

Cursul 1

Andrew Tanenbaum - "Rețele de calculatoare"

Socket-uri

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

Struc TCP/IPPrincipii fundamentale ale funcționării rețelelor de calculatoare:

- Dirijarea (routarea) pachetelor → routere
 surse → pachete → ne neapărat unică (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.
- Strucura de protocoale TCP/IP → RFC (Request for Comments)
 - protocol = set de reguli de comunicare

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

↓ (static) ↓ (dynamic)
 în sistemul de fizicii program în execuție (rulează pe RAM)

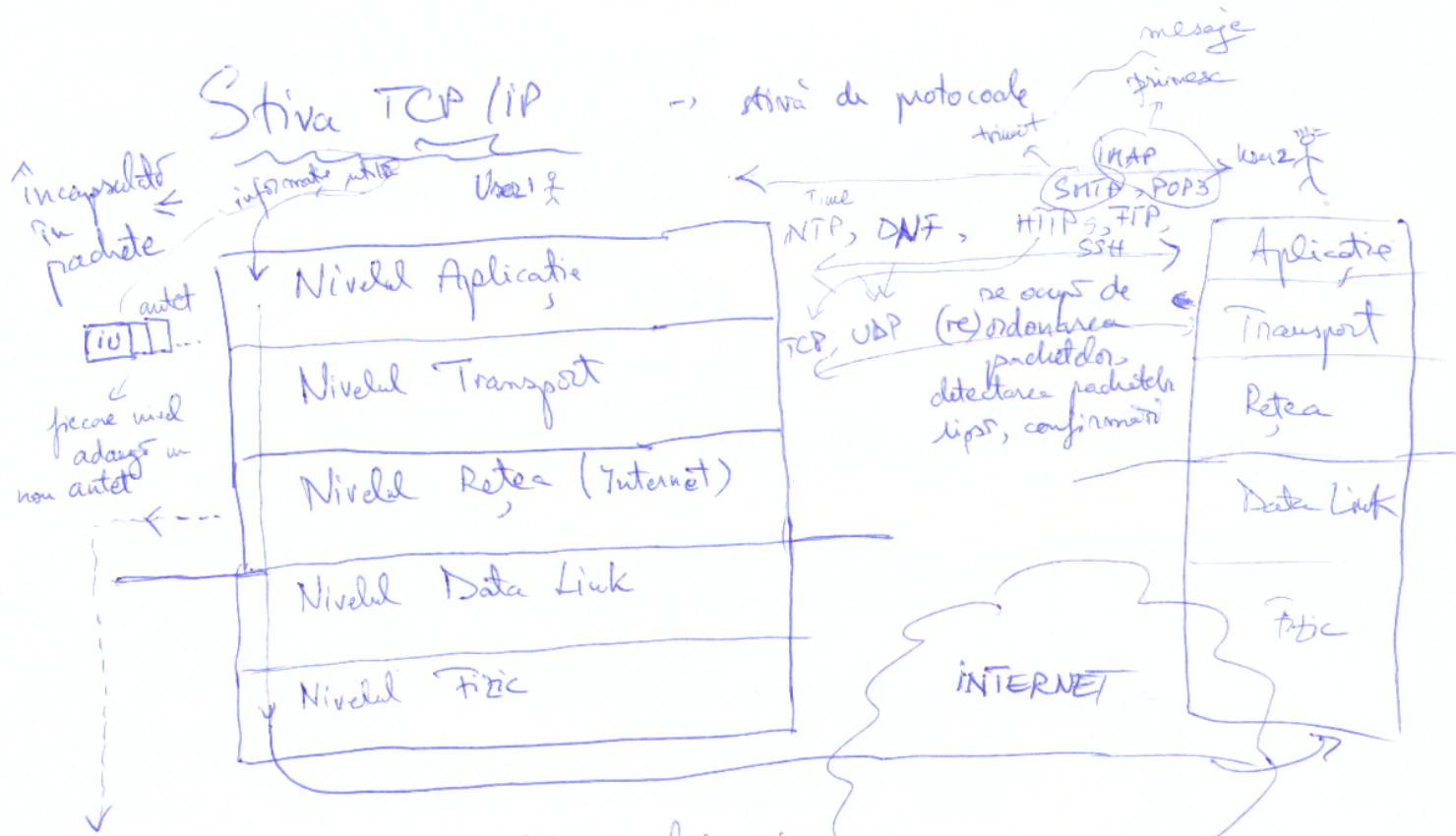
- protocoale sunt descrise în RFC-uri

IS - MI

Apache - Linux,

protocoale
 \descrise
 proprietary

HTTP Hypertext Transfer
HTTPS Protocol



acest nivel se ocupă de stabilita ca într-un traseu optim pentru linierea pachetelor de informație (se poate ca un pachet trimis mai târziu să ajung mai repede)
⇒ ordine ne-liniară

⇒ mai lent, mai sigur
F, protocol reliable

• Protocoale de la nivelul Transport

- TCP (face task verificabile
⇒ ordine, completitate)

- UDP

⇒ mai rapid, se pot pierde pachete

Aplicatie

- NTP, DNS

- HTTP, HTTPS, FTP, SFTP

SMTP, IMAP, POP3

routerul:

- porturi fixe (din switch) de la nivelul Transport

Default

HTTP → port 80

/etc/services

HTTPS → port 443

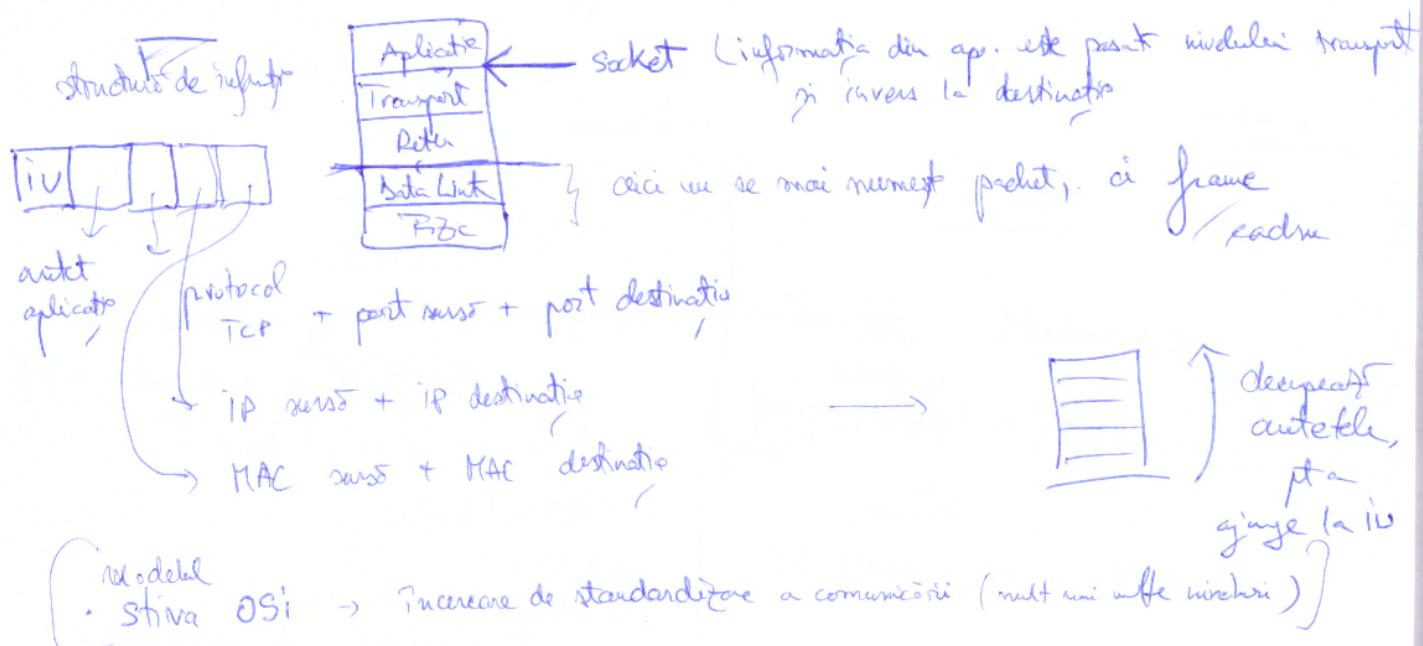
FTP → port 21

SSH → port 22

→ Nivelul Rețea → adresa IP (Internet Protocol)

