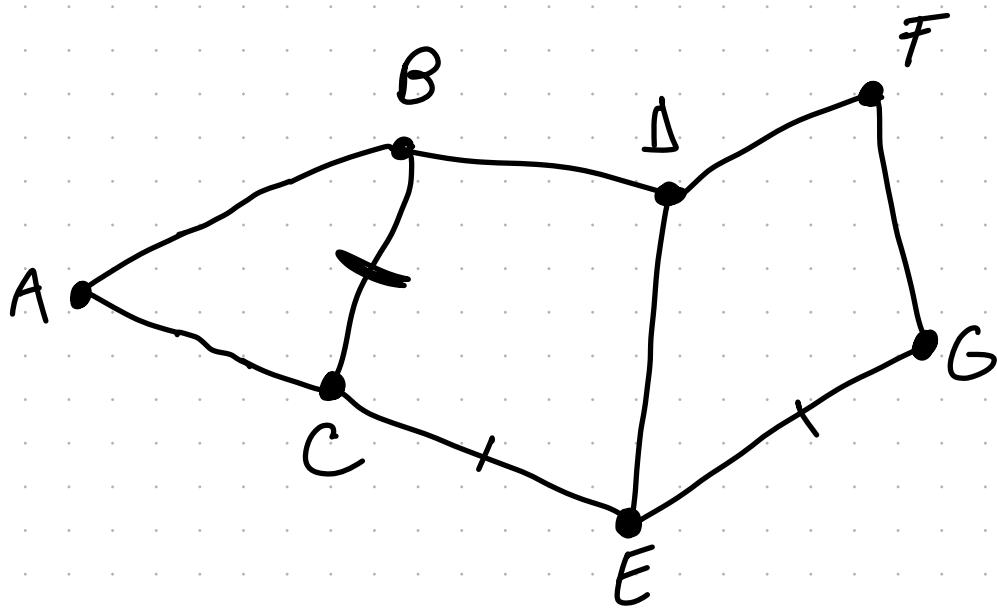
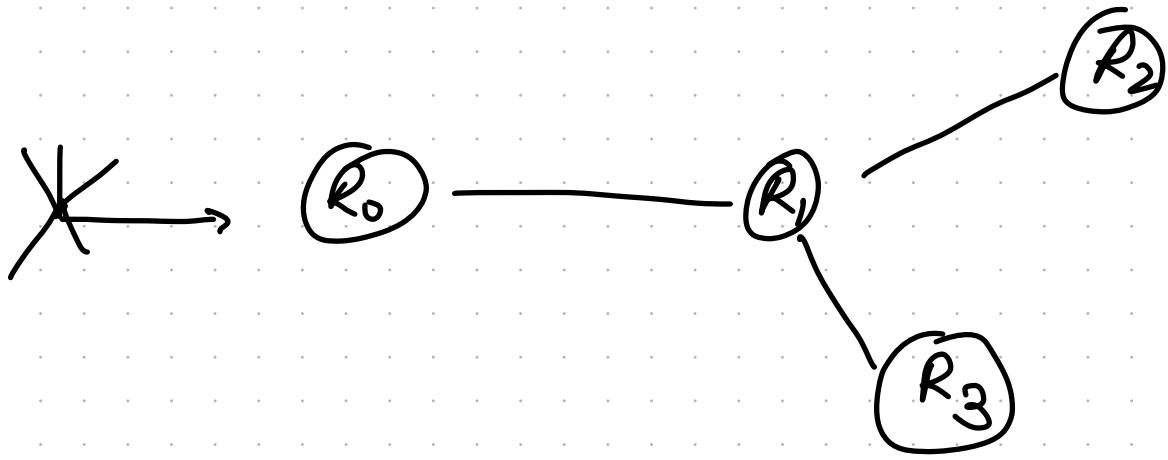
\Rightarrow 80 . 0 . 0 . 0 , 0 16 (conține 4 clase A)

- În câte clase cu netmask 255.255.255.252 se împarte clasa de dimensiune maximă care are aceeași adresă de broadcast ca clasa de dimensiune maximă care începe la adresa 193.244.0.0? (4 clase A maximă)
193.247.255.255 (adr. bcast)

Teuă !!

Curs 10
Dirijare (Routare)

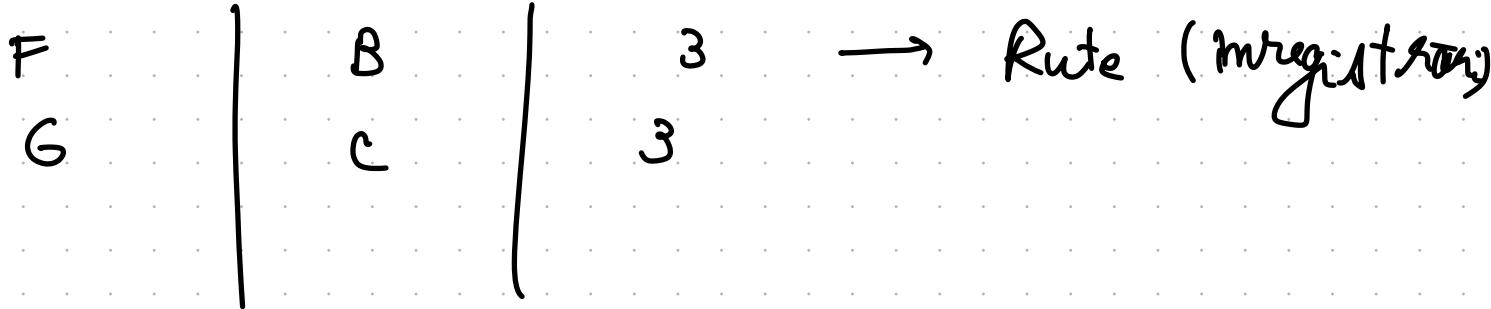




A:

