05.10.2022

Cursul 1

Andrew Tanenbaum – „Rețele de calculatoare"

Socket-uri

- comunicare între procese de pe calculatoare diferite (pot fi și sisteme de operare diferite)
- model client-server (browser → client)
- ✓ codurile pot fi scrise în orice limbaj de programare (cele care comunică nu trebuie să fie de același tip pt. a funcționa comunicarea)
- compatibilitatea este dată de protocolul de comunicare

Stiva TCP/IP

Principii fundamentale ale funcționării rețelelor de calculatoare:

→ Dirijarea (routarea) pachetelor → făcută de routere

sursă —[pachete]→ destinație → nu neapărat liniară (în ordinea inițială a informațiilor)

→ Mecanismul de adresare IP

→ Sistemul numelor de domeniu (DNS)

↳ translatarea din nume de calculatoare în adrese IP

↳ ajuts la localizarea altor sisteme DNS, la distribuirea de chei publice, etc.

→ Stiva de protocoale TCP/IP → RFC (Request for Comments) → anii '70-'80

• protocol = set de reguli de comunicare

Diferențe dintre un program și un proces. → un program poate fi instanțiat de mai multe procese

program (static) → în sistemul de fișiere

proces (dinamic) → program în execuție (rulează pe RAM)

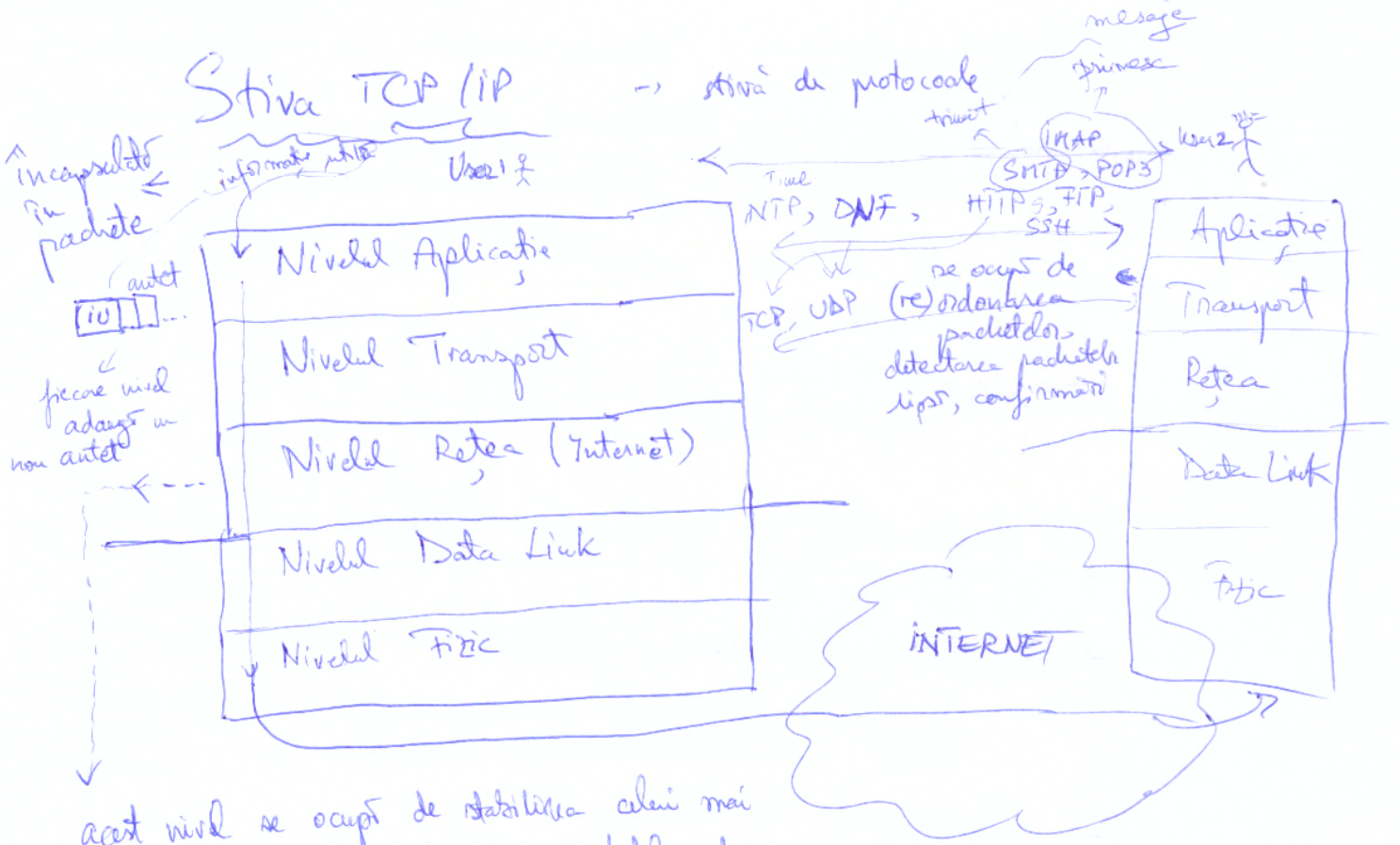
- protocoalele sunt descrise în RFC-uri

IS - Mi
Apache - Linux,

HTTP - Hypertext Transfer Protocol
HTTPS

protocoale < deschise
proprietary

Stiva TCP/IP → stivă de protocoale



acest nivel se ocupă de stabilirea celei mai optim traseu pentru livrarea pachetelor de informații (se poate ca un pachet trimis mai târziu să ajungă mai repede)
↳ ordine ne-liniară

- Protocoale de la nivelul Transport
  - TCP (face toate verificările) → mai lent, mai sigur, protocol reliable
    → ordine, completitudine
  - UDP
    ↳ mai rapid, se pot pierde pachete
- Aplicație
  - NTP, DNF
  - HTTP, HTTPS, FTP, SSH
  - SMTP, IMAP, POP3

routerul:

router, switch, access point

- porturi
  - fizice (din switch)
  - de la nivelul Transport

Default

HTTP → port 80

HTTPS → port 443

FTP → port 21

SSH → port 22

/etc/services

→ Nivelul Rețea → adresa IP (Internet Protocol)

→ Nivelul Data Link → switch, adresă MAC
/ethernet
/hardware

→ Nivelul Fizic → cablu, fibră, linie telefonică, wireless

structură de informație

socket (informația din apl. este pasată nivelului transport și invers la destinație

} căci nu se mai numește pachet, ci frame / cadru

antet aplicație

protocol TCP + port sursă + port destinație

IP sursă + IP destinație

MAC sursă + MAC destinație

→ decapsulează antetele, pt a ajunge la IU

modelul stiva OSI → Încercare de standardizare a comunicării (mult mai multe niveluri)

- VPN → se introduce niște pachete IP peste o altă stivă TCP/IP

12.10.2022

Cursul 2

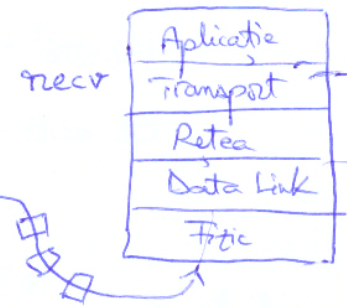
CLIENT ⟶ SERVER

IP, portul serverului

↓ identificarea procesului cu care se comunică

- Server → hardware / sistem de operare / proces

→ pachete →



→ aici se face diferența între ce proces ce pachete primește

⇑ voluul portului

frame-uri

CLIENT	SERVER
c = socket(,)	s = socket(,)

↳ e vorba de protocoalele:
- Transport – TCP
- Rețea – IPv4

→ pagina următoare

```
if ( connect(c, { - IP server, - port server }, sizeof(structură)) < 0 )
```

- tip structură*

↓ sub forma unei structuri

↓ eșuează

* tip structură : AF_INET

```
send(c, &i, sizeof(i), 0)
recv(c, &j, sizeof(j), MSG_WAITALL)
close(c)
```

SERVER

S = socket (,)

if(bind (S, { - INADDR_ANY (→ 0.0.0.0) / - port }, sizeof(structură)) < 0)

aceasta îi conferă atributul de server ← listen (S, 5)

["192.168.1.17"

inet_addr
inet_ntoa

$192 \cdot 256^3 + 168 \cdot 256^2 + 1 \cdot 256^1 + 17 \cdot 256^0$

$192 \ll 24 + 168 \ll 16 + 1 \ll 8 + 17$]

pachete

client ≡ IP sursă, port sursă IP destinație, port destinație ≡ server

roluri interschimbabile

⇒ socket-urile sunt bidirecționale

→ pentru a verifica ce porturi sunt ocupate: netstat

bind → la client este opțional (în general, nu se face), pentru că portul clientului nu este important și, de asemenea, dacă s-ar face bind la client există riscul de a ajunge pe un port deja ocupat

while (1) {

deservirea clientului respectiv

c' = accept (S, { - IP client / - port client }, "sizeof")

recv (c')

send (c')

close (c')

} close (S)

un server iterativ

! Obs: ordinea pt. send și recv, semantica + alte detalii / reguli de comunicare → RFC

→ transformarea serverului iterativ în server concurent:

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

înainte de trimitere	
htons	ntohs
htonl	ntohl
	după primire

fie send la client → send (c, &i, sizeof(i), 0)

↳ | E8 | 03 | 00 | 00 |

recv la server

este necesar ca arhitecturile să fie aceleași (dif. little endian vs. big endian)

- datele, în rețea, circulă în format big endian

19.10.2022

Cursul 3

Client $\xrightarrow{IP_{server} : PORT_{server}}$ Server

netstat → afișează ~~și~~ să vedem porturile libere

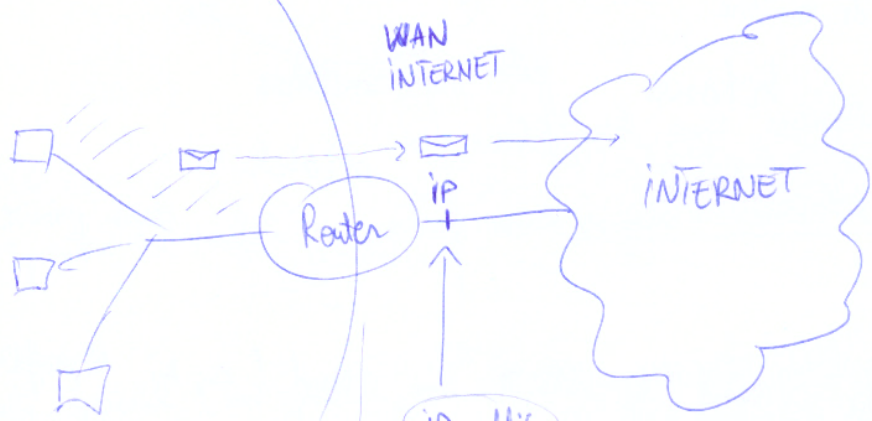
! trebuie verificat dacă s-a făcut:
- connect
- bind

ss - pe Linux mai nou

Local Area Network

LAN ← adrese IP private (false) (nerutabile)

ipconfig - Windows
ifconfig - Linux
ip addr



192.168.1.X
192.168.0.X
172.30.X.Y
diverse clase de adrese

IP public
(real / routabil)

! Obs: între 2 sau mai multe device-uri din LAN nu este necesar routerul

NAT Network ~~Address~~ Translation
SNAT Source —"—

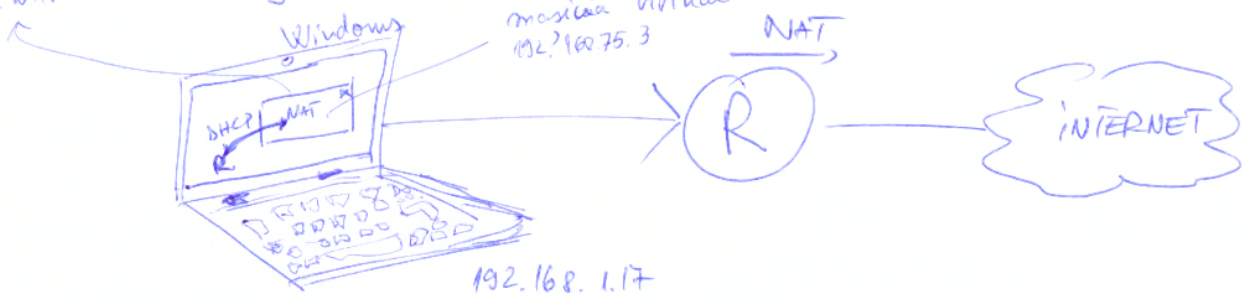
→ dacă merge ping între două IP-uri, va funcționa conexiunea

IP C: 192.168.1.17

IP S: 192.168.1.37

→ nu e neapărat să fie în aceeași rețea, doar observăm că fac parte din aceeași clasă de adrese IP.