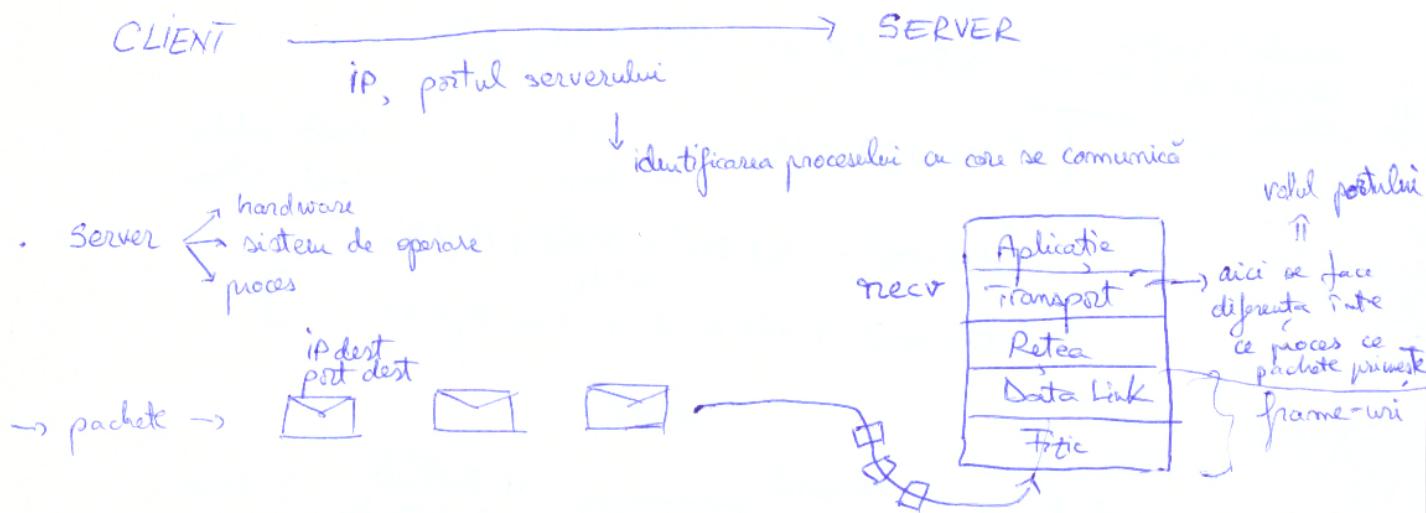
→ Nivelul Data Link → switch, adresă MAC
/ ethernet
/ hardware

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



- VPN → se introduc multe pachete IP peste o altă stivă TCP/IP

Cursul 2

CLIENT

c = socket(,)

↳ e valoare de指向可变的:

Transport - TCP

Retea - IPv4

- tip structura*

if (connect(c, &i, sizeof(i)) < 0)

{- port server}

sub forma unei structuri

* tip structura : AF_INET

SERVER

s = socket(,)

→ pagina urmatoare

↓

eroruri

Send(c, &i, sizeof(i), 0)

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

close(c)

SERVER

$s = \text{socket}(_, _, _)$ $\rightarrow 0.0.0.0$

$\text{if}(\ \text{bind}(s, \{ -\text{inaddr_ANY} \}), \text{sizeof}(\text{structura}) < 0)$

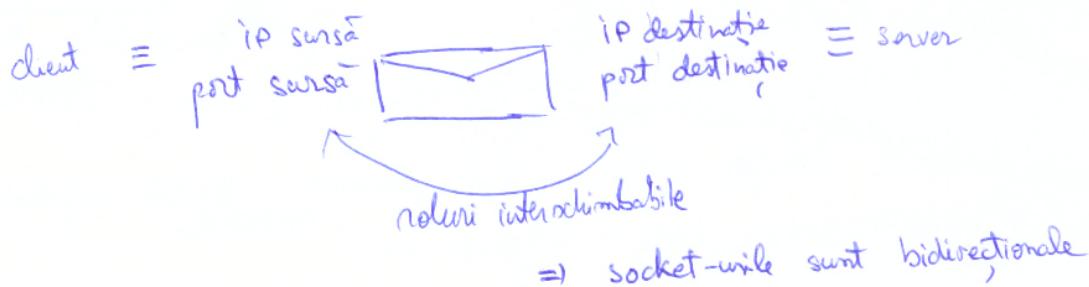
aceasta ii conferă
atributul de
server $\leftarrow \text{listen}(s, 5)$

$[\rightarrow 192.168.1.17]$ $\begin{matrix} \text{inet_addr} \\ \text{inet_ntoa} \end{matrix}$

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

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

pacete



\rightarrow pentru a verifica ce porturi sunt ocupate: netstat

bind \rightarrow la client este optional (în general, nu se face), pentru că portul clientului nu este important și, de asemenea, dacă se face bind la client există riscul de a ajunge pe un port deja ocupat

while (Δ) {

Generarea clientului respectiv	$c' = \text{accept}(s, \{ -\text{port client} \}, \text{"sizeof"})$	un server iterativ
	$\text{recv}(c')$	
	$\text{send}(c')$	
	$\text{close}(c')$	

$\}_{\text{close}(s)}$

! Obs: ordinea pt. send și recv,
semnificație + alte detalii / reguli de
comunicare \rightarrow RFC

→ transformarea serverului iterativ în server concurrent:

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

înainte de trimis

htons
htonl

ntohs
ntohl

după primire

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

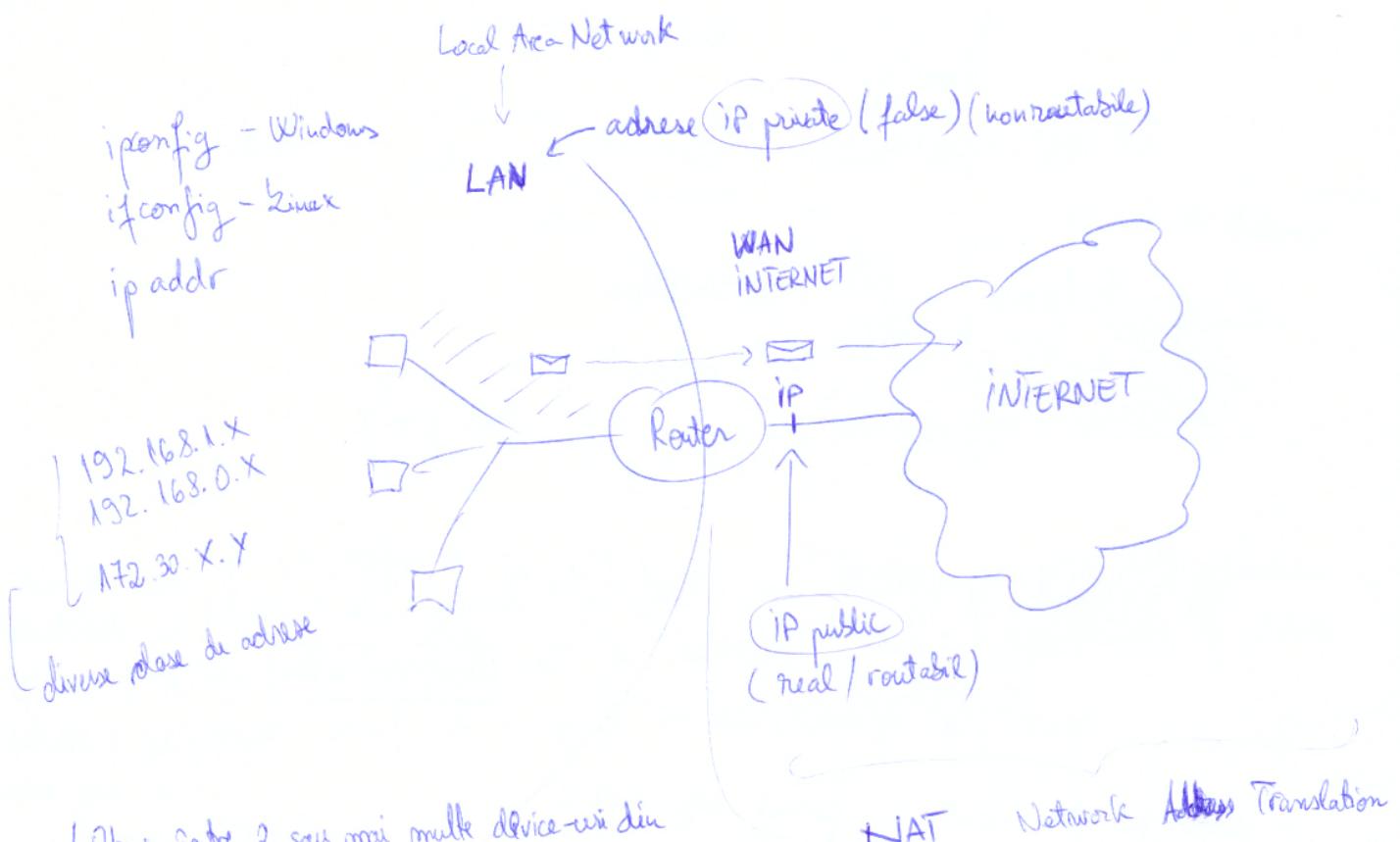
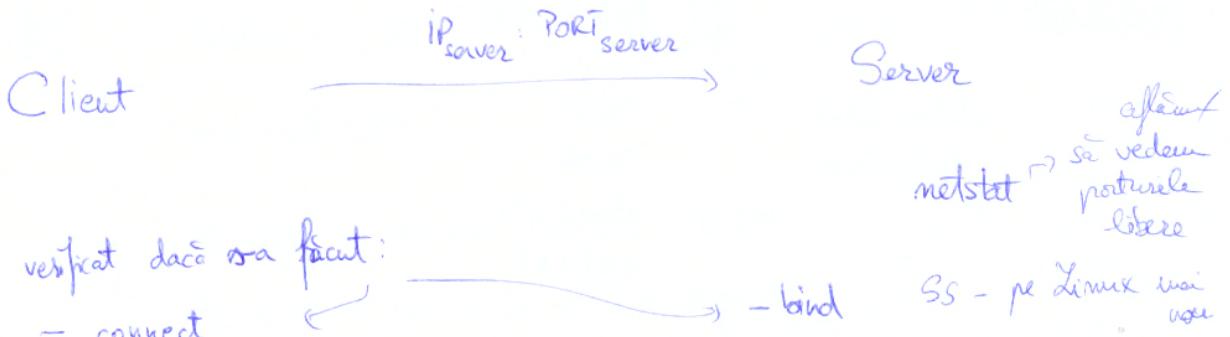
recv la server

E8|03|00|00

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

- datele, în rețea, circulă în format big endian

Cursul 3



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

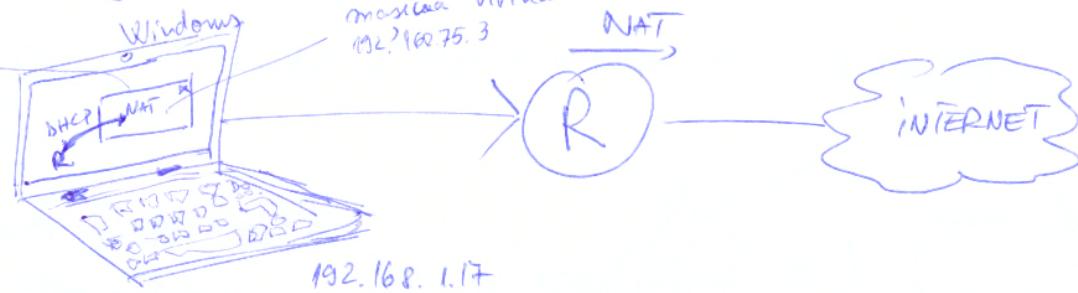
→ dacă merge ping între doi IP-uri, va trebui conexiunea

IP C: 192.168.1.14

IP S: 192.168.1.37

→ Nu e nevoie să fie în același rețea, doar observăm că fac parte din același clasa de adrese IP.