Dest	Vecin	Lung / Cost
A	A	0
B	B	1
C	C	1
D	B	2
E	C	2

Tabelă de
dirijare



Algoritmi de direcțare

- implementații în PROTOCOALE DE DIRECȚARE
(implementate și rulează de către răzăre)

- bazări pe :
 - a) vectori distanță (distance vector)
 - ↳ RIP, BGP Bellman-Ford (Kolaba)
 - b) starea legăturilor (link state)
 - ↳ Dijkstra

a) RIP - routing internet protocol
~~BGP~~ - border gateway protocol

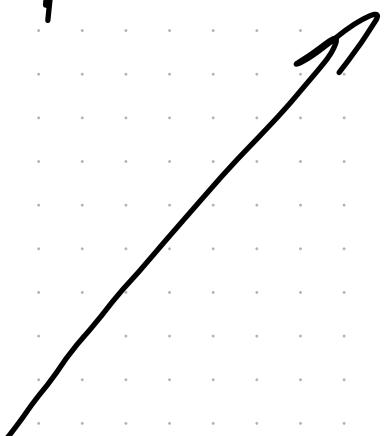
A:

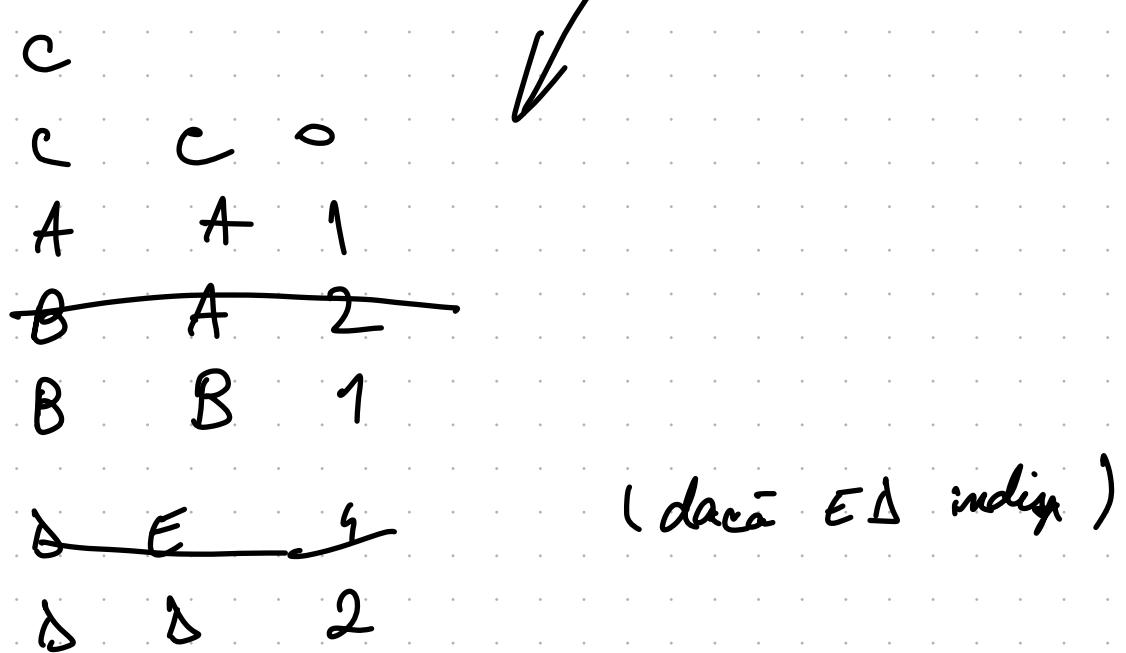
Lat	Vec	Lung
A	A	0
B	B	1
C	C	1



B:

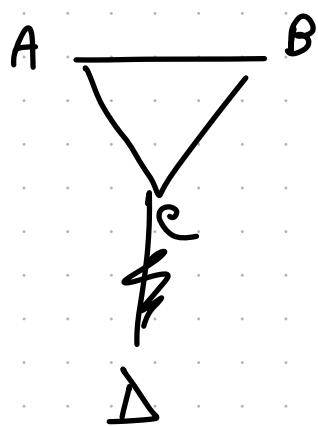
Jst	Vec	Lung
B	B	0
A	A	1
D	D	1





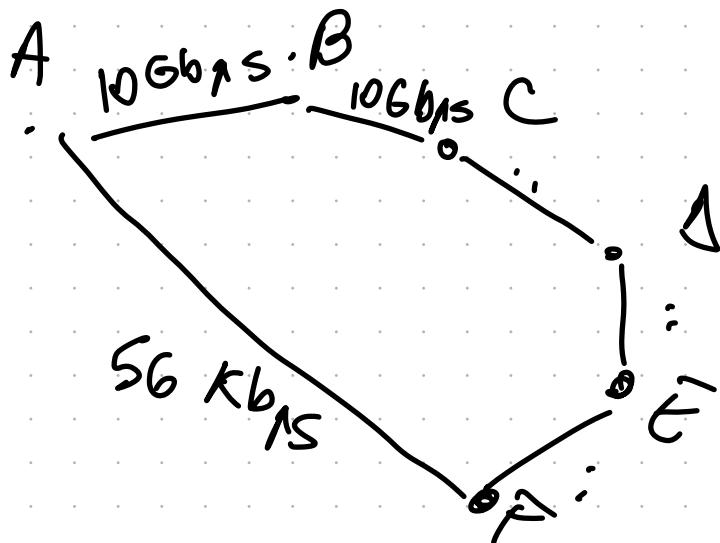
- Routerurile schimbă între ele tabele de direcție = propagare de rute
- info se propagă din ruta în ruta!
de ex: G învăță despre A de la F!!
- Router-ul și updatează tabela în funcție de tabelele vecinilor

Extra: Problema numărării la infinit

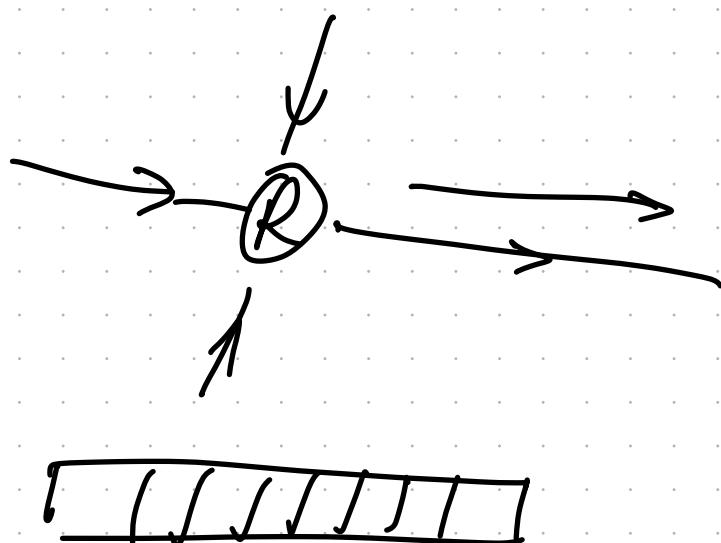


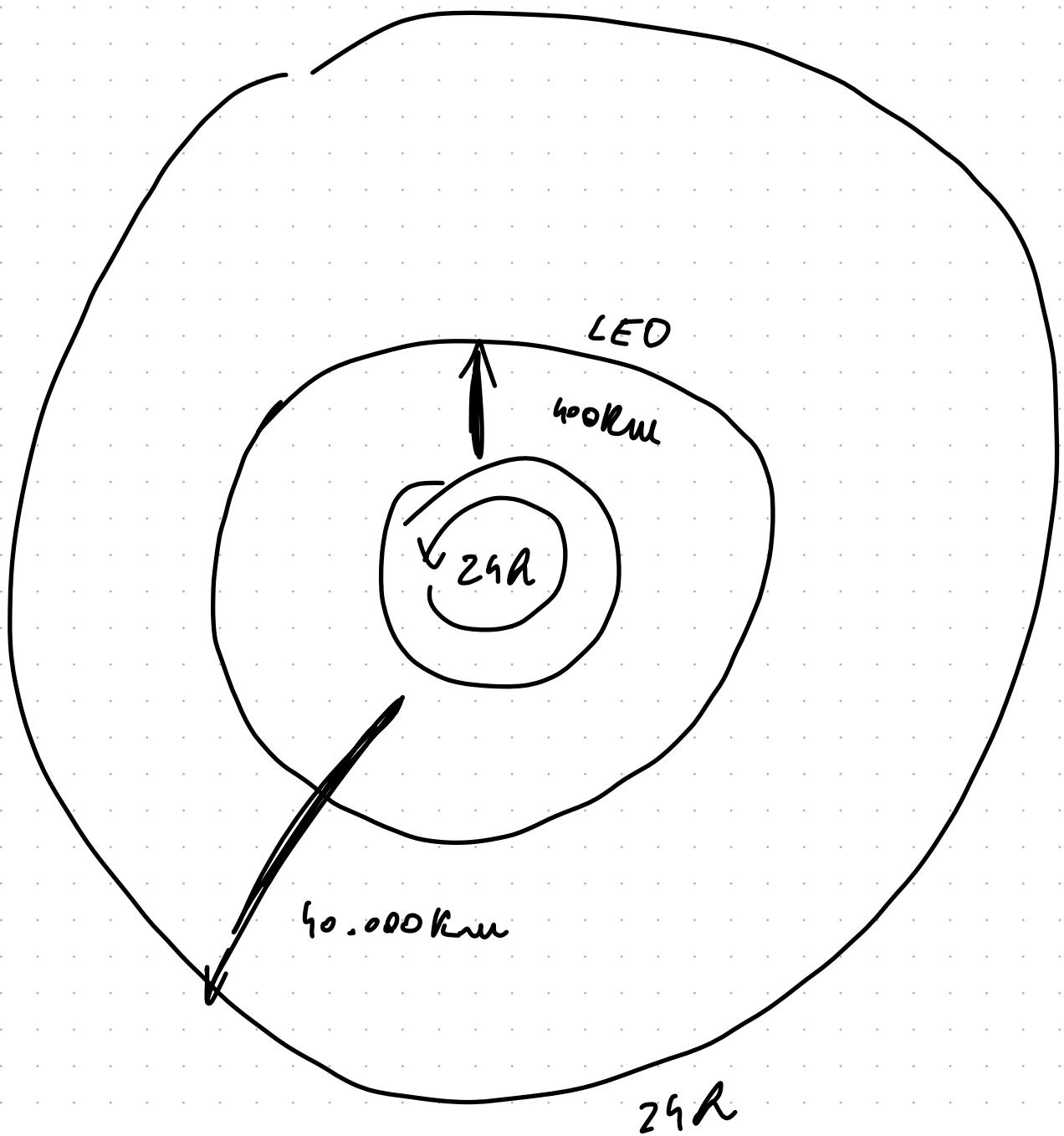
- niciun router nu are o vizionare completă a întregii rețele !!