Modif. NAT cu model Bridge



- Mașina fizică „joacă rolul de router" pentru mașina virtuală
- Dacă avem model Bridge pt. mașina virtuală (nu NAT, ca mai sus), mașina fizică nu mai e router pt. aceasta, căci însuși Routerul (R) îi va furnizat adresa IP. (forma 192.168.1.X)
- Nu este obligatoriu ca serverul DHCP să ruleze pe router.

(când un device vrea să se conecteze la un router/server, trimite un mesaj de tip broadcast)

→ client și server în rețele diferite:

- reală (publică, routabilă)

? dacă e privată
⇒ trebuie să fie un router

dacă cineva dorește să se conecteze la adresa ip a serverului ~~și dă redirect la adresa privată a serverului~~ trebuie să specifice adresa IP reală a routerului (extern) → clientul

DNAT
Port forwarding
Virtual Server

172.30.4.5 → nu va avea acces la internet !

! Obs: Adresele IP ale device-urilor dintr-o rețea trebuie să semene cu cea a router-ului.

C → S

```
uint16_t n;
n = htons(n)
send(c, &n, ...

for(i=0; i < ntohs(n); i++)
    x[i] = htons(x[i])
    send(c, &x[i], sizeof(x[i]), 0);
```

```
uint16_t *x;
x = malloc(...)
send(c, x, n*sizeof(uint16_t), 0)
```

Dacă e șir de caractere: C: n = strlen(x) → htons → send

C: send(c, x, strlen(x)+1, 0)

```
S: recv(c, &n, ..
   n = ntohs
   recv(c, x, n+1
   ~~x[n] = 0~~
```

! Important: câte date trimitem, atâtea citim + de același tip de date (important pt. size)

Ex:

```
int c;
char S[100];
uint16_t i, j;
int x;
c = socket ; connect --
scanf("%hu", &i)
```

↓

se citesc 2 octeți

(hu - short unsigned)

// scanf("%d", &i)

! → pune la adresa lui i 4 octeți, inst i are doar 2 octeți

=> se scrim 4 octeți care pot afecta alte date / variabile

strace / ltrace

./strace ./client

26.10.2022

Cursul 4

TCP < protocol orientat pe conexiune
< reliable

Socket-uri UDP

Emitator (client)

Receptor (server) (roluri „mai" egale; interschimbabile)

c = socket (AF_INET, SOCK_DGRAM

S = socket (AF_INET, SOCK_DGRAM)

* UDP - nu se poate face deosebire - între cazurile când
  - se pierd date de la E la R
  - nu există un proces cu care ar se comunica
  - se pierde rezultatul / răspunsul

* 64 Kb - dimensiunea maximă a unui packet (MTU - Maximum Transfer Unit)

Situație UDP (neverificate, ca în cazul TCP) < pierderi de date
< ajung datele, neordonate (out of order)

↓

este mai importantă rapiditatea (modelul interogare-răspuns)

bind (S, ~~listen~~ ~~accept~~

~~connect~~

sendto (c, de unde din memorie, cât, flag, structura, sizeof (structura))
- IP dest
- port dest

recvfrom (S, unde, cât, flag, structură, sizeof(str))
- IP emitator
- port emitator

recvfrom (

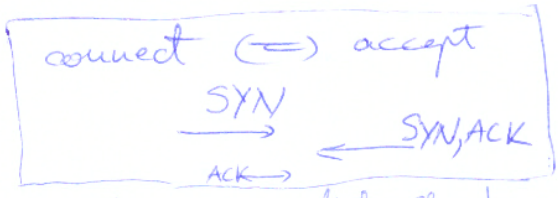
close (c)

Sendto (

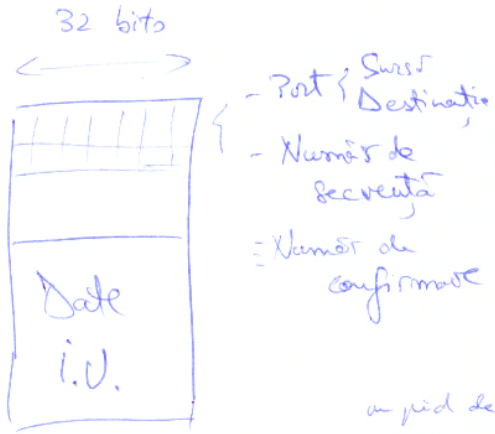
* sendto trimite datele unui partener ale cărui date sunt de identificare completate de recvfrom

(Dacă fac serverul concurent, după recvfrom generez un port random (pt. identificare), fac un fiu, iar „comunicarea" se face pe baza portului nou generat.)

TCP:



(UDP/IP), TCP packet structure



un pic de la close → FIN

⇝ pachete de control → fără date
(ex: se trimit pt. a confirma că încă are loc conexiunea și ce primesc date din sens invers)

→ confirmările nu se fac instant (ar supraaglomera fiul de conexiune / rețeaua cu pachete goale / de control)
↓
se confirmă câteva de date

ex:

```
c = socket(AF_INET, SOCK_STREAM)
connect(c)
...
close(c)
c = socket(AF_INET, SOCK_DGRAM)
...
```

„reciclarea" lui c → c nu e în același timp socket TCP și UDP

Sendto(c, 6000

recvfrom(c

Sendto(c, 6001

bind(s, 6000

recvfrom(s, 6000

if(fork() == 0){

bind(s', 6001

recv

Sendto(s, 6001

}

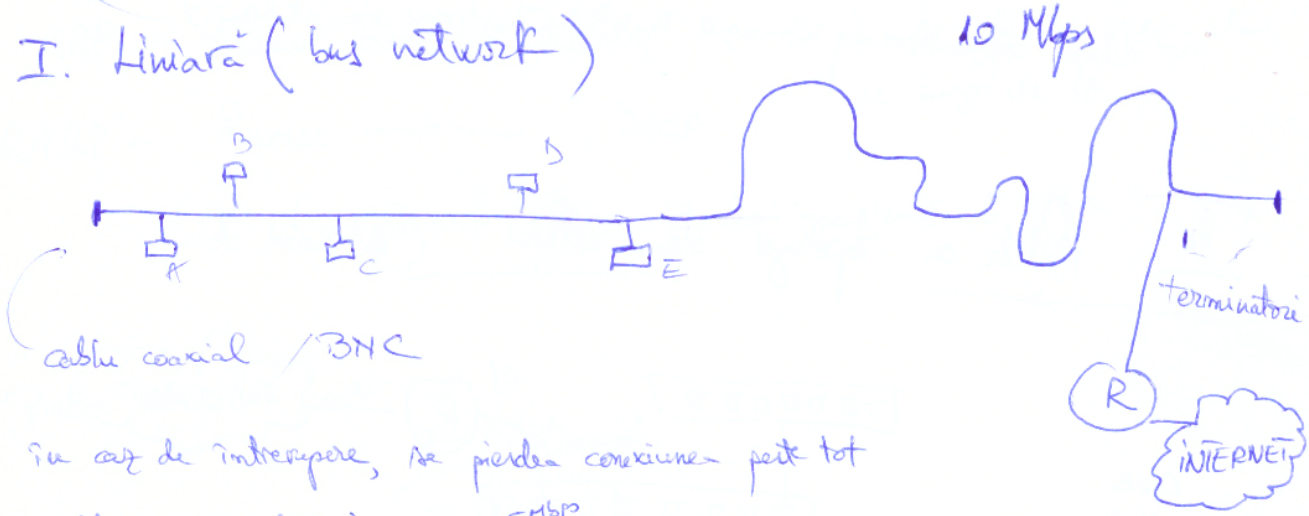
6001

02.11.2022

Cursul 5

Topologii de rețele locale

I. Liniară (bus network)



→ în caz de întrerupere, se pierde conexiunea peste tot

→ probleme de coliziuni $A \xleftrightarrow{5Mbps} B$

$C \xleftrightarrow{5Mbps} D$

→ rețele locale cu difuzare → Broadcast Domain

1 kilobit = 1000 biți

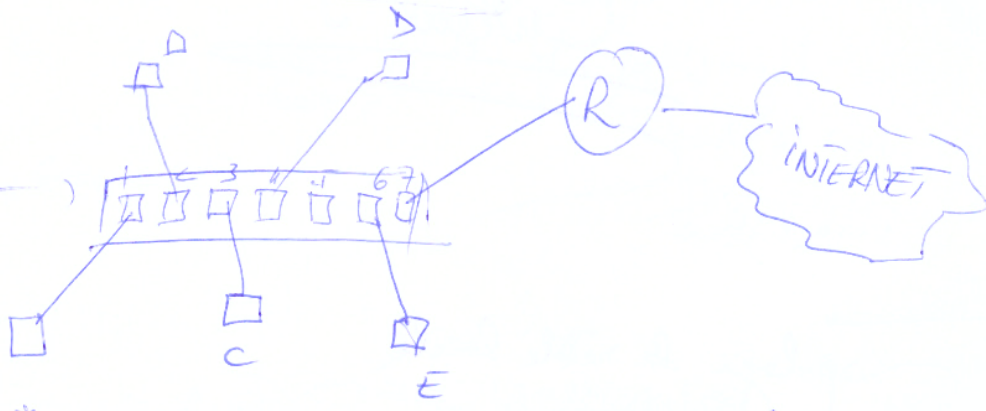
1 megabit = 1000 kilobiți

1 gigabit = 1000 megabiți

1 kilobyte = 1024 bytes

II. Rețele cu topologie stea

~~Hub~~
switch →



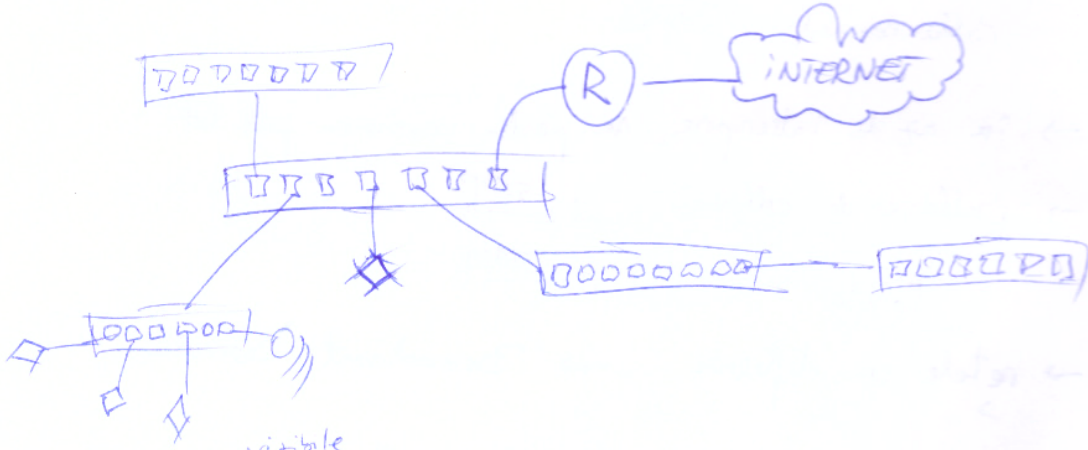
10 Mbps
100

are procesor, memorie
reține un fel de tabelă, care știe cum comunică device-uri, ce frame-uri

→ reduce coliziunile (în cazul hub-ului ăsta mai era problemă)

III. Rețele cu topologie stea extinsă (extended star)

core switches



vizibile la nivel data-link

Adrese MAC

- fizice
- hardware
- ethernet

↓
6 octeți

AB : 7C : 20 : 9E : 14 : 7D

↳ identifică modelul

↳ producătorul

→ între LAN-uri diferite nu pot se văd MAC-urile

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

ARP – Address Resolution Protocol → permite aflarea MAC-ului pe bază de IP

RARP – Reverse —"— , DHCP

↳ permite aflarea IP-ului pe bază de MAC

Pentru setări de bază
- IP
- Netmask
- DNS
- Default Gateway – adresa IP a router-ului
  ↓
  are min. 2 adr. IP

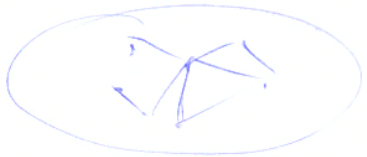
IV Rețele cu topologie de graf incomplet

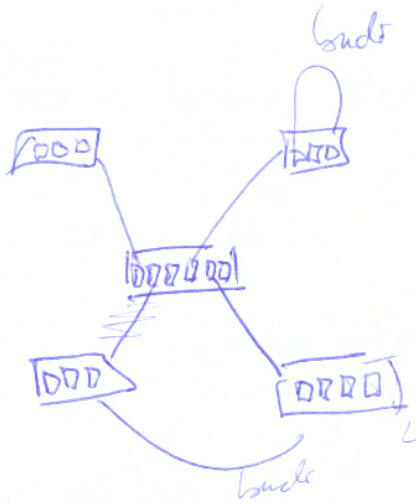
(așa e organizată rețeaua de internet)