Modif. NAT cu model Bridge

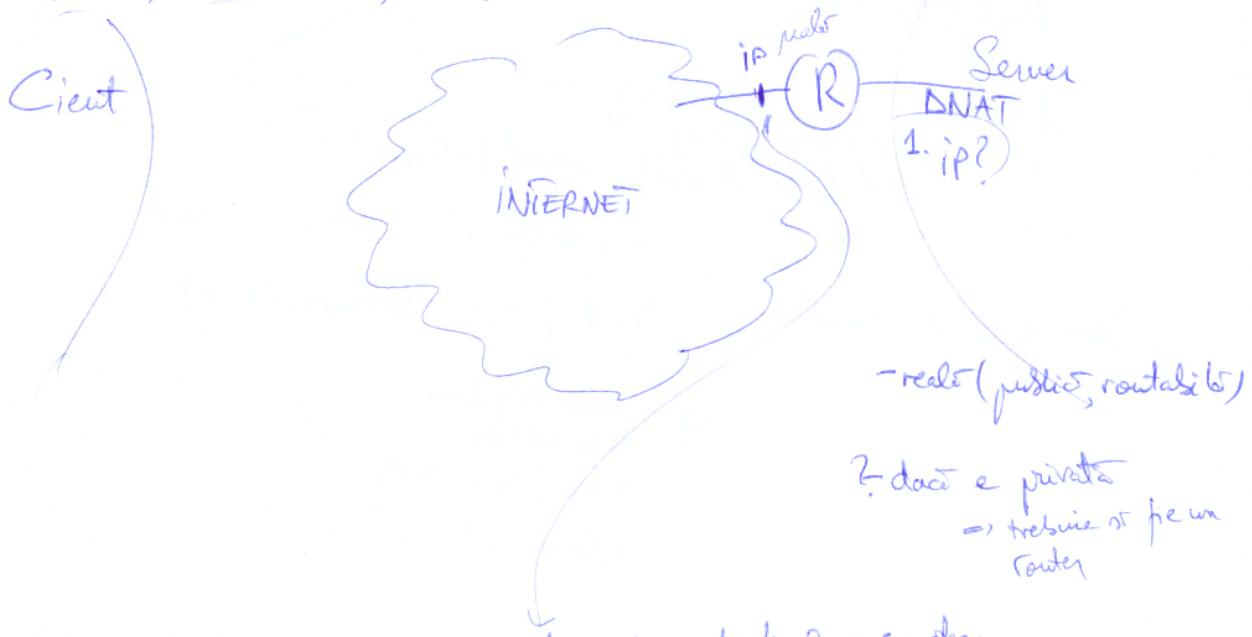


- Mașina fizică „joacă rolul de Router” pentru mașina virtuală
- Dacă avem model Bridge pt. mașina virtuală (nu NAT, ca mai sus), mașina fizică nu mai e router pt. aceasta, căci Poziția Routerului (R) îi va furniza adresa IP. (forma 192.168.1.X)
- Nu este obligatoriu ca serverul DHCP să ruleze pe router.

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

→ client și server în rețele diferite!

192.168.1.17

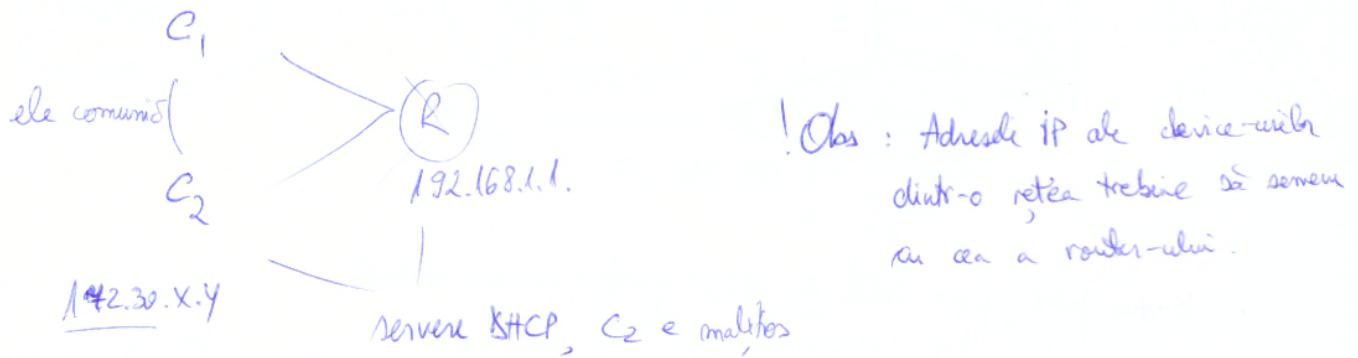


dacă cineva doar să se conecteze la adresa IP a serverului și să redirecționeze la adresa privată a serverului trebuie să specifică adresa IP reală a sistemului (extern) → clientul

DNAT

Port forwarding
Virtual Server

7.30.4.5 → nu va avea acces la internet!



C → S

↓

uint16_t m;

n = htons(m)

send (c, &n, ...)

for (i=0; i < ntohs(m); 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)

m = strlen(x) → htons → send

Bacă e să de cădere: } Send (c, x, strlen(x)+1, 0)

sf recv (c, &m, ...)

m = ntohs

recv (c, x, n+1,

(x[n]=0)