RIP



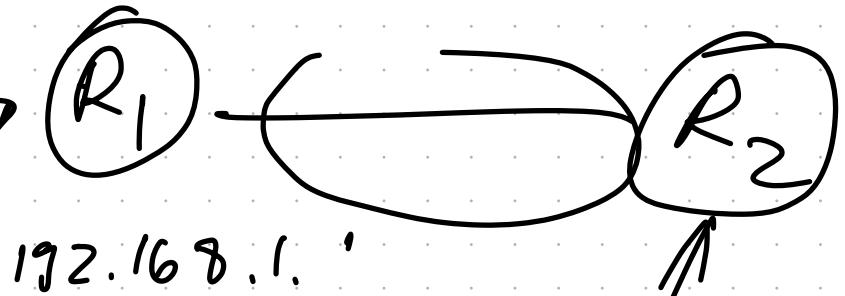
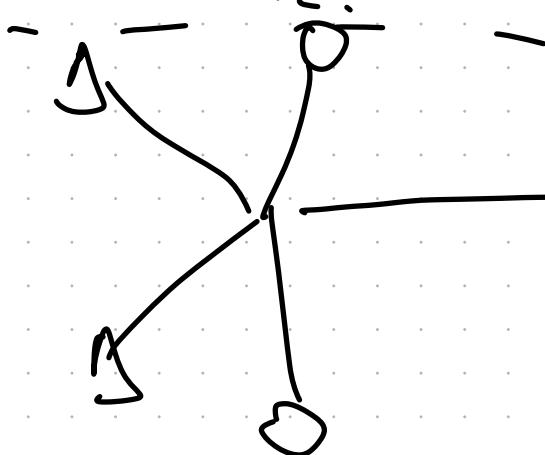
- nu introduceaza drumul de lungime minima e cel mai rapid!!
- se ia in calcul si viteza linilor