LAN – Local
MAN – Metropolitan } Area Network
WAN – Wide

Rețele de rețele de provideri

Zoom in



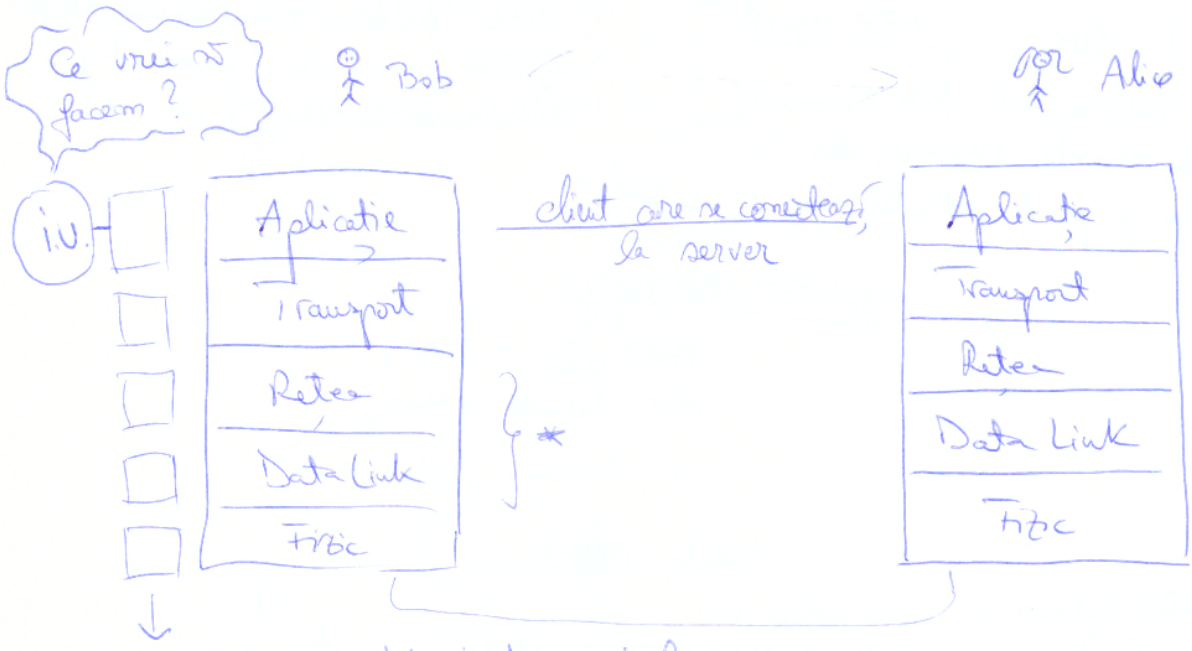


STP – Standing Tree Protocol
– rezolvă problema buclelor
(care pot fi și benefice)

Switch-uri cu management – pe port x port
fără management

09.11.2022

Cursul 6



se adaugă antete la fiecare nivel

- IP
- Network
- Default Gateway
- DNS (Domain Name Server)

Antete

- Nivel
  - transport: protocol (TCP / UDP) + port { sursă, destinație
  - rețea: IP { sursă, destinație
  - data link: MAC { sursă, destinație

arp -n (Linux)
-a (Windows)

tcpdump (arată ARP request-urile, nu și reply-urile)

Bob ⟷ Alice

B ⟷ A

sursă: 192.168.1.17 (frame)

destinație: 192.168.1.39

MAC$_B$ —ARP request→

MAC destinație — se adresează tuturor
FF:FF:FF:FF:FF:FF (broadcast)
MAC sursă = MAC$_B$

îşi descoperă MAC-ul, dar nu îl pe al destinației, aşa îl trimite

←ARP reply—

MAC sursă = MAC$_A$
MAC destinație = MAC$_B$

---

rețele diferite

B ⟷ A

sursă: 192.168.1.17

destinație: 80.81.82.83

—ARP request→ 192.168.1.1.

←ARP reply—

192.168.1.1
îmi pune MAC-ul

B

minimum 2 adrese IP ← - - - - R — INTERNET

Default Gateway
↓
adresa IP a routerului, prin care e conectat la rețeaua locală

Când se dorește transmiterea informațiilor în afara rețelei locale, pachetele se trimit router-ului (Default Gateway). Pentru aceasta, rețeaua află (prin ARP request) MAC-ul routerului.

Rețea
DataLink

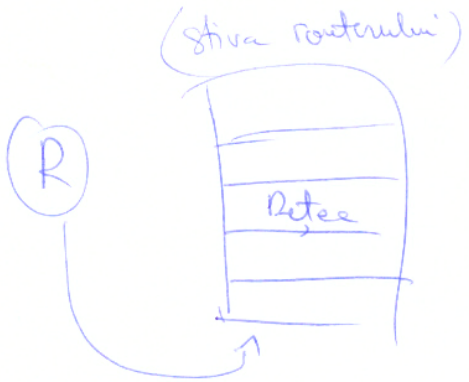
IP_{dest} = 80.81.82.83

MAC_{dest} = MAC(pt IP 192.168.1.1.)

RARP → determină IP pe baza MAC

↳ a fost înlocuit de DHCP (Dynamic Host Configuration Protocol)

- IP
- Network
- Default Gateway
- DNS



→ urcă până la nivelul rețea și observă că adresa IP a destinației nu corespunde cu niciunul dintre IP-urile routerului

⇓

Se fac ARP request-uri repetate, din router în router, până se ajunge la 80.81.82.83

! Obs: La nivelul rețelei locale nu este necesar nici accesul la internet, nici existența unui router!

· pot să am 2 (servere) adrese DHCP în aceeași rețea

↳ da trebuie să am adrese IP din aceeași clasă cu cele de pe placa de rețea internă a routerului

ipconfig /all → ne zice și adresa serverului DHCP

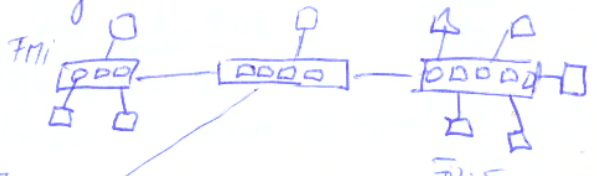
16.11.2022

Cursul 7

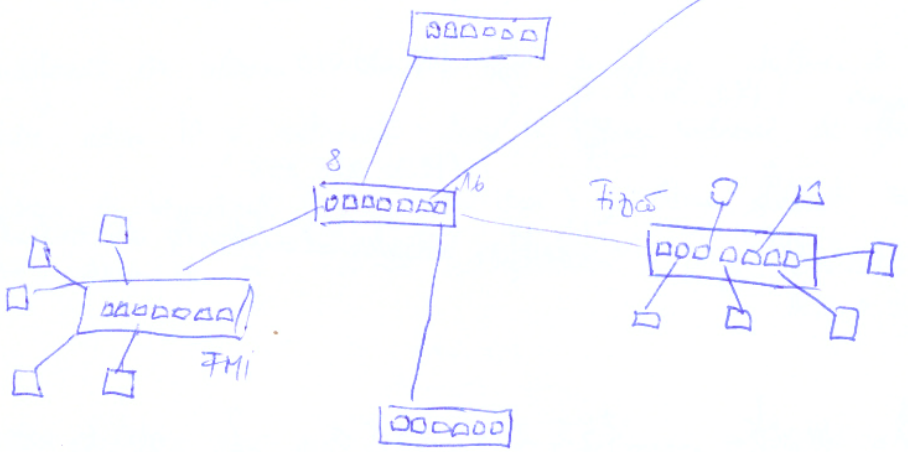
Virtual

VLAN - uri

→ a nu se confunda cu VPN

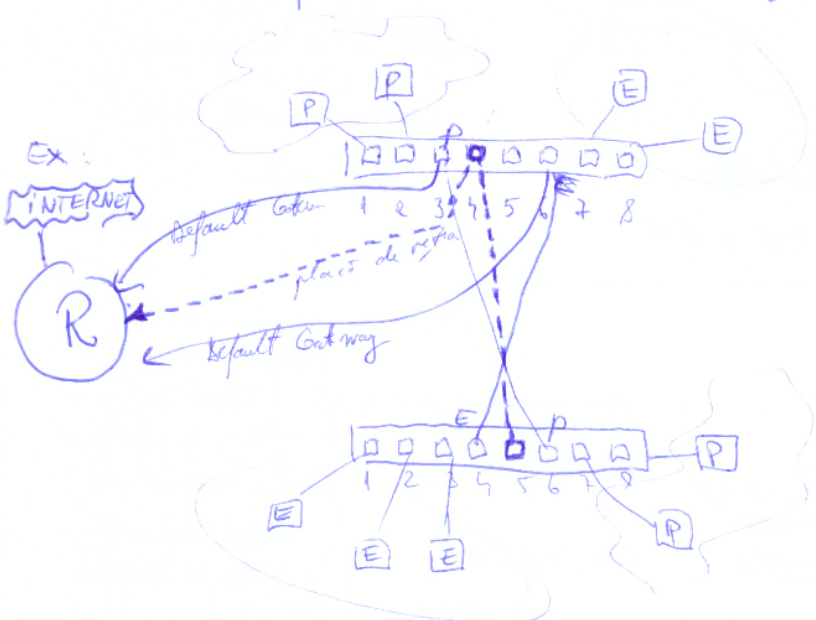


VLAN ID



→ fiecărei port din switch-uri îi este asignat un VLAN ID (se recunosc mai ușor prin nume)

Ex:



(versiunea elementară / rudimentară, unde sunt necesare 2 cablări între cele 2 switchuri, fiind 2 categorii de useri E, P)

Soluție: - - - -

porturile legate astfel pot aparține mai multor VLAN-uri

port în mod Run

! → model mai sensibil din punctul de vedere al securității

Firewall – un fel de router care filtrează pachetele și permite sau respinge realizarea transportului de date

Clase de adrese

$\underbrace{192.168.1.}_{\text{adresa rețelei}} X \begin{cases} 0 \\ \vdots \\ 255 \end{cases}$

↳ toate device-urile conectate la aceeași rețea au în comun primii trei octeți, diferențierea lor făcându-se prin ultimul octet

în campus $\underbrace{172.30.}_{\text{adresa rețelei}} \underbrace{x.y}_{} \begin{cases} 172.30.0.0 \\ \vdots \\ 179.30.255.255 \end{cases}$

↳ identifică adresele calculatoarelor din cadrul rețelei

Subnet mask

$\overbrace{255.255.255}^{192.168.1}.0$

$\underbrace{255.255}_{172.30}.0.0$

} știind câți octeți sunt 255, atâtia mapează adresa rețelei, respectiv octeți de 0 desemnează adresa device-urile din rețea

Ex: 10.1.2.3
10.17.253.242 } → netmask 255.0.0.0

192.168.1.17 → 192.168.1.135
255.255.255.0 — se face shiftare de biți pe netmask — 255.255.255.0
192.168.1.0 — 192.168.1.0

◇ fiind aceeași adresă de rețea, se poate face direct un ARP request către destinatar

adresă de rețea → este întotdeauna prima adresă IP din bază (nu poate fi folosită și setată pentru un alt device)

192.168.1.17 ——————→ 80.81.82.83

255.255.255.0 255.255.255.0

shiftarea pe netmask se face tot netmaskul sursei

192.168.1.0 ≠ 80.81.82.0

↳ fiind diferite, se face ARP request către router

! Obs: ◇ 192.168.1.0 → nu se folosește, fiindcă este adresa rețelei

1.1 → prima adresă care poate fi folosită

1.255 → nu se folosește, deoarece este adresă de broadcast

◇ indiferent de adresa IP dintr-o clasă de adrese, shiftarea de biți dintre adresa IP și netmask duce la aflarea adresei de rețea

◇ adresa de broadcast (ultima din clasă) se obține făcând sau pe biți între adresa IP și not-ul netmaskului (pt. 255.255.255.0 este 0.0.0.255)

192.168.1.0 / 255.255.255.0 → exprimarea vizuală a clasei de rețea este formată din adresa de rețea și netmask

192.168.1.0/ 24

172.30.0.0/ 16

10.0.0.0 / 8

23.11.2022

Cursul 8

		Clasă
192.168.1.0 / 255.255.255.0 → adresă de rețea, prima /24 → alternativă de notație pt. netmask 192.168.1.255 → adresă de broadcast, ultima	→ 2^8 adrese IP	C
172.30.0.0 / 255.255.0.0 172.30.255.255 / 16	→ 2^{16} adrese IP	B
10.0.0.0 / 255.0.0.0 10.255.255.255 / 8	→ 2^{24} adrese IP	A

$\left\{ \begin{array}{l} 0.0.0.0 \\ 255.255.255.255 \end{array} \right.$ 2^{32}

0.0.0.0 – INET ADDR ANY

127.0.0.1

↳ spațiu împărțit în clase de adrese A, B, C

223.255.255.255 → ultima adresă IP validă pentru utilizare

> 224

$\left\{ \begin{array}{l} 0000\ 0000. \\ 0111\ 1111. \end{array} \right. \Rightarrow \left\{ \begin{array}{l} 0. \\ 127. \end{array} \right. \Rightarrow$ clase de tip A (primul bit din primul octet = 0)

	Rețea		Broadcast
A	~~0.0.0.0 / 255.0.0.0~~	; 0.x.y.z	0.255.255.255
	1.0.0.0 / 255.0.0.0	; 1.x.y.z	1.255.255.255
	2.0.0.0 / 255.0.0.0	; 2.x.y.z	2.255.255.255
	⋮		
	126.0.0.0 / 8		
	~~127.0.0.0 / 8~~	→ nu se folosesc	

$\begin{cases} 10\,00\;0000 \\ 10\,11\;1111 \end{cases} \Rightarrow \begin{cases} 128. \\ 191. \end{cases} \Rightarrow$ <u>clase de tip B</u> (primii 2 biți din primul octet sunt 10)

128.17.0.0 / 255.255.0.0 → poate fi și 128.17.35.0 / 255.255.255.0

140.255.0.0/16

172.30.0.0/16

în general ne vom referi la aceasta ca fiind de clasă C, pentru a exprima dimensiunea clasei, nu valoarea primului octet

$\begin{cases} 110\,0\;0000 \\ 110\,1\;1111 \end{cases} \Rightarrow \begin{cases} 192. \\ 223. \end{cases} \Rightarrow$ <u>clase de tip C</u> (primii 3 ~~octeți~~ biți din primul octet 110)

192. . .