! Important: cîte date trimitem, atâtceasă + 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 citește 2 octeti
    (hu - short unsigned)
```

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

! → pune la adresa lui i 4 octeti,
înse i are doar 2 octeti
⇒ se scrie 4 octetii care pot
affecta alte date / variabile

strace / ltrace

strace ./client

Cursul 4

Socket-uri UDP

TCP < protocol orientat pe conexiune
reliable

Emitător
(client)

Receptor
(server) (roluri "mai"
egale; interchimbabile)

c = socket (AF_INET, SOCK_DGRAM)

s = socket (AF_INET, SOCK_DGRAM)

* UDP - nu se poate face deosebire între cauzile când
 { - se pierde date de la E la R
 - nu există un proces cu care să
 se comunică
 - se pierde rezultatul/răspunsul

* 64 Kb - dimensiunea maximă a unui packet

(MTU - Maximum Transfer Unit)

Situatii UDP (nereverificate, ca în cazul TCP)

↓
 este mai importantă rapiditate
 (modelul interogare-răspuns)

pierdere de date
 jumătate, neordonate
 (out of order)

~~connect~~

Sendto (c, de unde
 din memorie, cat, flg,

structure, sizeof(structura))
 - ip dest
 - port dest

bind (s,

~~accept~~

~~listen~~

recvfrom (s, unde, cat,

flg, structură, &flg(str))
 - IP emittor
 - port emittor

J

recvfrom()

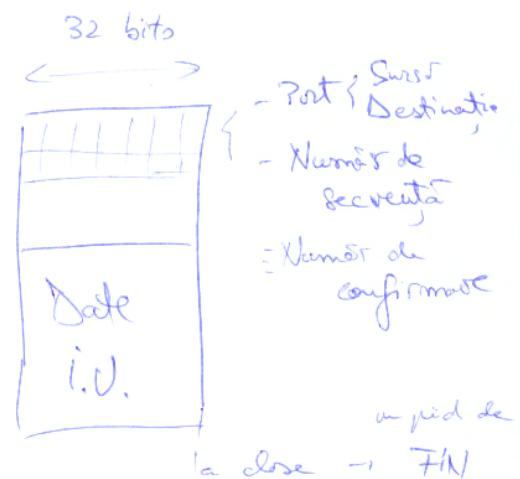
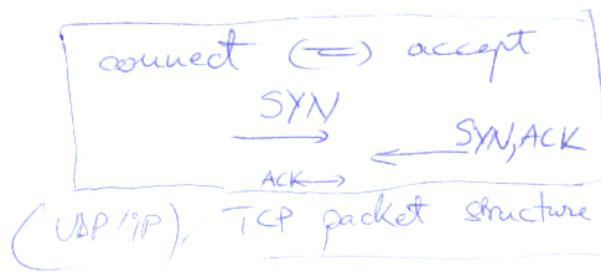
close(c)

Sendto(

* sendto trimite datele care
pertinente ale cărui date sunt
de identificare
completate de recvfrom

(Dacă fac severul concurent, după recvfrom generez un port
random (pt. identificare), fac un fiu, iar "conunicare" se face
pe baza portului noi generat.)

TCP:



→ pachete de control \rightarrow fizice date
(ex: se trimite și a confirmă în final
aceea care conexiunea și a primulă
date din sens invers)

→ confirmările nu se fac instant (ar supraclona final de conexiune / teama
cu pachete goale / de control)
↓
se confirmă copleșirea de date

ex:

C = socket (AF_INET, SOCK_STREAM)

connect(c) -

close(c)

C = socket (AF_INET, SOCK_DGRAM)

{ "recidarea" lui c \rightarrow c nu e în același timp
socket TCP și UDP

Sendto(x, 6000)

recvfrom(c

Sendto(c, 6001)

6001

bind(s, 6000)

recvfrom(s, 6000)

if (j & k) == 0 {

bind(s', 6001)

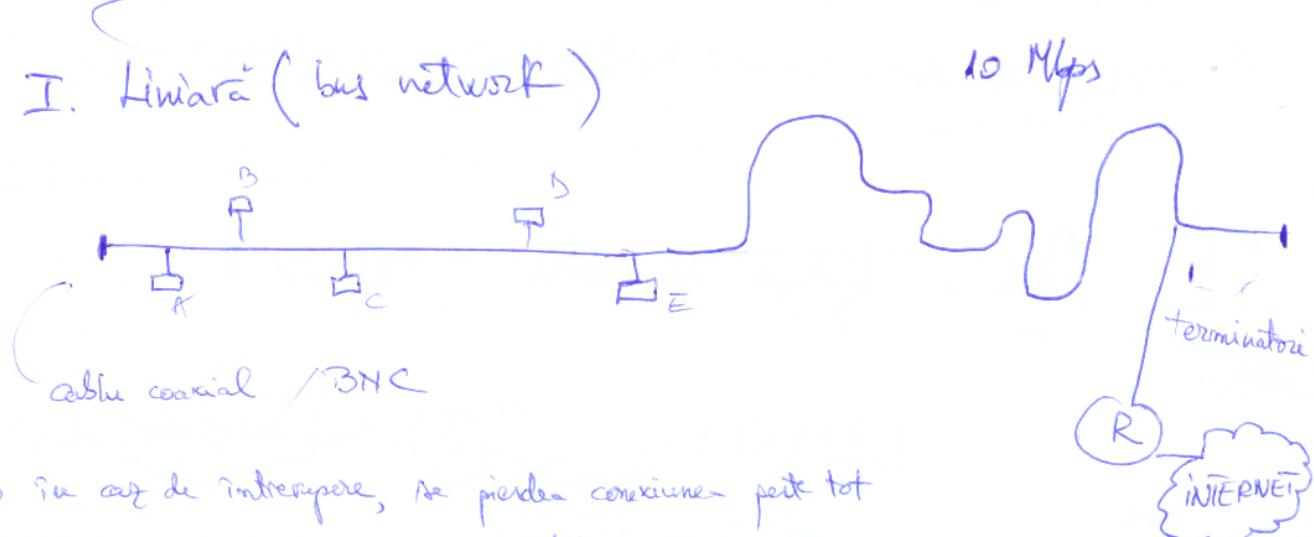
recv

Sendto(s, 6001)

Curgul 5

Topologie de rețele locale

I. Liniară (bus network)



→ în caz de întârziere, se pierde conexiunea pe tot

→ problema de coliziuni

local

A $\xleftarrow{5 \text{ Mbps}}$ B

C $\xleftarrow{5 \text{ Mbps}}$ D

→ rețele cu difuzare \rightarrow Broadcast Domain

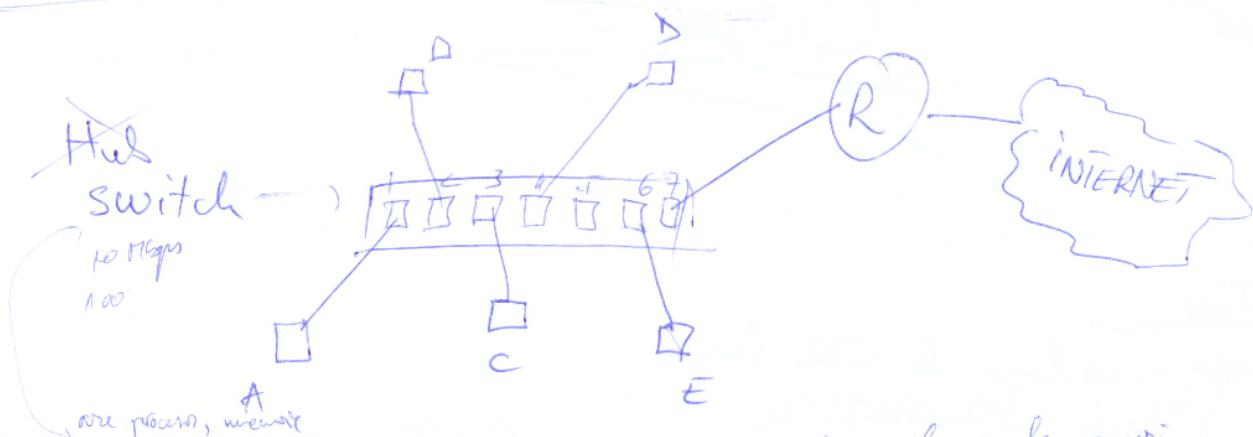
1 kibit = 1000 biti

1 megabit = 1000 kilobiti

1 gigabit = 1000 megabiti

1 bilobyte = 1024 bytes

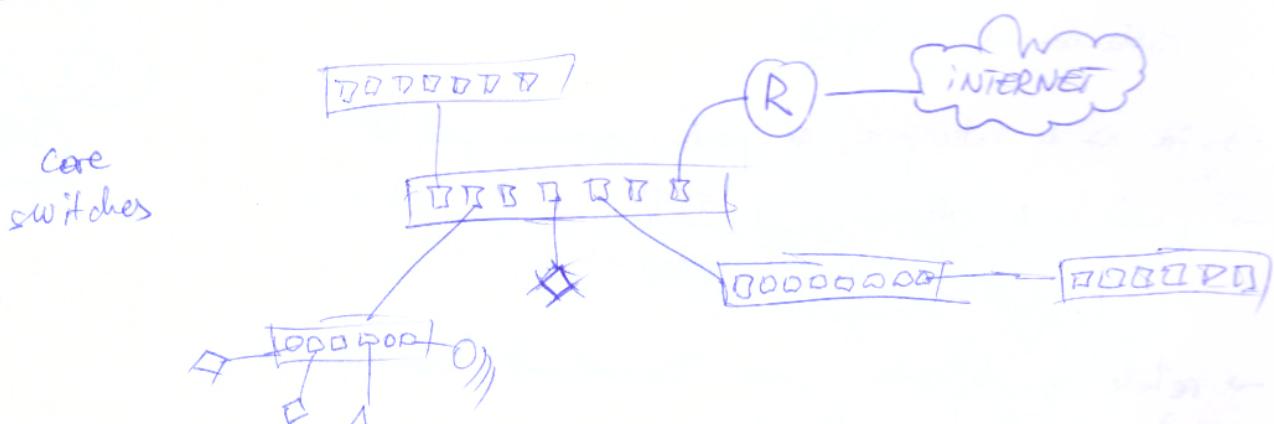
II. Rețele cu topologie stea



Definiează un fel de tabătă, care să fie cum comunică deținătorii, a frame-urii

→ reduce coliziunile (în cazul hub-ului nu se mai era problema)

III. Rețele cu topologie stea extinsă (extended star)



viziabile la nivel data-link

Adresa MAC

- fizic
- hardware
- ethernet

6 octeti

AB : FC : 2D : 9E : 14 : 7D

↳ identifică modelul

↳ produsul



→ între LAN-uri diferențe nu pot să văd MAC-urile

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

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

RARP - Reverse ————— , DHCP

↳ permite aflarea IP-ului pe baza de MAC

Patru seturi de fact - IP
- Netmask

- DNS

- Default Gateway - adresa IP a roțitorului

are min. 2 adr. IP

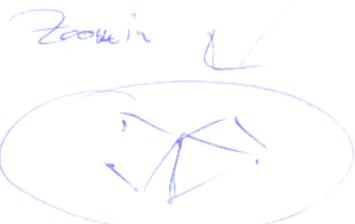
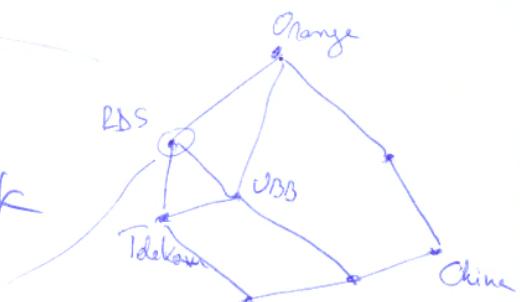
IV Retele cu topologie de graf incoplit

(caz e organizația rețea de internet)

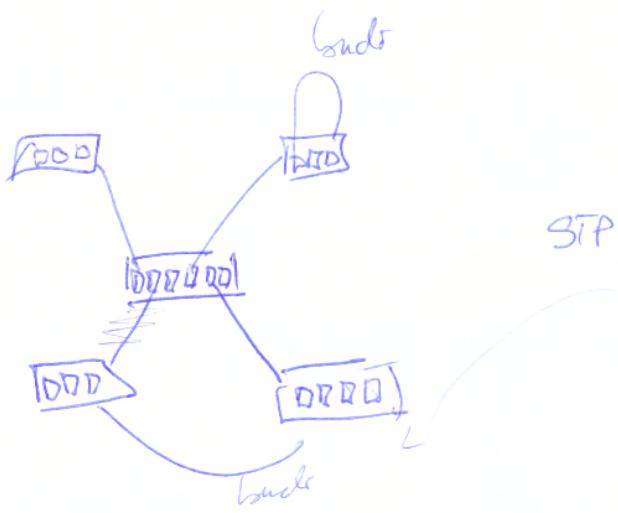
LAN - Local

MAN - Metropolitan } Area Network