crossover
la fel - punctul (rație - rație)
diferite - Normală

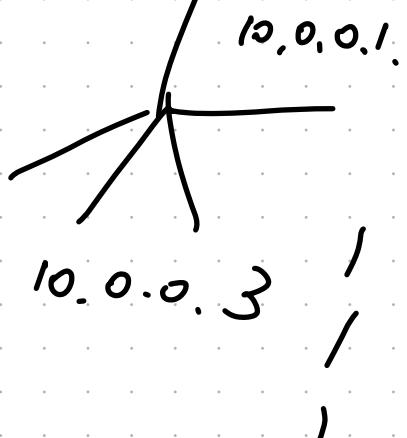
192.168.1.0/24



Dst

0.0.0.0/0 vecin
192.168.0.1

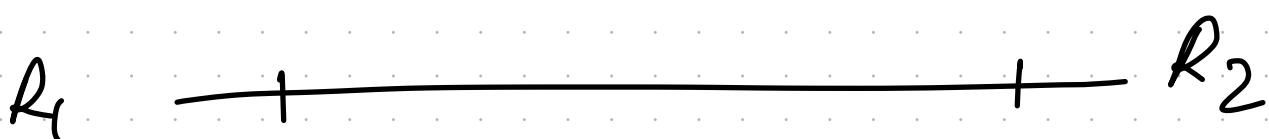
A
Ruta default

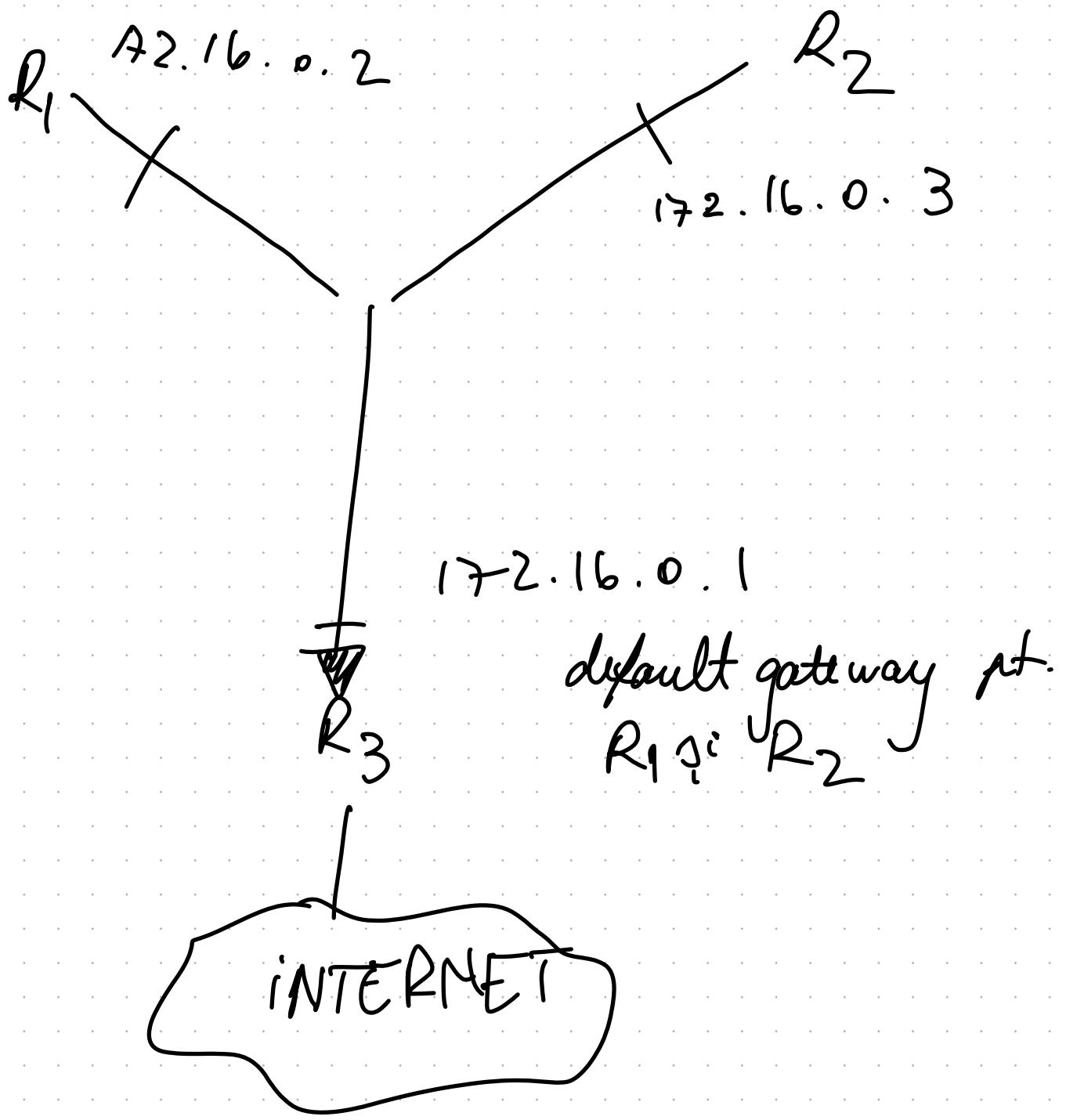


3 retele !!

intre R₁ și R₂ ori cablu punctat
ori switch intre ele !!

subclasa de 4





5192.168.1.0/24 → 10.0.0.3

- I. trimite la R₁
- II. R₁: și pe hizi între 10.0.0.3 și adr. lui trimite la R₃ !! gresit

R₁:

Dest

10.0.0.0/8

0.0.0.0/0

Vecin

R₂, 172.16.0.3

R₃, 172.16.0.1

R₂: Dest

192.168.1.0 124

0.0.0.0/0

Vecin

172.16.0.2

172.16.0.1

windows: route print

193.231.18.0/24

193.231.19.0/24

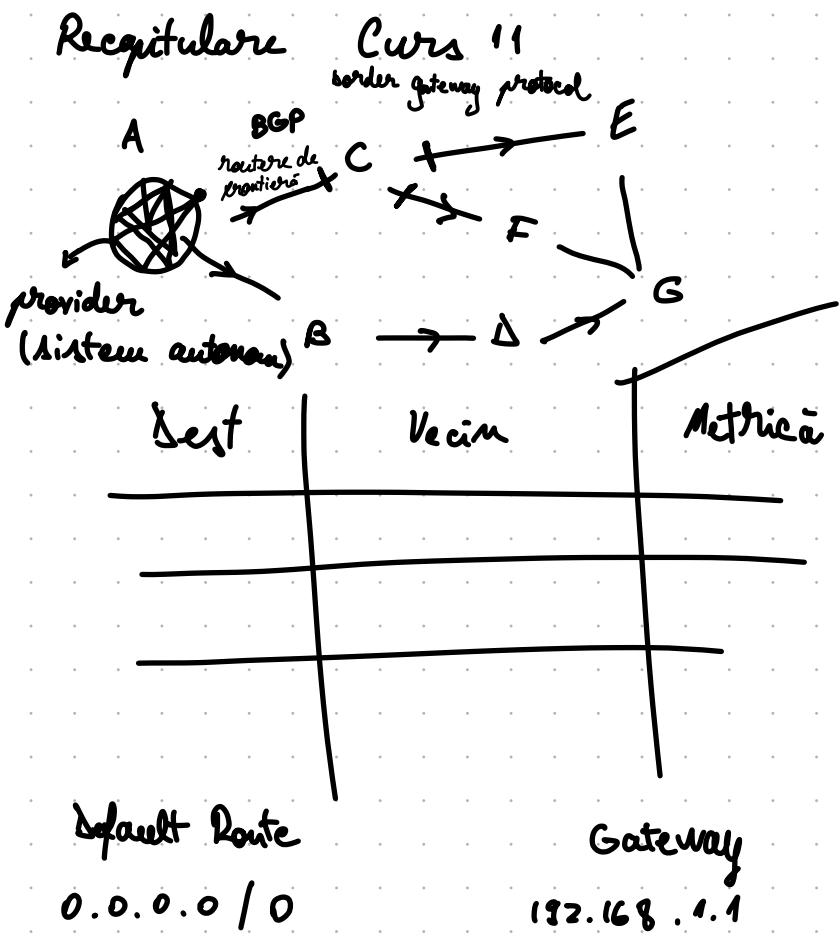
193.231.20.0/24

}

193.231.18.0/23

rt.0/0 via

Curs 12



Next Hop

Rute

b) stocarea legăturilor (link state)

„inundare”

(A, C, 1, timestamp)

(A, B, 1, timestamp)

→ identifică dacă un anumit pachet e mai mare

- dezavantaj → sunt mai intensive computational > alg. bazată pe vectori distanță

→ sunt utili doar în rețele mai mici (Venezuela - Thailandia ?!)

Rute de frontieră → conectate și cu rutele din propria rețea și cu rutele din

Reteaua altor provideri

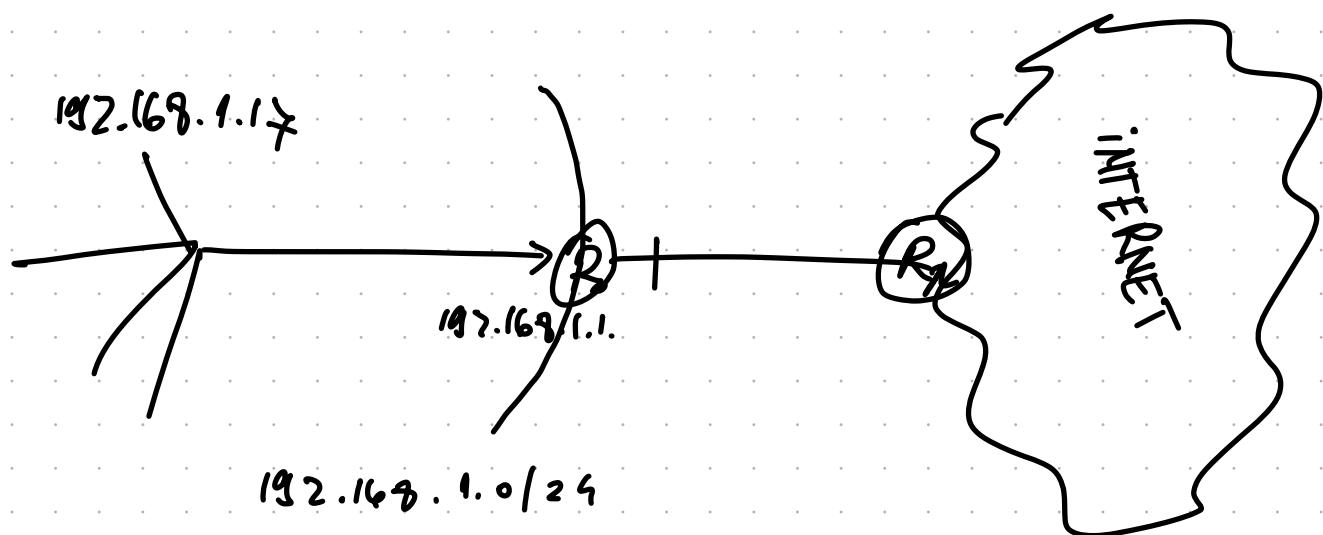
- Fiecare provider are un nr. de AS (autonomous system)