223.255.255.255

224. ... → clase experimentale : D → 1110

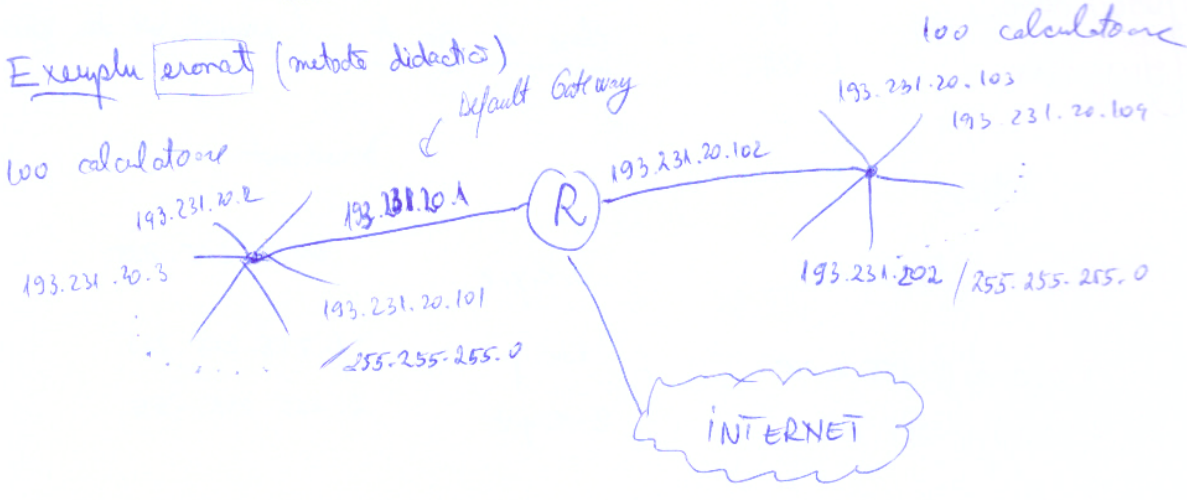
E → 11110

...

193.231.20.0 / 24

255.255.255.0

<u>Exemplu</u> $\boxed{\text{eronat}}$ (metodă didactică)



S.		d.
193.231.20.2 l	⟶	193.231.20.110 l
255.255.255.0		255.255.255.0
193.231.20.0		193.231.20.0

⟶ sursa crede că destinația se află în aceeași rețea cu aceasta

Soluție: împart clasa în 2:

128

193.231.20.0
.0000 0000
⋮
193.231.20.127
.0111 1111

Netmask:

255.255.255.1000 0000

/25

255.255.255.128

128

193.231.20.128
.1000 0000
193.231.20.255
.1111 1111

Netmask:

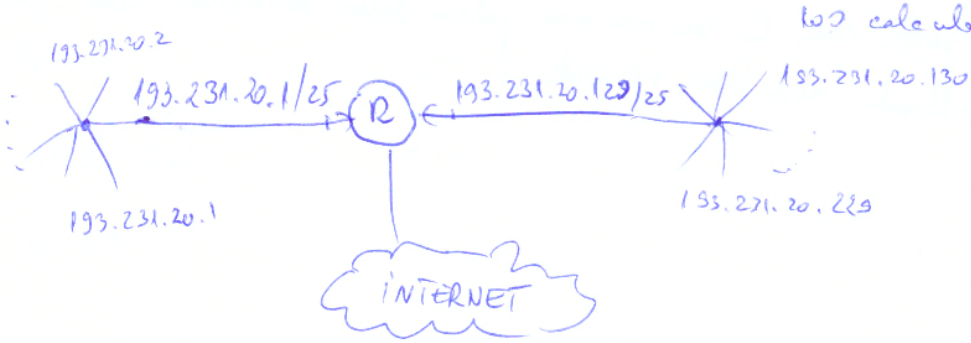
255.255.255.128

⟶ adresele de aici au în comun primii 25 de biți, diferențiindu-ne prin ultimii 7 biți

193.231.20.0

193.231.20.0/255.255.255.128 | 193.231.20.128/25

100 calculatoare | 100 calculatoare



193.231.20.2 & ⟶ 193.231.20.130 &

255.255.255.128 ⟶ tot net mask-ul sursei

193.231.20.0 ≠ 193.231.20.128

⇒ sursa înțelege că destinația se află în altă rețea

193.231.20.0 / 255.255.255.0

- 0 … 127 — /255.255.255.128 /25
  - 0 … 63 — 64 de adrese
  - 64 … 127 — 255.255.255.192 /26
- 128 … 255 — /255.255.255.128 /25
  - 128 … 191
  - 192 … 255

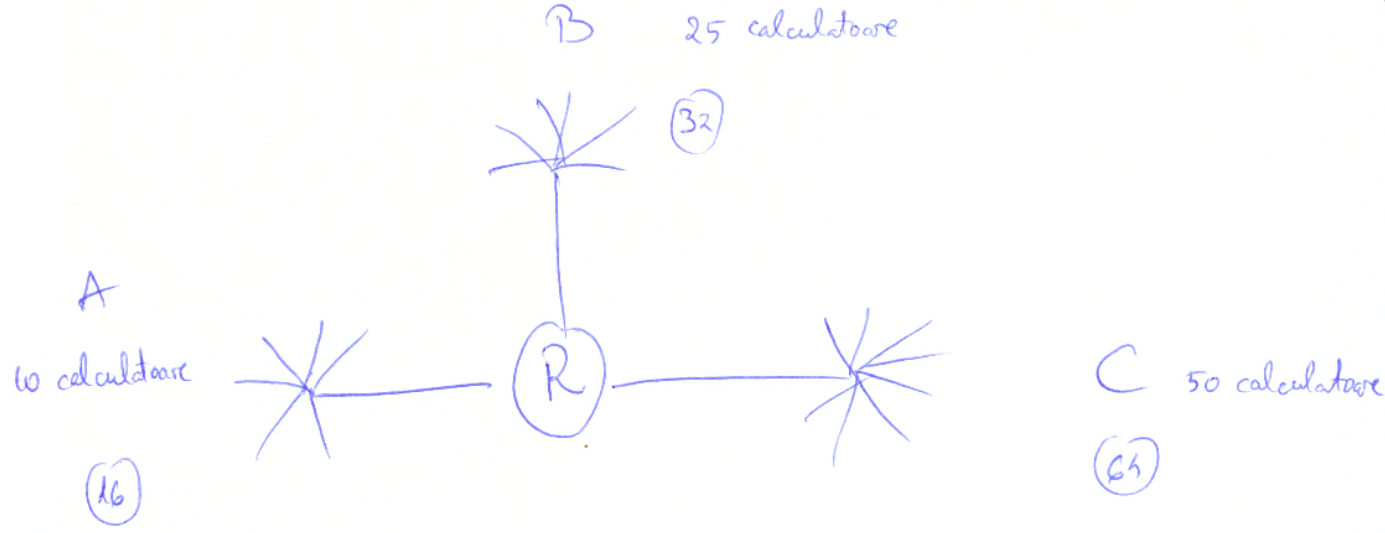
→ fiecare dintre cele 4 subclase a câte 64 de adrese are adresă de rețea și adresă de broadcast

0 … 63 — 0 … 31 (32), 32 … 63 (32) — /255.255.255.224 /27 ...

subclase de 4: 0 … 3 (4), 4 … 7 (4) — /255.255.255.252 /30

! Adresa de rețea este multiplu de dimensiunea clasei !

- putem împărți clasa în 3? → da, dar nu de dimensiuni egale (ex: împart în 2 jumătăți, iar pe prima o împart din nou în 2)
- pentru clase C sau mai mici putem deduce dimensiunea unei (sub)clase scăzând din 256 ultimul octet din net mask



Se dă clasa 193.231.20.0/24

A 193.231.20.0 → Rețea /255.255.255.240
20.15 → broadcast /28

× B 193.231.20.16 /255.255.255.224
20.47

De ce e greșit?

⇒ 193.231.20.17 & → 193.231.20.33 &
255.255.255.224 255.255.255.224

.00010001 .00100001
.11100000 .11100000

193.231.20.0 ≠ 193.231.20.32

B 193.231.20.32 / 27
20.63

C 193.231.20.64 / 26
20.127

07.12.2022

Cursul 9

193.231.20.0 / 255.255.255.0
/ 24 → numărul de biți de 1

193.231.20.0 / 255.255.255.128
/25
193.231.20.128 / 255.255.255.128
/25

193.231.20.0 / 255.255.255.192
/26
193.231.20.64 /26
193.231.20.128 /26
193.231.20.192 /26

...

Netmask ⇒ câți biți au în comun toate adresele de la aceeași rețea
(partea comună din adresa IP = adresa rețelei)

→ cea mai mică subclasă pe care o pot obține este de 4 biți
$4 = 2^2 \Rightarrow 32-2 = 30$
/ 255.255.255.252
- 193.231.20.0/30
- 193.231.20.4/30
- 193.231.20.8/30

...

193.231.20.0 → adresă de rețea
193.231.20.1
193.231.20.2
193.231.20.3 → adresă de broadcast

la fel

0000 00	00
0000 00	01
0000 00	10
0000 00	11

⇒ 2 biți diferiți
↓
netmask /30

↳ sunt 30 de biți la fel ⇒ netmask 255.255.255. $\underbrace{1111\ 1100}_{252}$

Exemplu pb. examen

5 calculatoare

! 193.231.20.15 (trebuia să fie adresa unui calculator)

⇒ dacă am alege să divizăm în câte 8 ⇒ greșit

.15 → broadcast în clase de 16

Obs.: o clasă trebuie să înceapă la multiplu de dimensiunea clasei

→ Pentru clase mai mici de 256, dacă dorim să aflăm netmaskul, scădem din 256 dimensiunea subclasei

ex: 256 − 256 ⇒ 0 → 255.255.255.0
256 − 128 = 128 → 255.255.255.128
256 − 64 ⇒ 192 ⇒ 255.255.255.192
⋮

clasă de 256
193.231.20.0 / 255.255.255.0
/24

clasă de 256
193.231.21.0 / 24

clasă de 512 ⟹ 193.231.20.0 / 23 ← clasă agregată
agregare

193.231.20.0 ← rețea
193.231.21.255 ⇐ broadcast

cum îl aflăm?

$512 = 2^{(9)} \Rightarrow 32 - 9 = \underline{23}$

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

$\rightarrow 2^{10} \Rightarrow 32-10 = \underline{22}$

1024 adrese

adresa de rețea

$\Rightarrow$ 193.231.20.0/22
/255.255.252.0

$\rightarrow$ 193.231.21.0/24
193.231.22.0/24 $\rightarrow$ nu se poate face agregare

193.231.21.0/23 e greșit, pentru că nu e multiplu de 512

! O clasă de n trebuie să înceapă la multiplu de n

193.231.21.0000 0000
1 0000 0000
↓
256

193.231.21.0000 0000
10 0000 0000
↓
512

nu se împarte

X.y.z.0
2^8 $\rightarrow$ clasa de 256 se împarte la 2^8

2 biți de 0
193.231.20.0
3 biți de 0
1024 $\rightarrow$ se împarte la 2^{10}

$\rightarrow$ în scrierea binară a lui 20: 0001 0100 $\rightarrow$ 2 biți de 0

$\Rightarrow$ poate începe o clasă de maxim 2^{10}

Le. clasă de 32 $= 2^5 \Rightarrow 32-5 = 27 \rightarrow$ netmask 255.255.255.224

193.231.20.17 &
255.255.255.224
193.231.20.0

193.231.20.33 &
255.255.255.224
193.231.20.32

nu aparțin aceleiași clase de 32

193.231.21.0 ? o putem seta pe un calculator!