WAN - Wide



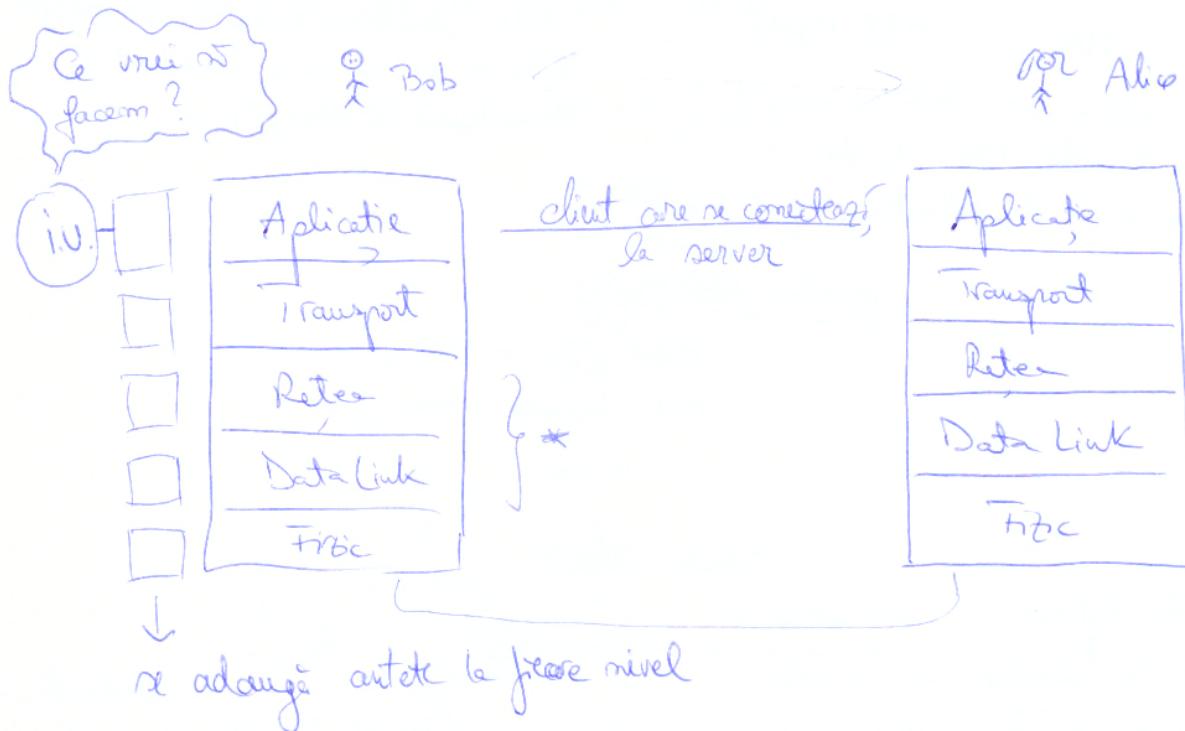
Retele de rețele de proprietăți



SIP - Standing Tree Protocol
 - reduces polling load
 (core port for service)

Switch wiring management - report × port
 Link management

Cursul 6



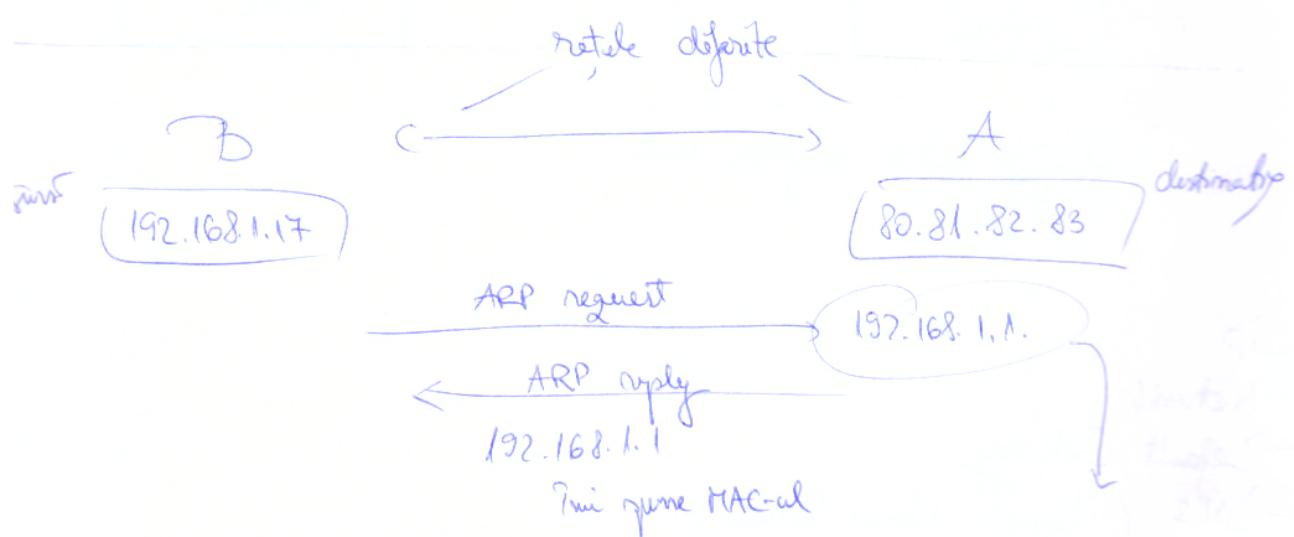
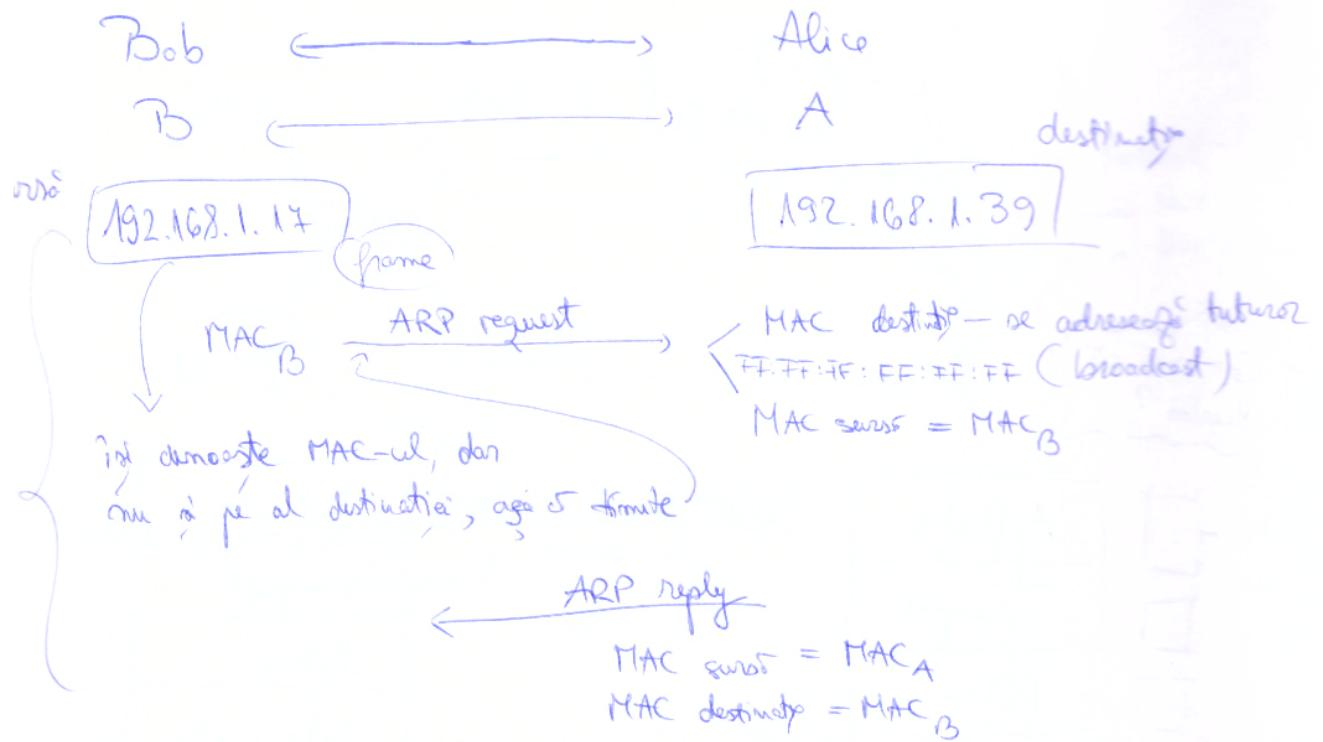
- IP
- Network
- Default Gateway
- DNS (Domain Name Server)

Adăugări

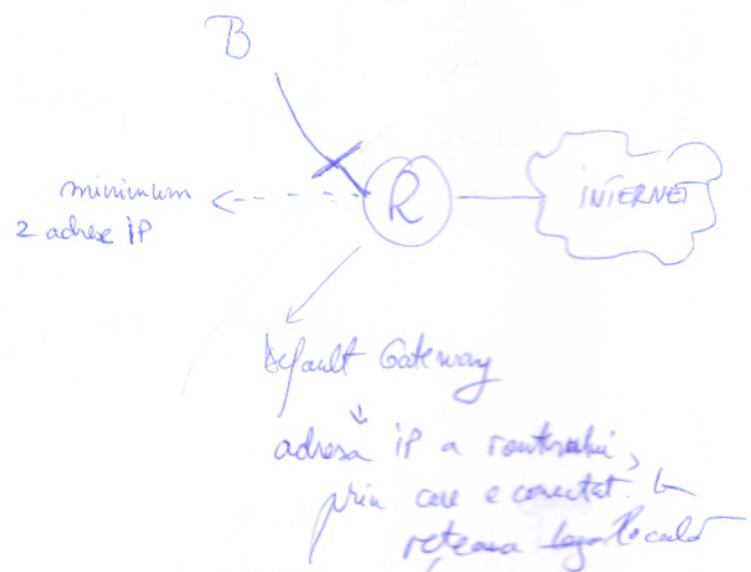
- Nivel transport: protocol (TCP / UDP) + port $\begin{cases} \text{server} \\ \text{destinație} \end{cases}$
- data: IP $\begin{cases} \text{server} \\ \text{destinație} \end{cases}$
- data link: MAC $\begin{cases} \text{server} \\ \text{destinație} \end{cases}$

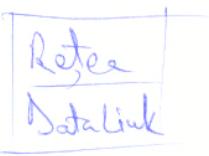
arp -m (Linux)
- a (Windows)

tcpdump (crează ARP request-uri, și în reply-uri)



Când se doară transmisie de informații în afara rețelei locale, se doară să trimitem router-ului (Default Gateway). Pentru acest lucru, se lansează o apelare (prin ARP request) către MAC-ul routerului.





$$IP_{dest} = 80.81.82.83$$

$$MAC_{dest} = MAC(pt\ IP\ 192.168.1.1)$$

RARP → determină IP pe baza MAC

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

- IP
- Network
- Default Gateway
- DNS



→ urcă pînă la nivelul reței și observă că adresa IP a destinației nu corespunde cu niciunul dintre IP-urile routerei



↓
Se fac ARP request-uri repetate, din router în router, pînă 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 același router

↪ trebuie să am adrese IP din același class ca cele de pe placă de rețea interconectată la routerului

ipconfig /all → ne dizează adresa serverului DHCP

16.11.2022

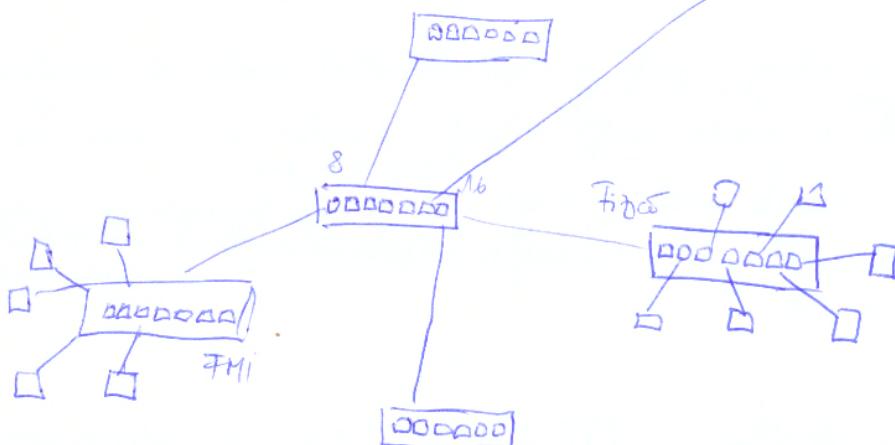
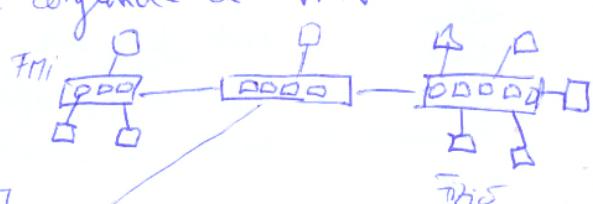
Cursul 7

Virtual

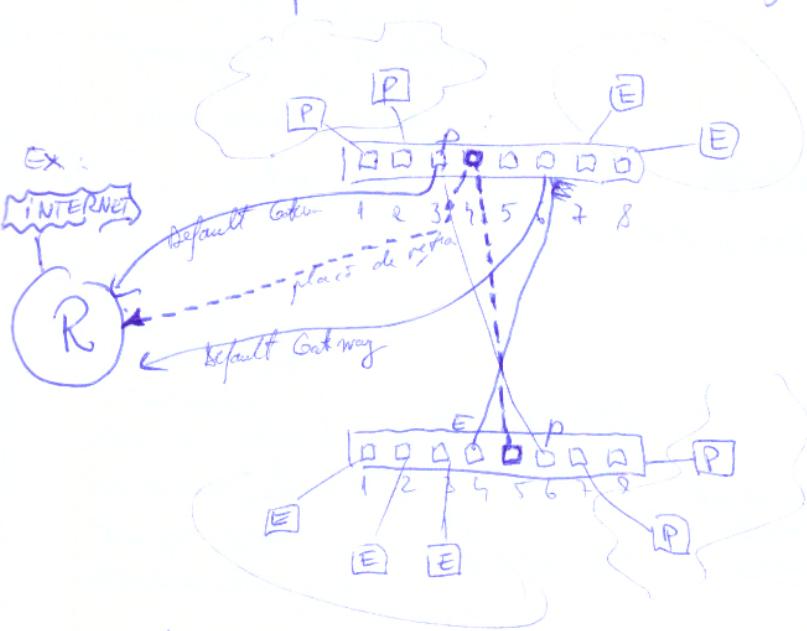
VLAN - wii

→ a nu se confunda cu VPN

VLAN id



→ fiecare port din switch-uri îi este asignat un VLAN id (se reamorcă mai ușor prin numere)



! → model mai sensibil din punctul de vedere al securității

(veronica elementara / rudimentară,
unde sunt necesare 2
cabluri între cele 2
switchuri, fiind
2 categorii de
acevi E, P)

Soluție: ---
{ paturile lyate astfel
pot apartine mai multor
port în mod liniar VLAN-uri

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

Clase de adrese

192.168.1.X |
 0
 :
 255

adresa
rețelei

b) toate device-urile conectate la același răbra cu un comun primul
trai octet, diferențiere fiind făcându-ne prin ultimul octet

în campus 192.30.X.y |
 :
 192.30.0.0

adresa
rețelei

179.30.255.255

identifică adresele calculate din cadrul rețelei

Subnet mask

255.255.255.0

255.255.0.0

192.20

stind căi octetii sunt
255, atâtia reprezentă
adresa rețelei, respectiv
octetii de o desemnată
adresă device-ului
din rețea

Ex: 10.1.2.3
10.17.253.242

→ netmask

255.0.0.0

192.168.1.14

255.255.255.0

192.168.1.0

se face schimbare de biti
pe netmask

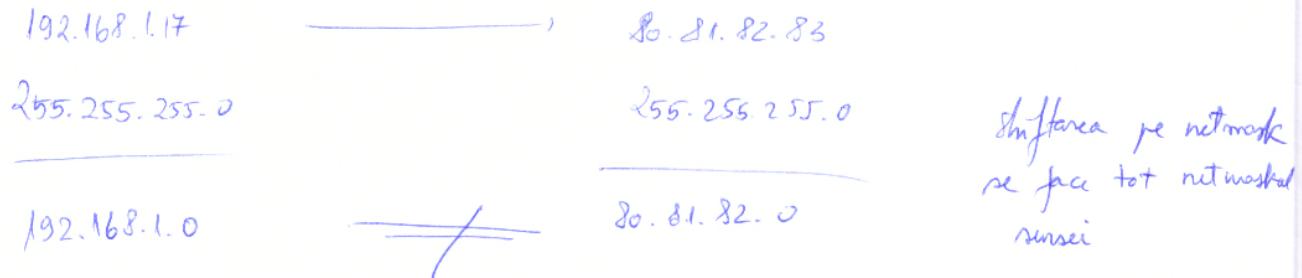
192.168.1.135

255.255.255.0

192.168.1.0

• se poate accesa
adresă de
rețea, se poate
face direct un
ARP request
către destinatar

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



↳ fiind difuză, se face ARP request către router

! Obs: \diamond 192.168.1.0 \rightarrow nu se folosește, fiindcă este adresa rețelei

1.1 \rightarrow prima adresă care poate fi folosită

1.255 \rightarrow nu se folosește, deoarece este adresa de broadcast

\diamond indiferent de adresa IP dintr-o clasă de adrese, shiftarea de biti între adresa IP și netmask ducă la aflarea adresei de rețea

\diamond adresa de broadcast (ultima din clasa) se obține făcând sau pe biti 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 \rightarrow exprimarea vîjurant 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