Prin care se identifică în graful "global"

public route servers
routers

netty / nsr
netty / telnet
> traceroute

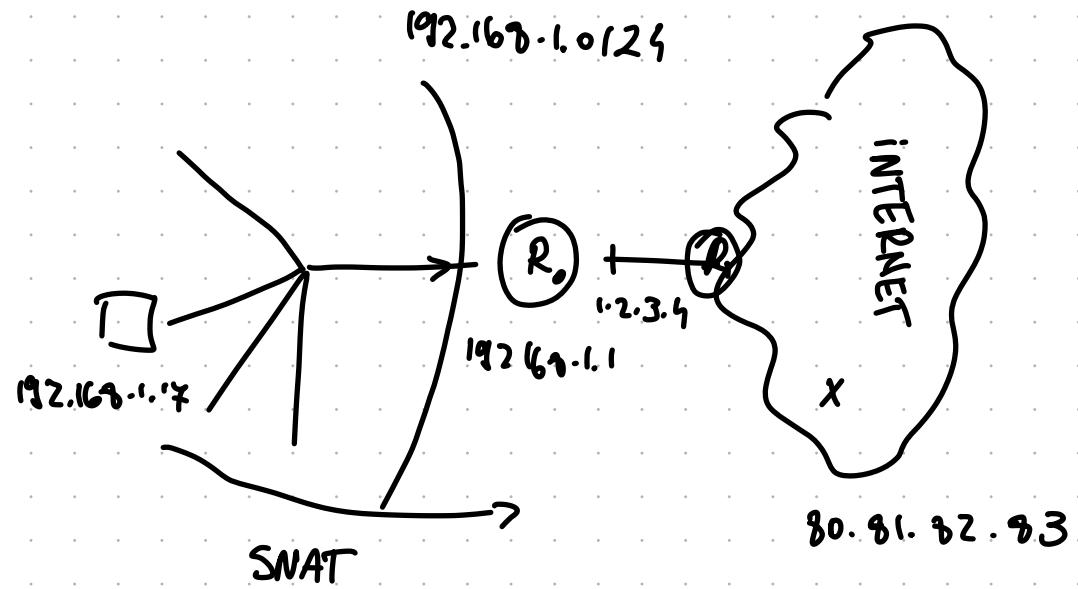
ex. Reteaua de acasă



80.81.72.0/24
↳ facebook

Ro nu implementază algoritmi de dirijare !!

clase false / private



10.0.0.0/8 1 clasa A

172.16.0.0/16 2

..
172.31.0.0/16 16 clase B

192.168.0.0/24

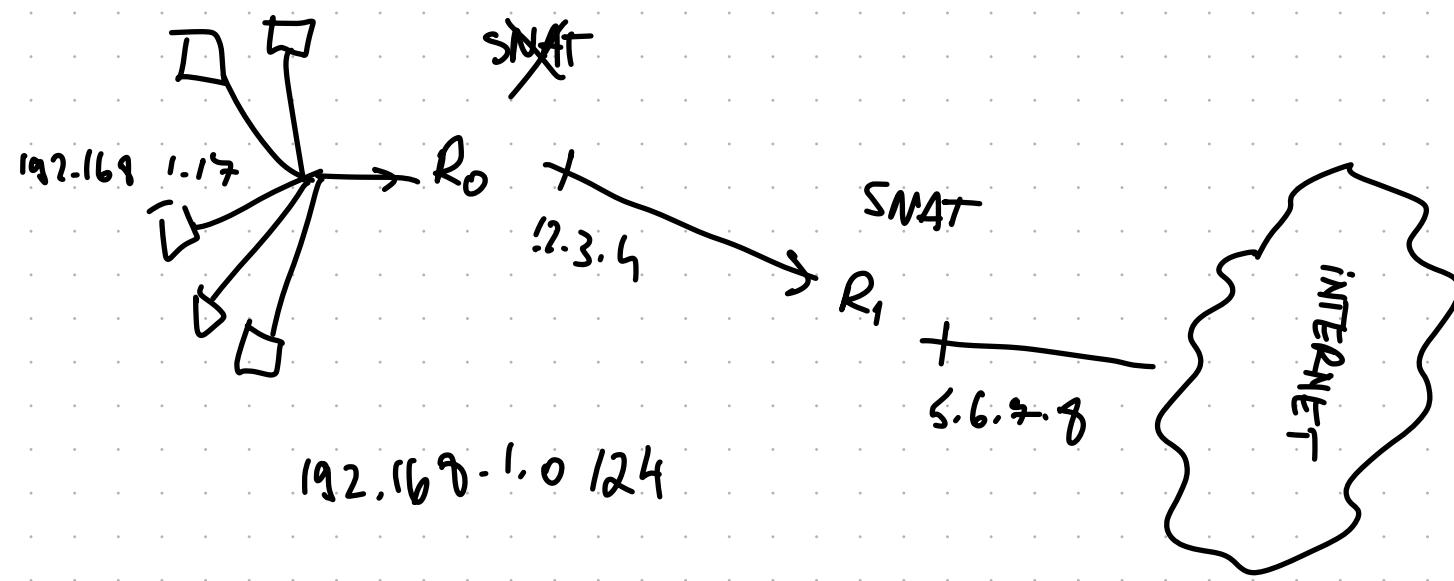
192.168.1.0/24

192.168.255.0/24

256 clase C

SNAT - adresa versă nonruteabilă \Rightarrow adresa publică a routerului

Examen:



80.81.82.83

P_e R₁:

Dest

192.168.1.0/24

V

R₀: 1.2.3.4

ex:

172.30.0.0/16

172.30.0.4

172.16.0.2/16

172.30.0.4

SNAT

172.16.0.1/16

192.231.20.20

AS

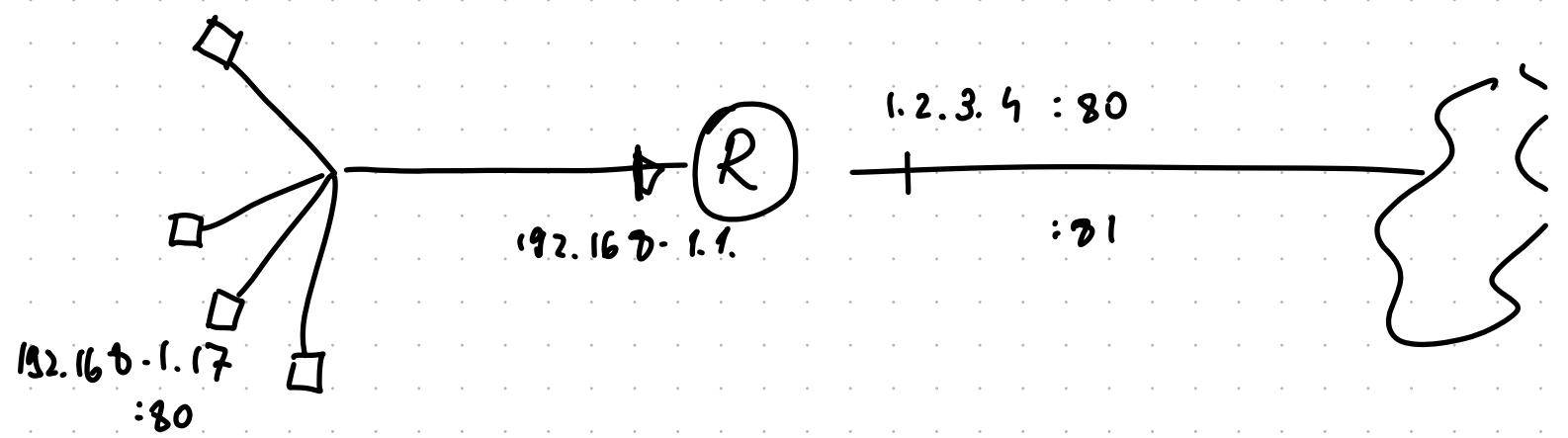
Vecile

172.16.0.0/16

172.30.0.4

DNAT

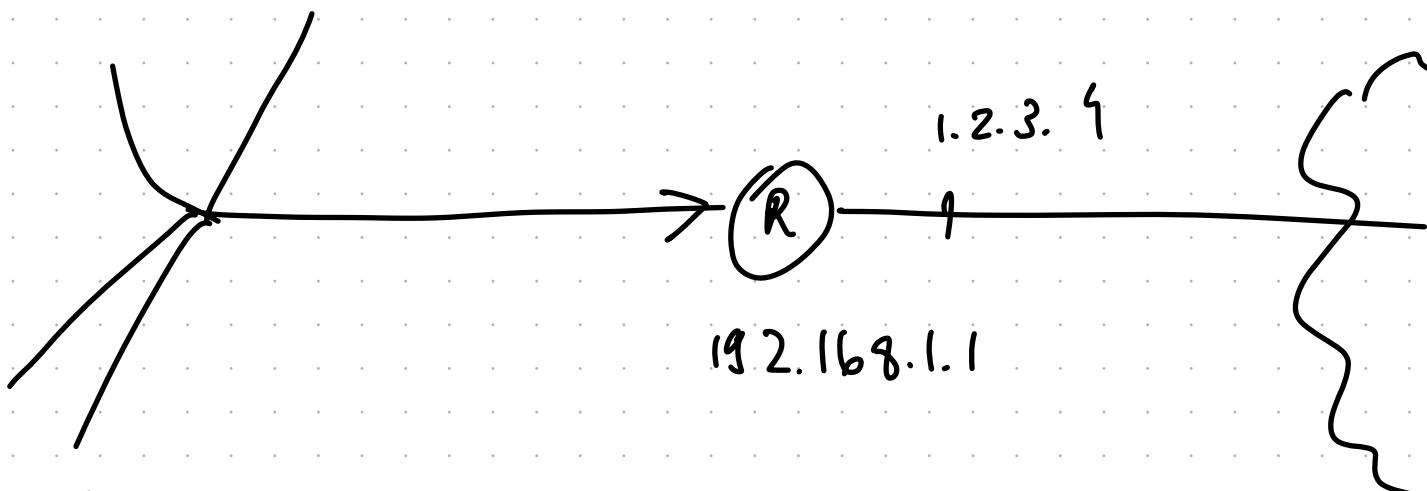
192.168.1.0/24



DNAT (virtual servers
port forwarding)

- SNAT - 1. + traducătoare versoā (din :Ports în iPs tral)
2. pe răspuns se face DNAT
(modifică iP destinație)

192.168.1.100



192.168.1.17

80.81.82.83

:80

Packet manteine SNAT

IP verso:	Port verso	IP dest	Port dest
192.168.1.17	2179	80.81.82.83	80
192.168.1.100	5076	90.81.82.83	80

Dopo SNAT

IP verso:	Port Verso	IP dest	Port dest
1.2.3.4	2179	80.81.82.83	80
1.2.3.4	5076	80.81.82.83	80

Tabla NAT