↳ nu se știe dacă e adresă de rețea până când nu se cunoaște și netmask-ul

↳ ar putea fi parte dintr-o clasă mare și să fie undeva pe la mijloc

193.231.21.255

↳ aceeași poveste ca mai sus

20.0.0.0 / 8
/ 255.0.0.0

împărțim în subclase de 2^{16}

⇒ 20.0.0.0 ... 20.0.255.255 → 20.0.0.0/16
/255.255.0.0

20.1.0.0 /16
20.2.0.0 /16
20.3.0.0 /16

2^{16} adrese

20.254.0.0 /16
20.255.0.0 /16

20.3.0.0 /16 → o rup în 2 subclase

20.3.x.y

⇓

20.3.0.0 /17 și 20.3.128.0/17

20.3.0.0 ..... 20.3.127.255 ← 2^{15} /17

20.3.128.0 .... 20.3.255.255 ← 2^{15} /17

↳ au în comun primii 17 biți

⇓

255.255.$\underbrace{1000\ 0000}_{128}$.0

/17

Exercițiu:

[20.21.22.48 ... 192.168.17.95]

↳ să fie adresă de rețea ↳ să fie adresă de broadcast

• să se scrie intervalul ca reuniune de clase de adrese (maximale)

{ 0.0.0.0 → cea mai mică adresă ⇒ 0.0.0.0 / 0.0.0.0 / 0

255.255.255.255 → cea mai mare

$$\begin{array}{l} x.y.z.t \ \& \\ \underline{0.0.0.0} \\ 0.0.0.0 \end{array}$$

}

→ 20.21.22.48 / 255.255.255.240 (256-16)
/ 28 (32-4, $2^4=16$)

↓ cea mai mare clasă care poate conține e de la 16 (48 : 16)

→ 20.21.22.64 / 255.255.255.192
/26
↳ poate conține clasă de 64

→ 20.21.22.128 / 255.255.255.128
/25
↳ poate conține clasă de 128

... până la 20.21.22.255

→ 20.21.23.0 /24

↳ poate conține o clasă de 256, iar nu mai mare (23 impus)

→ 20.21.24.0 /21

/255.255.248.0

↳ clasă de max. 2^{11}, 24 = 0001 1000

3+8 = 11

$$8 \left\{ \begin{array}{l} 20.21.24.X \\ 20.21.25.X \\ 20.21.26.X \\ \dots \\ 20.21.31.X \end{array} \right. \rightarrow 8 \cdot 256 = 2^3 \cdot 2^8 = 2^{11}$$

→ 20.21.32.0 / 19

↳ clasă de max. 2^{13}

$$32 \left\{ \begin{array}{l} 20.21.32.X \\ 20.21.33.X \\ \dots \\ 20.21.63.X \end{array} \right. \rightarrow 32 \cdot 256 = 2^5 \cdot 2^8 = 2^{13}$$

→ 20.21.64.0 / 13

↳ clasă de max. 2^{14}

$$64 \left\{ \begin{array}{l} 20.21.64.X \\ \dots \\ 20.21.127.X \end{array} \right. \rightarrow 64 \cdot 256 = 2^{14}$$

→ 20.21.128.0 / 17

↳ clasă de 2^{15}, jumătate de clasă B

$$128 \left\{ \begin{array}{l} 20.21.128.X \\ \dots \\ 20.21.255.X \end{array} \right. \rightarrow 128 \cdot 256 = 2^{15}$$

20.21.255.255 → broadcast

→ 20.22.0.0 / 15
/ 255.254.0.0 → 2 clase B

→ 20.24.0.0 / 13
→ 8 clase B ⇒ $8 \cdot 2^{16} = 2^{19}$, $32-19 = 13$

20.31.255.255 → broadcast

→ 20.32.0.0 / 11
→ $32 \cdot 2^{16} = 2^5 \cdot 2^{16} = 2^{21}$, $32-21 = 11$

→ 20.64.0.0 / 10 ← 256−64 clase B = 192
/ 255.192.0.0

→ $64 \cdot 2^{16} = 2^{22}$

→ 20.128.0.0 / 9
→ 128 clase B , $128 \cdot 2^{16} = 2^{23}$

→ 21.0.0.0 / 8 (2^{24}
→ 1 clasă A

→ 22.0.0.0 / 7 (2^{25}
→ 2

→ 24.0.0.0 / 5 ——— 256 − 8 clase A = 248 Obs: nu avem network are 255
/ 248.0.0.0
→ 8 clase A , $8 \cdot 2^{24} = 2^{27}$

→ 32.0.0.0 / 3 → 32 clase A
→ 64.0.0.0 / 2 → 64 clase A
/ 192.0.0.0

→ 128.0.0.0 / 2 → pentru a mai depăși 192.168.17.95 iau clase cu 64, iar mai cu 128

→ 192.0.0.0 / 9

↳ jumătate de clasă A / 128 clase B

→ 192.128.0.0 / 11

↳ pot să iau 32 clase B → $32 \cdot 2^{16} = 2^{21}$, $32 - 21 = 11$

192.159.255.255 → broadcast

→ 192.160.0.0 / 13

↳ pot să iau 8 clase B → $8 \cdot 2^{16} = 2^{19}$, $32 - 19 = 13$

192.167.255.255 → broadcast

→ 192.168.0.0 / 20

↳ pot să iau 16 clase C → $16 \cdot 2^{8} = 2^{12}$, $32 - 12 = 20$

→ 192.168.16.0 / 24

↳ clasă simplă

→ 192.168.17.0 / 20

↳ pot începe cu 64

→ 192.168.17.64 / 27
/ 255.255.255.224

↳ clasă de $32 = 2^5$, $32 - 5 = 27$

⇒ $\boxed{192.168.17.95}$ → broadcast

→ dacă aș fi avut .96 ⇒ 192.168.17.96 / 32
/ 255.255.255.255 → clasă de o singură adresă IP sau (nu e clară)

Problemă: (1) 80. 81. 82. 83
82. 83. 84. 85

- Precizați cea mai mică posibilă clasă de adrese, care să le conțină pe ambele (cu cel mai mic număr de adrese IP; netmask cu cât mai puțini biți de 0)

80 = 0101 0000 . 81. 82. 83

82 = 0101 0010 . 83. 84. 85

→ netmask: 1111 1100 . 0.0.0 ← 6 biți de 1, 26 biți de 0

/ 252.0.0.0

/ 6

→ dacă facem & pe biți ⇒ clasa 80.0.0.0 ← adresa de rețea
83.255.255.255 ← adresa de broadcast

răspuns: 80.0.0.0/6

(2) În câte clase cu netmask 255.255.255.252 se împarte clasa de dimensiune maximă, care are aceeași adresă de broadcast cu clasa de dimensiune maximă, care începe la adresa 193.244.0.0 ?
↓
multiplu de 4

→ 193.244.0.0 ← adresă de rețea

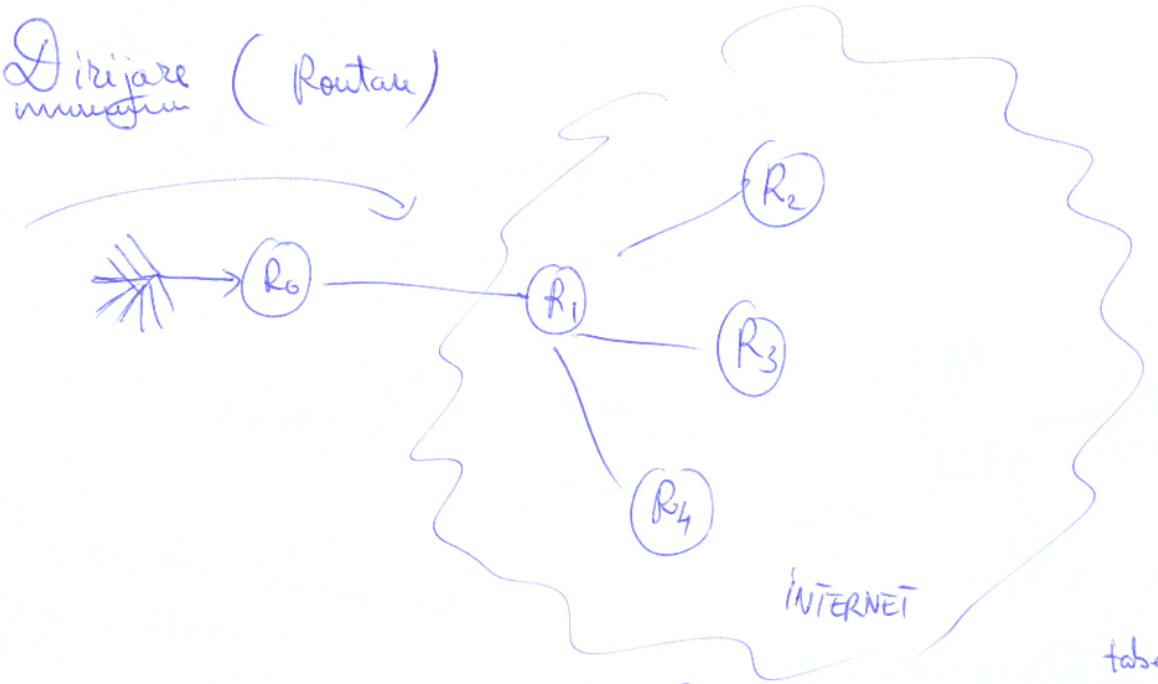
→ 193.247.255.255 ← adresă de broadcast

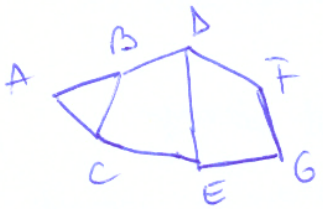
193.248
↳ multiplu 8 → clasă de 8 agregate

14.12.2022

Cursul 10

Dirijare (Routare)





$A \xrightarrow{?} G$

tabelă de dirijare

(A)

Dest	Vecin	Lungime / Cost
A	A	0
B	B	1
C	C	1 ← rută
D	B	2
E	C	2
F	B	3
G	C	3

Algoritmi de dirijare

↓

Protocoale de dirijare
(implementate și rulate de Routere

a) vectori distanță (Bellman-Ford) : RIP (Routing Internet Protocol)
b) starea legăturilor (Dijkstra) : BGP (Border Gateway Protocol)

o stare inițială (primul rând)

a) Schimb de informație între tabelele de dirijare ale routerelor, pornind de la

A:

Dest	Vecini	Lungime
A	A	0
B	B	1
C	C	1

↔

B:

Dest	Vecini	Lungime
B	B	0
A	A	1
D	D	1

C:

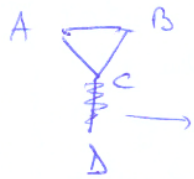
Dest	Vecini	Lungime
C	C	0
A	A	1 (0+1 → cât e de la A la C)
~~B~~	~~A~~	~~2~~ (1+1
B	B	1 ✓

↳ a găsit un drum mai bun și îl elimină pe cel vechi

- Tabelele de dirijare se actualizează în cazul în care:
  - → se găsește o rută mai bună
  - → se pierde o legătură

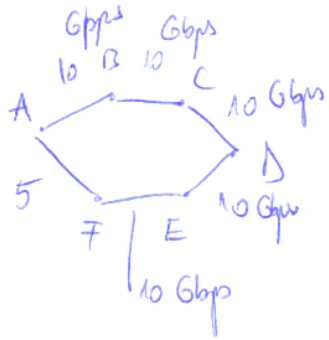
  } → se notifică și vecinii, pentru a exista sincronizare între tabelele de dirijare ale routerelor

↳ problema numărării la infinit (suplimentar)



→ pică legătura, A → D : 3, C → D : 4 ...

ex:



A:

Dest	Vecin	Lungime
F	?	