→ adresa de rețea, prima
 $192.168.1.0 / 255.255.255.0 \rightarrow 2^5$ adrese IP
 /24 → alternativă de notare pt. netmask
 $192.168.1.255 \rightarrow$ adresa de broadcast, ultima

Clasă

C

$172.30.0.0 / 255.255.0.0 \rightarrow 2^{16}$ adrese IP
 $172.30.255.255 / 16$

B

$10.0.0.0 / 255.0.0.0 \rightarrow 2^{24}$ adrese IP
 $10.255.255.255 / 8$

A

$\begin{cases} 0.0.0.0 \\ 255.255.255.255 \end{cases} \quad 2^{32}$ 0.0.0.0 - INET ADDR ANY
 127.0.0.1

↳ spațiu împărțit în
 clase de adrese A, B, C
 $223.255.255.255 \rightarrow$ ultima adresa IP
 validă pentru utilizare
 > 224

$\begin{cases} 0000\ 0000. \\ 0111\ 1111. \end{cases} \Rightarrow \begin{cases} 0. \\ 127. \end{cases} =,$ clase de ip + (primul bit din primul octet=0)

Retea 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
	:		
	$126.0.0.0 / 8$		0.255.255.255

→ nu se folosesc

$\left\{ \begin{array}{l} 10000000 \\ 10111111 \end{array} \right\} = \left\{ \begin{array}{l} 128. \\ 191. \end{array} \right\} \Rightarrow \underline{\text{clase de tip B}} \quad (\text{primii 2 biti din primul octet sunt } 10)$

128. 14. 0. 0 / 255. 255. 0. 0

→ poate fi și

128. 14. 39. 0 / 255. 255. 255. 0

140. 255. 0. 0 / 16

172. 30. 0. 0 / 16

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

$\left\{ \begin{array}{l} 11000000 \\ 11011111 \end{array} \right\} = \left\{ \begin{array}{l} 192. \\ 223. \end{array} \right\} \Rightarrow \underline{\text{clase de tip C}} \quad (\text{primii 3 octeti din primul octet } 110)$

192. ...

223. 255. 255. 255

224. ...

→ clase experimentale : D → 1110

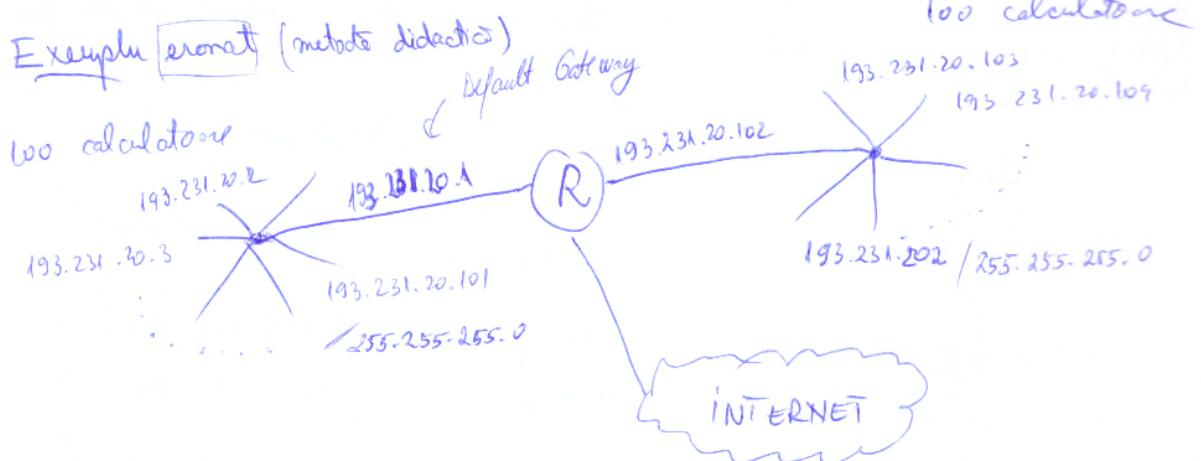
E → 11110

...

193. 231. 20. 0 / 24

255. 255. 255. 0

Exemplu eronat (metoda didactică)



S.

193.231.20.2 l —>

255.255.255.0

193.231.20.0

d.

193.231.20.110 l

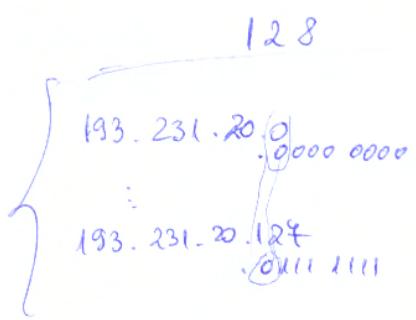
255.255.255.0

193.231.20.0

—>

sursa crede că destinație
se află în același rețea ca
acesta

Soluție: imparț. clasa în 2:

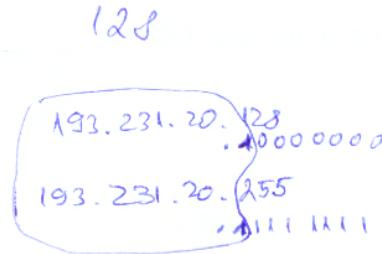


Netmask:

255.255.255.1000 0000

/25

255.255.255.128



Netmask:

255.255.255.128

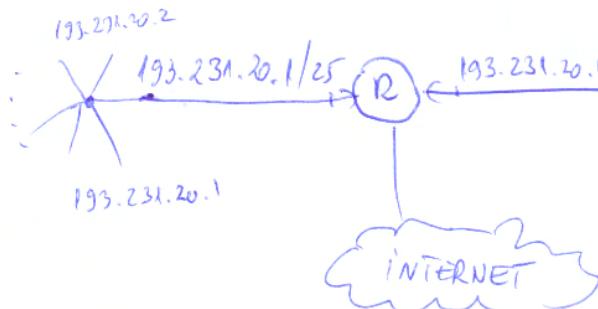
→ adresele de căci au în comun
primii 25 de biti, diferențiindu-ne
pe ultimii 7 biti

193.231.20.0

193.231.20.0 /255.255.255.128

193.231.20.128 /25

loc. calculatoare

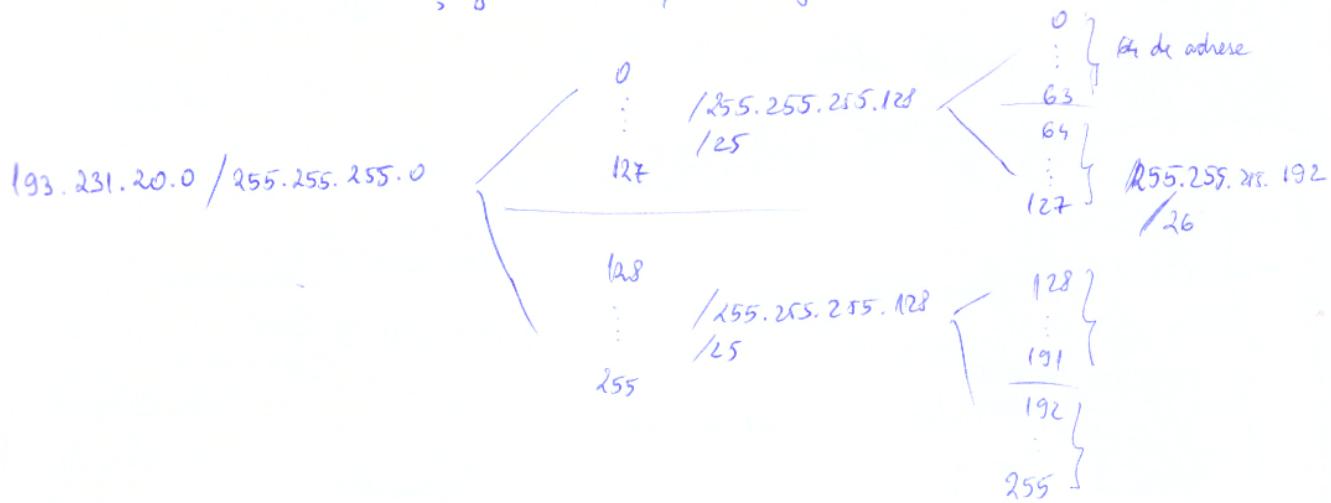


loc. calculatoare

193.231.20.130
⋮
193.231.20.229

$193.231.20.2$ & $193.231.20.130$ &
 $255.255.255.128$
 \hline
 $193.231.20.0$ \neq $193.231.20.128$

\Rightarrow sursa înțelege că destinație se află în altă rețea



\rightarrow fiecare dintre cele 4 subdrese a către fișă de adrese are adresa de rețea și adresă de broadcast



! Adresa de rețea este multiplu de dimensiunea clasei!

- putem imparti clasa în 3? \rightarrow Da, dar nu de dimensiuni egale (ex: impart în 2 jumătăți, înc pe prima o jumătate și împart din nou pe 2)
- pentru clase C sau mai mici putem deduce dimensiunea unei (sub)clase răzând din 256 ultimul octet din netmask

B

25 calculatoare



A

10 calculatoare

(16)



C

50 calculatoare

(65)

Să dă clasa 193.231.20.0 /24

A 193.231.20.0 → Rețea /255.255.255.240
 20.15 → broadcast /28

X B 193.231.20.16 /255.255.255.224
 20.67

De ce e greșit?

=> 193.231.20.14 & →
255.255.255.224
 . 00010001
 . 11100000

 193.231.20.0 ≠

193.231.20.33 &
 255.255.255.224

 . 00100001
 . 11100000

 193.231.20.32

P 193.231.20.32 /27
 20.63

C 193.231.20.64 /26
 20.124

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

...

→ cea mai mică subclasă pe care o poate 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

...

→ adresă de rețea

{ 193.231.20.0

193.231.20.1

193.231.20.2

193.231.20.3

→ adresă de broadcast

Netmask ⇒ căți biți au în comun toate adresele de la același rețea

(partea comună din adresa IP = adresa rețelei)

la fel	
0	0000 0000
0	0000 0001
0	0000 0010
0	0000 0011

⇒ 2 biți diferenți

↓
netmask / 30

↪ sumă 30 de biți la fel = netmask 255.255.255. 1111 1100
252

Exemplu pb examen



5 calculatoare

! 193.231.20.15 (trebuie să fie adresa unui calculator)

⇒ dacă am alege să divizăm în cîte 8 ⇒ gresit

.15 → broadcast în clase de 16

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

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

$$\text{ex: } 256 - 256 \Rightarrow 0 \rightarrow 255.255.255.0$$

$$256 - 128 = 128 \rightarrow 255.255.255.128$$

$$256 - 64 = 192 \Rightarrow 255.255.255.192$$

:

clase de 256

$$193.231.20.0 / 255.255.255.0$$

/24

clase de 256

$$193.231.21.0 / 24$$

clase de 512

agregare

clase agregata

$$193.231.20.0 / 23$$



cum il afili?

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

< 193.231.20.0 ← rețea

< 193.231.21.255 ← broadcast

$$\left. \begin{array}{l} 193.231.20.0 / 24 \\ 193.231.21.0 / 24 \\ 193.231.22.0 / 24 \\ 193.231.23.0 / 24 \end{array} \right\} \rightarrow 2^{10} \Rightarrow 32-10 = \underline{22} \quad \text{adresa de retea} \\ 1024 \text{ adresa} \quad \Rightarrow 193.231.20.0 / 22 \\ | 255.255.252.0$$

$\rightarrow \left. \begin{array}{l} 193.231.21.0 / 24 \\ 193.231.22.0 / 24 \end{array} \right\} \rightarrow \text{nu se poate face agugne } 193.231.21.0 / 23 \text{ e putin},$
 pentru că nu e multiplu de 512

! O clasa de reteze trebuie să împarte la multiplul de n

$$\begin{array}{r} 193.231.21.0000 0000 \\ \hline 10000 0000 \\ \downarrow \\ 256 \end{array}$$

$$\begin{array}{r} 193.231.21.0000 0000 \\ \hline 10 0000 0000 \\ \downarrow \\ 512 \end{array}$$

nu se
împarte

$$\begin{array}{r} X.y.z.0 \\ \hline 2^8 \end{array} \rightarrow \text{clasa de } 256 \text{ se împarte la } \underline{2^8}$$

$$\begin{array}{r} 2 \text{ bit de } 0 \\ 193.231.20.(\underline{0}) \quad 8 \text{ biti de } 0 \\ \hline 1024 \end{array} \rightarrow \text{se împarte la } 2^{10}$$

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

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

cl. clase de 3L = $2^5 \rightarrow 32-5 = 27 \rightarrow$ netmask 255.255.255.223

$$\begin{array}{r} 193.231.20.17 \\ 255.255.255.223 \\ \hline 193.231.20.0 \end{array}$$

$$\begin{array}{r} 193.231.20.33 \\ 255.255.255.223 \\ \hline 193.231.20.32 \end{array}$$

nu aparțin aceluiși clase de 32

$\left\{ \begin{array}{l} 193.231.21.0 ? \text{ o putem seta pe un calculator!} \end{array} \right.$

\hookrightarrow nu se stie daca e adresă de rețea prima când nu se cunoaște și netmask-ul

\hookrightarrow ar putea fi parte dintr-o clasa mare și ar trebui să se ușureze pe 193.231.21.255 la mijloc

\hookrightarrow accesă poveste ce să mai săn

20.0.0.0 /8
/ 255.0.0.0

împărțim în sub clase de 2^{16}

$\Rightarrow 20.0.0.0 \dots 20.0.255.255 \rightarrow 20.0.0.0/16$
 $/255.255.0.0$

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 \rightarrow împărțim în 2 sub clase

20.3. x. y

\swarrow 20.3.0.0 20.3.127.255 $\Leftarrow 2^{15}$ /17
 \searrow 20.3.128.0 20.3.255.255 $\Leftarrow 2^{15}$ /17

20.3.0.0 /17 și 20.3.128.0 /17

4 au în comun primii 17 biti

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

/17

Exercițiu:

[20.21.22.48 ... 192.168.17.95]

↳ nr. de adrese de rețea

↳ nr. de adrese de broadcast

- și se scrie intervalul ca reunirea de clase de adrese (maximale)

$$\left\{ \begin{array}{l} 0.0.0.0 \rightarrow \text{cea mai mică adresă} \\ 255.255.255.255 \rightarrow \text{cea mai mare} \end{array} \right. \Rightarrow \begin{array}{c} 0.0.0.0 / 0.0.0.0 \\ / 0 \end{array}$$