RIP ar zice:

F 1

folosit doar didactic

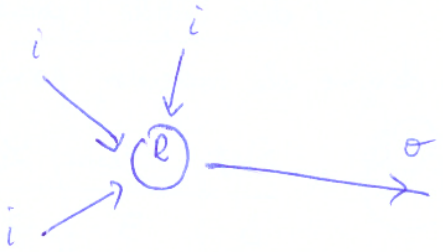
BGP ar zice:

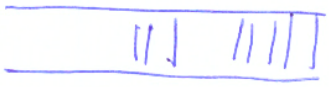
B 5

folosit în practică

(viteze mai mare de transmitere a datelor

BGP → ia în calcul și viteza de transmitere, și aglomerarea rutelor





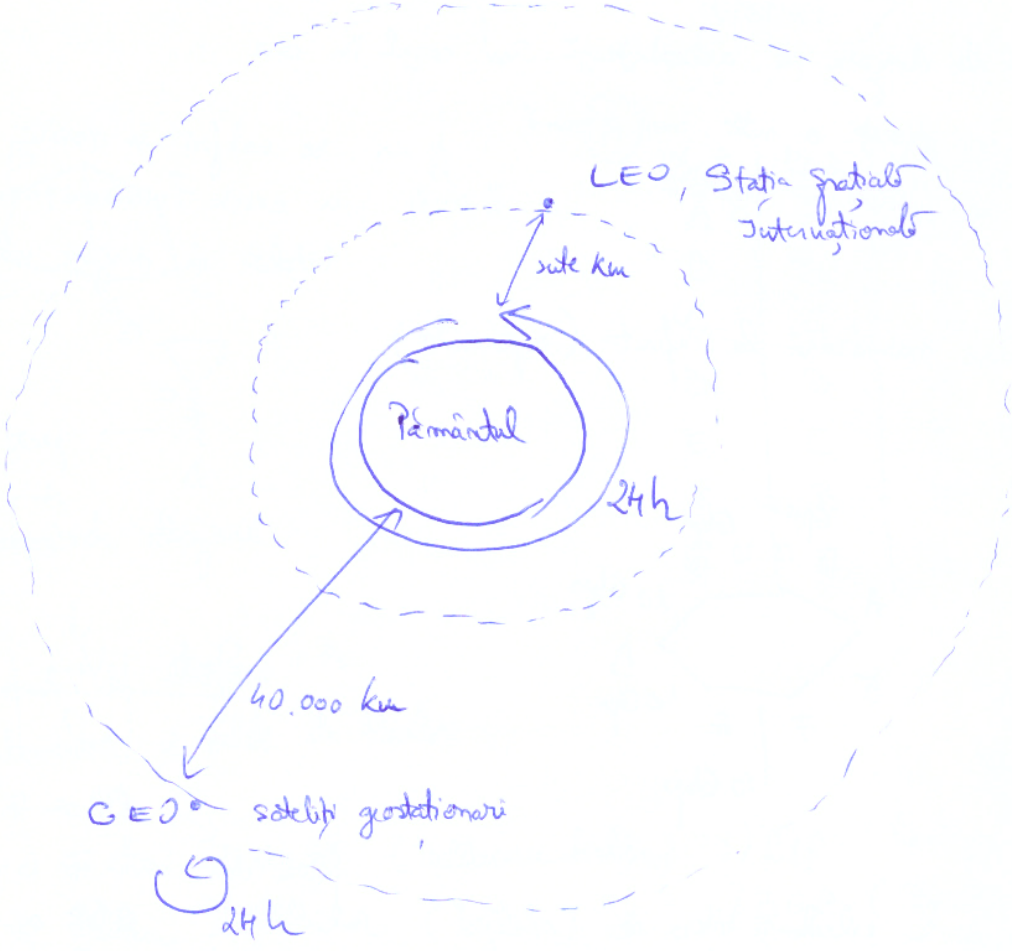
↳ un fel de buffer / o coadă a routerului

– cu cât e mai gol, cu atât e traficul mai fibre

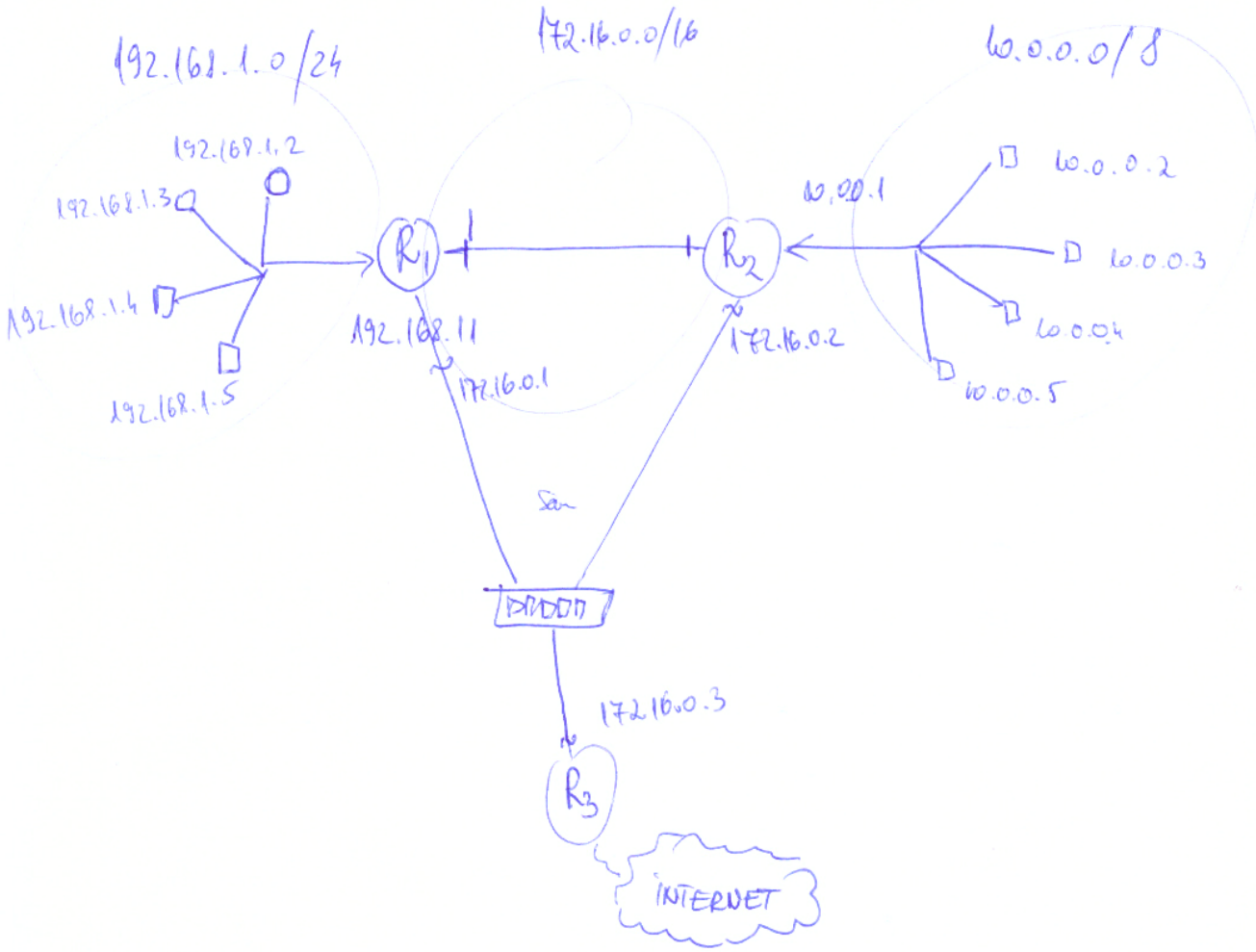
plină, –"– trafic aglomerat

! Nu tot timpul conexiunile cele mai rapide sunt favorizate

→ exemplu: chiar dacă viteza de transmitere a informației între Pământ și sateliți geostationari este viteza luminii, distanța mare face comunicarea mai întârziată (față de comunicarea cu sateliți LEO)



↳ pare că stau pe loc (având același timp de rotație în jurul pământului cu mișcarea de rotație a Pământului)



192.168.1.2 ⟶ 10.0.0.3

(R1):

	Dest	Vecin
S	10.0.0.0/8	R2 172.16.0.2
	0.0.0.0/0	R3 172.16.0.3

→ routerul face & pe biți între adresa ip unde trebuie să ajungă date și toate netmask-urile din tabela de dirijare

(R2):

	Dest	Vecin
S	192.168.1.0/24	R1 172.16.0.1
	0.0.0.0/0	R3 172.16.0.3

↳ rută statică

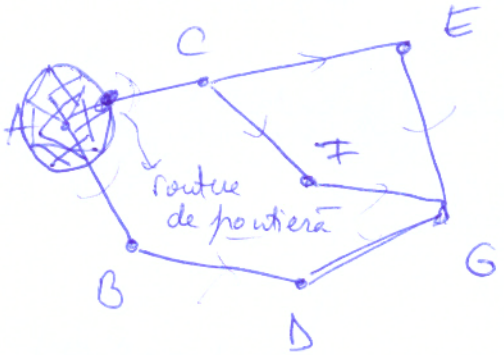
Comanda: route print pe Windows

Dest	Vecin
0.0.0.0/0.0.0.0	192.168.1.2

/0 ← Rută Default

21.12.2022

Cursul 11



A

Destinație	Vecin	Metrică
rute →		
0.0.0.0/0.0.0.0	192.168.1.1	

rută default (↳ "spre oraș") ↳ adresa routerului (default gateway)

Continuare curs 10

b) starea legăturilor

(A, C, 1) timestamp
(A, B, 1) timestamp → pentru cunoașterea celor mai recente informații

↳ informații trimise atât lui B, cât și lui C → acestea trimit mai departe

↳ tehnica inundării

a) VS. b)

a) fiecare R face calcule parțiale

b) fiecare R face toate calculele pentru stabilirea tabelelor de dirijare

mai intens computațional

↓ evită problema numărării la infinit

↓ nu se pretează pe rețele foarte mari

→ în realitate, în interiorul rețelelor providerilor sunt folosiți algoritmii b), iar în exterior (mai general) algoritmii a)

↳ între provideri → BGP
Border Gateway Protocol

↳ protocol de schimb inf. (tabele de dirijare) între routerele de frontieră

Sistemele autonome (AS)

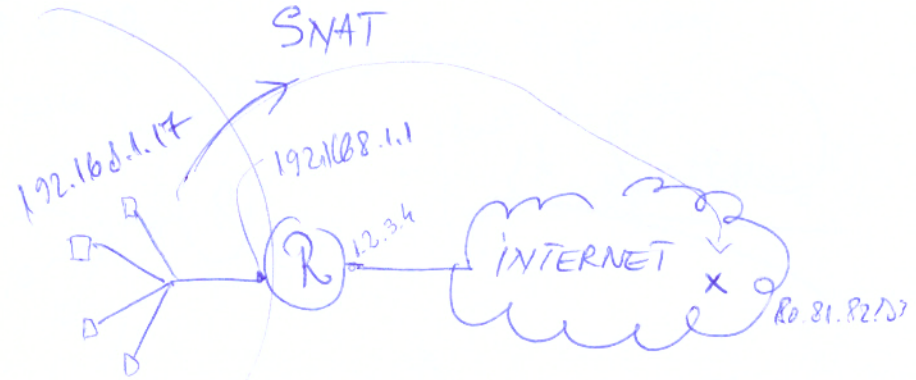
nr. AS → fiecare provider are un nr. AS, cu care se identifică între furnizorii din graf

public route servers / routers servers

ssh / telnet

putty →

traceroute



192.168.1.0/24 → clasă falsă (non-routabilă)

80.81.83.0/24

Clase false/ private

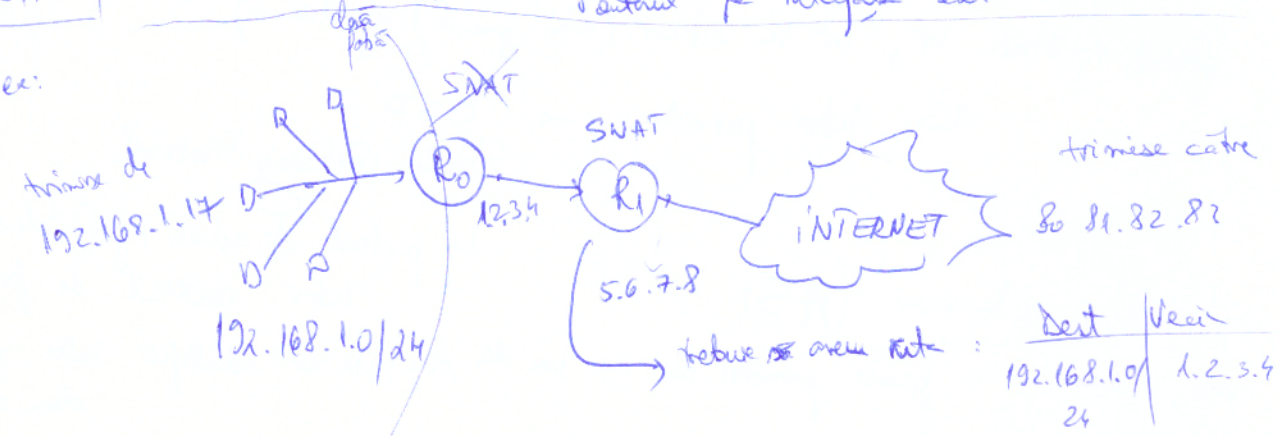
10.0.0.0/8 — 1 clasă A

172.16.0.0/16 … 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 IP privată (a sursei) este schimbată în adresa reală pe care o are routerul pe interfața externă

Pb. examen ex:



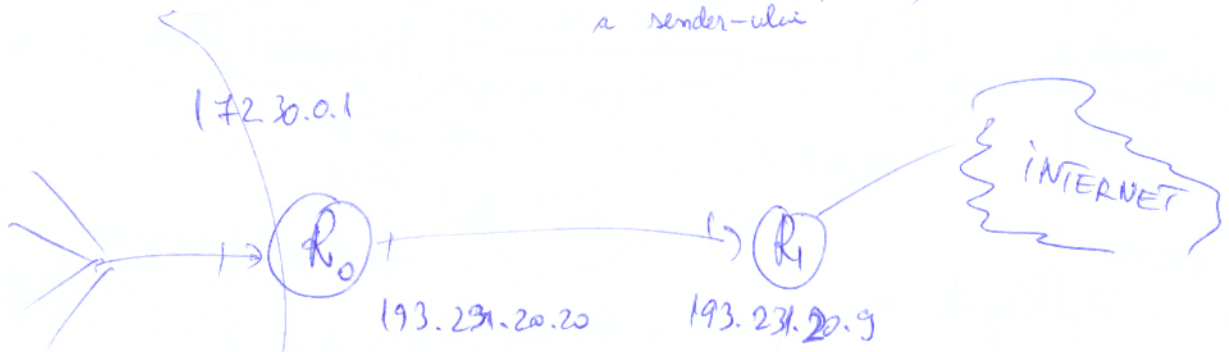
– routerul R0 nu face SNAT. ? au calculatoarele acces la internet?

Ce se poate face ca ele să aibă acces la net?

Răspuns: R_1 trebuie să facă SNAT și să memoreze ruta spre dest falsă

192.168.1.0/24

(↳ deci transformă răspunsul venit pe 5.6.7.8 în adresa ip falsă a sender-ului)



Dest	Vecin
172.30.0.0/16	193.231.20.20

172.30.0.0/16

← nu are setat default gateway

videoproiector

172.30.0.9/16

172.16.0.1/16

172.30.x.y

serverul

172.30.0.4

laptop wireless

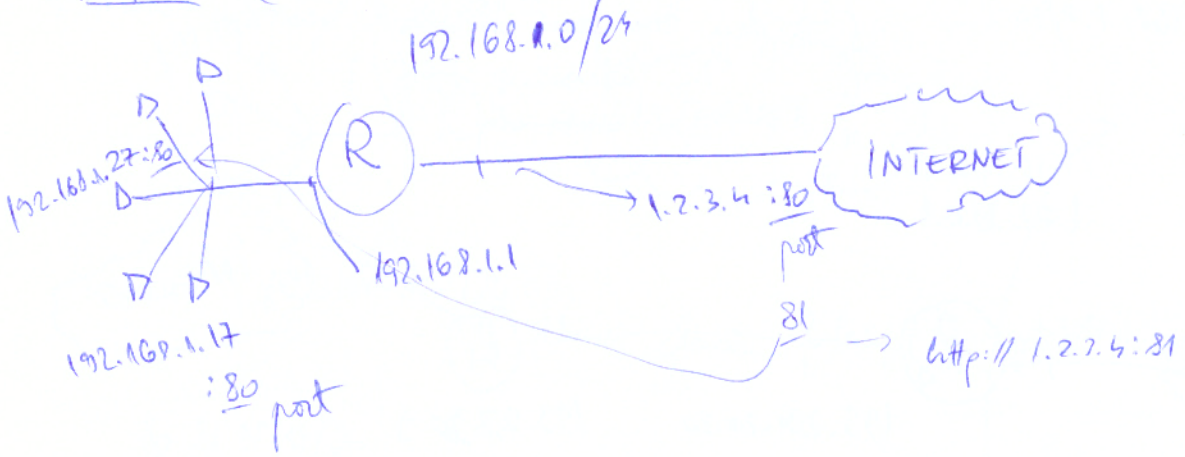
172.16.0.2

↑ o adăugat ruta:

Dest	Vecin
172.16.0.0/16	172.30.0.4

SNAT → așa putea să trimită videoproiectorul date înapoi, căci vede adresa 192.16.0.2 ca fiind din aceeași rețea cu el și serverul putea converti adresa în cea a sender-ului

DNAT (port forwarding / virtual servers) – translatarea adresei destinatie



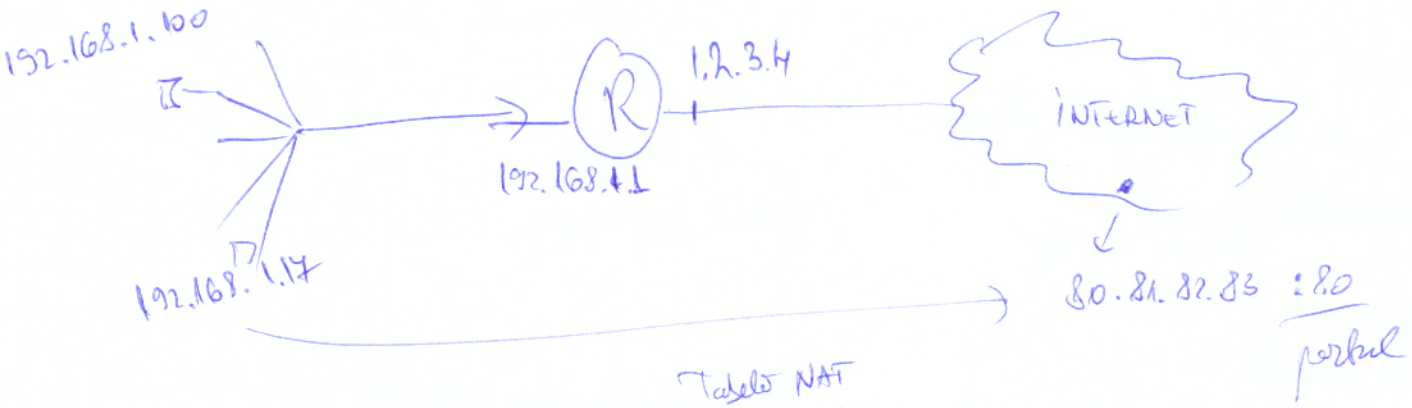


Tabela NAT

Pachet înainte de SNAT

IP sursă	Port sursă	IP dest	Port dest
192.168.1.17	2179	80.81.82.83	80
192.168.1.100	5076	80.81.82.83	80
	2179		

Pachet după SNAT

IP sursă	Port sursă	IP dest	Port dest
1.2.3.4	2179	80.81.82.83	80
1.2.3.4	5076	80.81.82.83	80
	2180		

2179 → îl ignoră pe 80.81.82.83

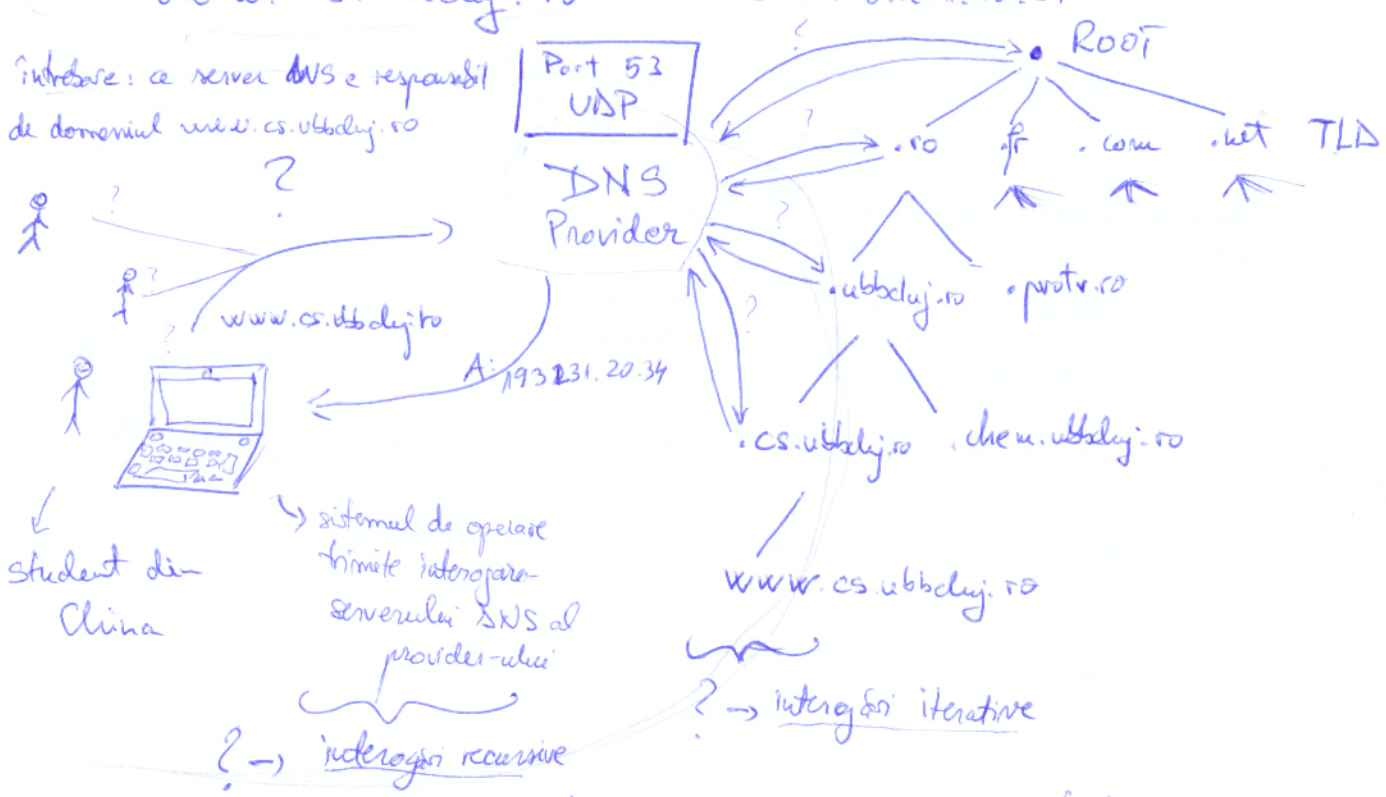
→ se poate întâmpla doar dacă nu am altă conexiune pe portul 2179, dacă routerul are folosit portul 2179 (pentru o conexiune deschisă de el cu exterior)

11.01.2023

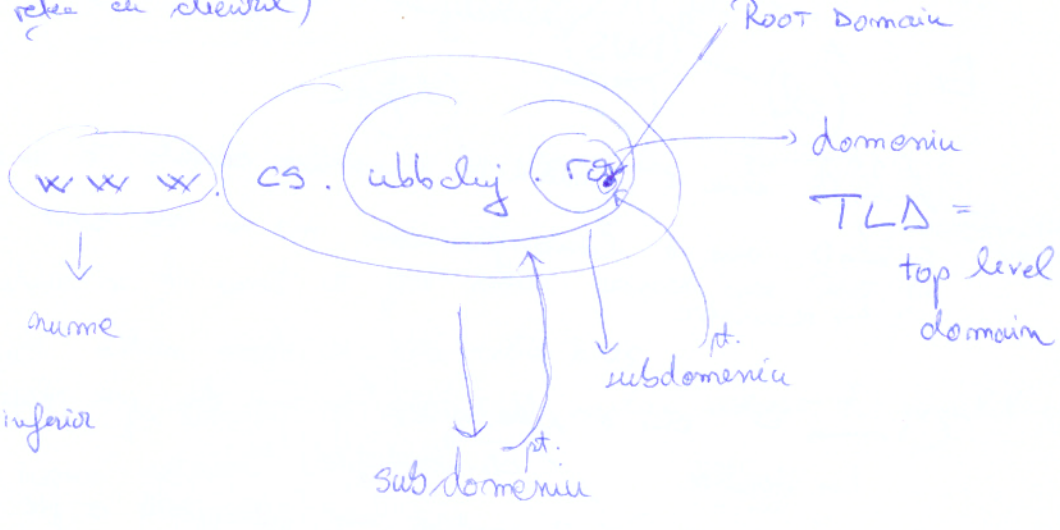
Cursul 12

DNS - Domain Name System

www.cs.ubbcluj.ro $\xrightarrow{A}$ 193.231.20.34



DNS → e plasat la nivel aplicație, ceea ce are ca urmare faptul că serverul poate fi situat oriunde în internet (adică nu neapărat în aceeași rețea cu clientul)



* fiecare nivel superior din arbore este responsabil și reține adresele serverelor de pe nivelul imediat inferior (fii direcți)

3600 s

TTL – Time To Live – mai multe inf. „reținute", „cache"-uite

ipconfig / displaydns – arată adresele reținute, cât timp TTL > 0

/ flushdns – ștergerea informațiilor

↳ după ce dăm un nou ping la www.cs.ubbcluj.ro, nu va apărea TTL de 3600, ci mai mic, pentru că reinterogheazăm serverul DNS al providerului, care, la rândul lui, a reținut TTL și transmite informațiile (adresa IP + TTL) clientului

Responsabilitățile serverului DNS

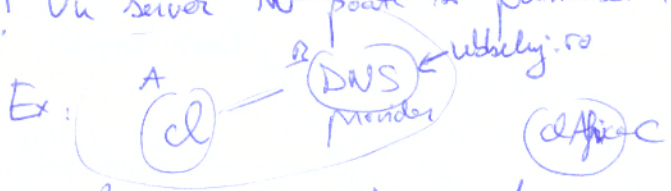
1. deservirea clienților → translatarea numelor de domeniu în adrese IP
2. reținerea adreselor (serverelor DNS) de care este responsabil, adică a celor aflate pe nivelul imediat inferior domeniilor

↳ acestea pot fi pe rând sau ambele deodată

DNS 1.
Name Server 2.

! Un server NU poate să primească întrebări de la oricine pentru orice

Ex:



A $\xrightarrow{?}$ B cine e ferrari.it → are sens

C $\xrightarrow{?}$ B cine e lamborghini.com → nu are sens

↳ recursive

C $\xrightarrow{?}$ B cine e ubbcluj.ro? → da, va răspunde

↳ iterative

cs.ubbcluj.ro

2 condiții → interogările să fie iterative și legate de domeniile pentru care este responsabil serverul DNS

Tipuri de înregistrări DNS

- A → adresă
- AAAA → IPv6
- PTR → traduce adrese IP în nume de calculatoare
- NS → Name Server (cele de sus conțin înregistrări despre cele de jos - în arbore)
- MX → pentru mail
- CNAME → Canonical Name → alias-uri

  ex: www.firma17.ro este un alias pt. hosting.provider.ro
- TXT → orice

firma17.com

? merge → doar dacă ambele traduc, în final, aceeași adresă IP

↳ www.cs.ubbcluj.ro are ca alias cs.ubbcluj.ro

merg ambele { cs.ubbcluj.ro → 193.231.20.34

→ să ziceem că vreau să schimb adresa IP a serverului în 1.2.3.4.

Ce fac?

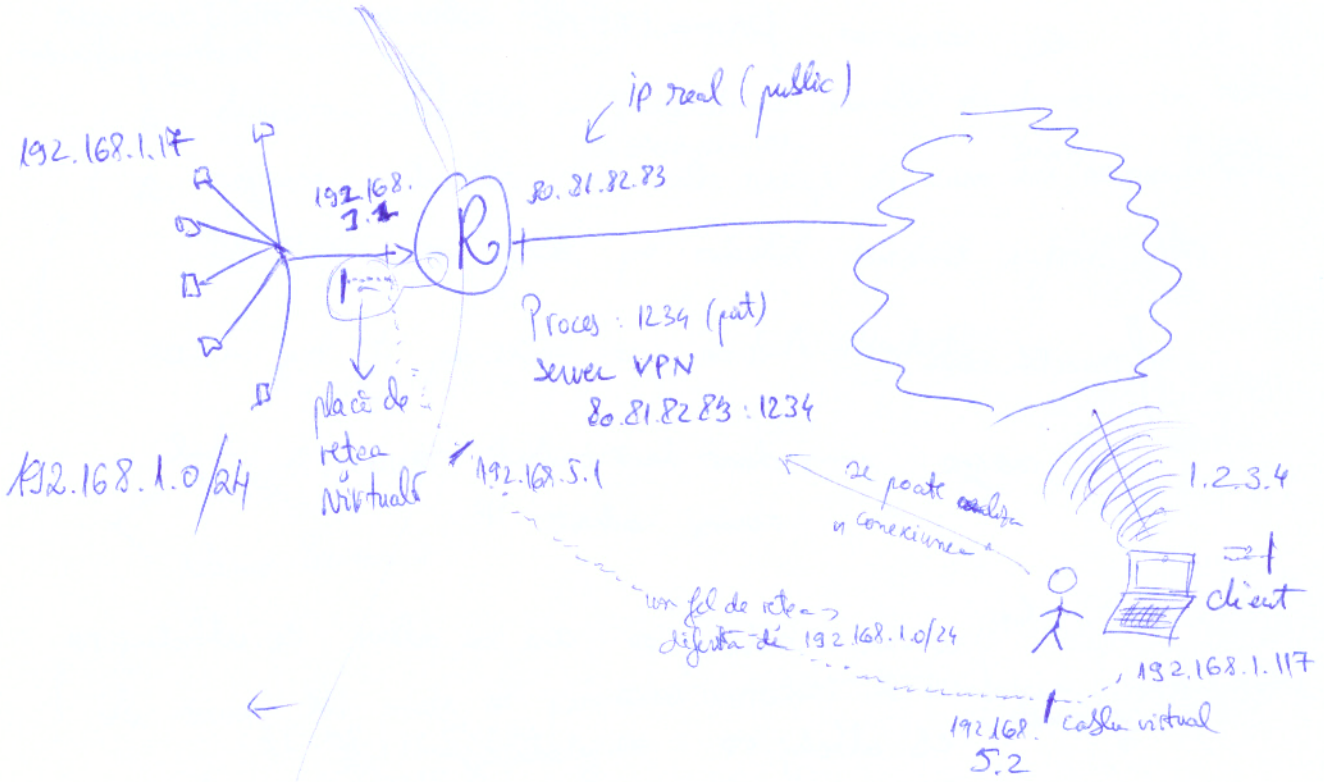
↳ trebuie să fiu cont că pot exista servere care încă în multe minute adresa veche a serverului DNS (multe minute: TTL)

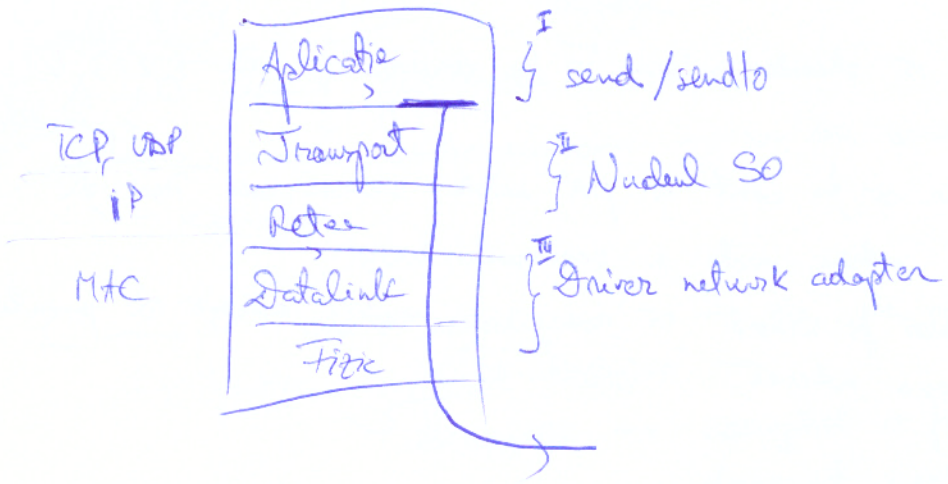
⇒ voi fi nevoit să merg cu vechea adresă în paralel cu noua adresă

18.01.2023

Curgul 13

VPN (Virtual Private Network)





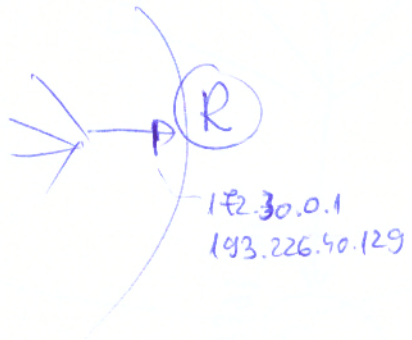
Obs: atât clientul, cât și serverul VPN trebuie să aibă placă de rețea virtuală

• De ce aș pune pe o placă de rețea mai multe adrese IP?

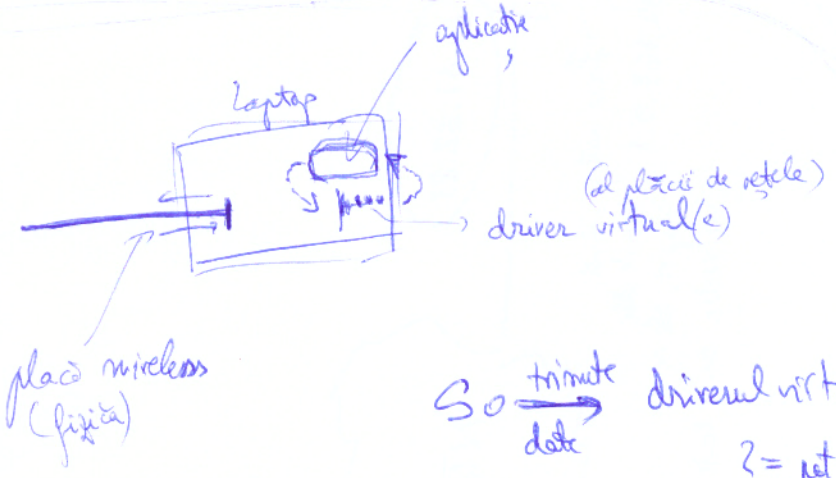
(2 interfețe ridicate pe aceeași placă de rețea)

172.30.0.0/16
193.226.40.128/27

↳ pot face asta, diferențiind drept port, ce rol să îndeplinească dispozitivul (care ar putea fi și server DHCP, și altceva).



* la nivelul III se verifică dacă o adresă face parte dintre cele ale plăcii de rețea
→ da ⇒ urcă în nivel
→ nu ⇒ se comportă ca router și trimite mai departe

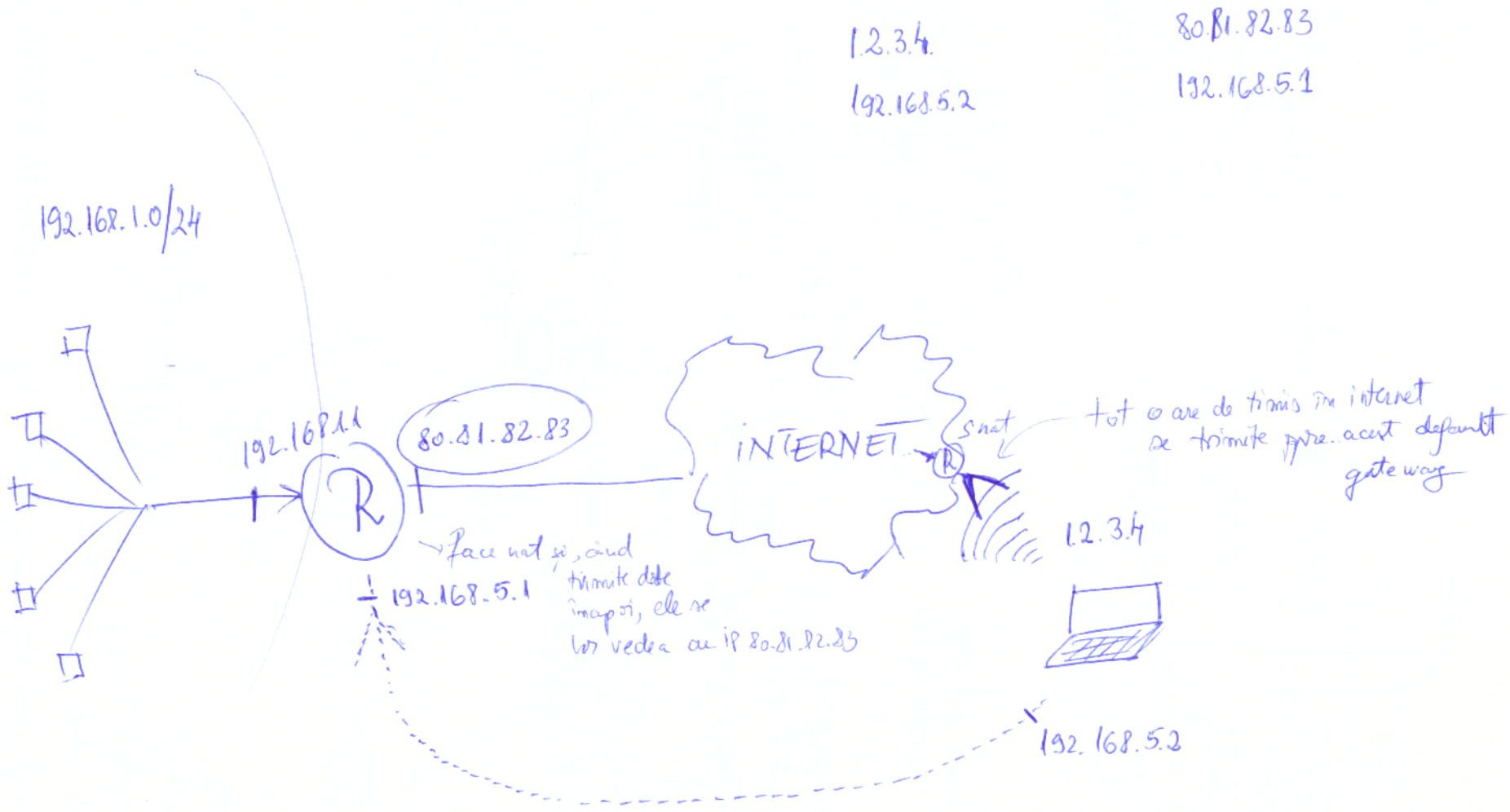


S0 $\xrightarrow[\text{date}]{\text{trimite}}$ driverul virtual

?= pot să trimit datele ✓

↓ cui?

nu în exterior, dacă pot spre o → aplicație (la nivel proces)



Client de VPN

Dst	Gwin
192.168.1.0/24	192.168.5.1
0.0.0.0/0.0.0.0	192.168.5.1 → ! modificarea default gateway-ului
80.81.82.83/32	default gateway (vechi fizic)