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

}

$$\rightarrow 20.21.22.48 / 255.255.255.240 \quad (256-16) \\ \downarrow \quad / 28 \quad (32-4, 2^5=16) \\ \text{cea mai mare clasa care poate înapele de la 16} \quad (48:16)$$

$$\rightarrow 20.21.22.64 / 255.255.255.192 \\ / 26 \\ \hookrightarrow \text{poate înapele clase de 64}$$

$$\rightarrow 20.21.22.128 / 255.255.255.128 \\ / 25 \\ \hookrightarrow \text{poate înapele clase de 128}$$

... prima în 20.21.22.255

$\rightarrow 20.21.23.0 / 24$

\hookrightarrow poate fi capăt o clasă de 256 și este mai mare (23 împărt).

$\rightarrow 20.21.24.0 / 21$

/ 255.255.248.0

\hookrightarrow clasă de maxim. 2^{11} , $2^4 = 0001 \underline{1000}$
 $3+8=11$

$$8 \left\{ \begin{array}{l} 20.21.24.\times \\ 20.21.25.\times \\ 20.21.26.\times \\ \dots \\ 20.21.31.\times \end{array} \right.$$

$\rightarrow 8 \cdot 256 = 2^3 \cdot 2^8 = 2^{11}$

$\rightarrow 20.21.32.0 / 19$

\hookrightarrow clasă de maxim. 2^{13}

$$32 \left\{ \begin{array}{l} 20.21.32.\times \\ 20.21.33.\times \\ \dots \\ 20.21.63.\times \end{array} \right.$$

$\rightarrow 32 \cdot 256 = 2^5 \cdot 2^8 = 2^{13}$

$\rightarrow 20.21.64.0 / 18$

\hookrightarrow clasă de maxim. 2^{14}

$$64 \left\{ \begin{array}{l} 20.21.64.\times \\ \dots \\ 20.21.127.\times \end{array} \right.$$

$\rightarrow 64 \cdot 256 = 2^{14}$

$\rightarrow 20.21.128.0 / 17$

\hookrightarrow clasă de 2^{15} , jumătate de clasă B

$$128 \left\{ \begin{array}{l} 20.21.128.\times \\ \dots \\ 20.21.255.\times \end{array} \right.$$

$\rightarrow 128 \cdot 256 = 2^{15}$

20.21.255.255 \rightarrow broadcast

$\rightarrow 20.22.0.0 / 15$
 $/ 255.254.0.0 \rightarrow 2 \text{ class B}$

$\rightarrow 20.24.0.0 / 13$
 $\rightarrow 8 \text{ class B} = 8 \cdot 2^{16} = 2^{19} \rightarrow 32-15 = 13$

$20.31.255.255 \rightarrow \text{broadcast}$

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

$\rightarrow 20.64.0.0 / 10$ $\curvearrowleft 256-64 \text{ class B} = 192$
 $/ 192.0.0$
 $\rightarrow 64 \cdot 2^{16} = 2^{22}$

$\rightarrow 20.128.0.0 / 9$
 $\rightarrow 128 \text{ class D}, 128 \cdot 2^{16} = 2^{23}$

$\rightarrow 21.0.0.0 / 8 (2^{24})$
 $\rightarrow 1 \text{ class A}$

$\rightarrow 22.0.0.0 / 7 (2^{25})$
 $\rightarrow 2$

$\rightarrow 24.0.0.0 / 5$ $\curvearrowleft 256-8 \text{ class A} = 248$ Obs: max size network are 255
 $/ 248.0.0.0$
 $\rightarrow 8 \text{ class A}, 8 \cdot 2^{24} = 2^{27}$

$\rightarrow 32.0.0.0 / 3 \rightarrow 32 \text{ class A}$

$\rightarrow 64.0.0.0 / 2 \rightarrow 64 \text{ class A}$
 $/ 192.0.0.0$

$\rightarrow 128.0.0.0 / 2$ → pentru a nu depăși 192.168.17.95 iau clase cu 64, iar ne cu 128

$\rightarrow 192.0.0.0 / 9$
↳ jumătate de clasa A / 128 clase B

$\rightarrow 192.128.0.0 / 11$
↳ pot să iau 32 clase B → $32 \cdot 2^6 = 2^{11}$, $32 - 21 = 11$

192.159.255.255 → broadcast

$\rightarrow 192.160.0.0 / 13$
↳ pot să iau 8 clase B → $8 \cdot 2^6 = 2^{13}$, $32 - 19 = 13$

192.167.255.255 → broadcast

$\rightarrow 192.168.0.0 / 20$
↳ pot să iau 16 clase C → $16 \cdot 2^8 = 2^{12}$, $32 - 12 = 20$

$\rightarrow 192.168.16.0 / 24$
↳ clase mici

$\rightarrow 192.168.17.0 / 26$
↳ pot să iape cu 64

$\rightarrow 192.168.17.64 / 27$
/ 255.255.255.224
↳ clase de $32 - 2^5$, $32 - 5 = 27$

$\Rightarrow \boxed{192.168.17.95} \rightarrow \text{broadcast}$

→ clase de o singură adresă IP

→ decon apă avut .36 ⇒ 192.168.17.56 / 32
/ 255.255.255.255 (nu e clă)

Problema: ① 80. 81. 82. 83
82. 83. 84. 85

- Precizați cea mai mică poziție deținătoare > care să îl conțină pe ambele (cu cel mai mic număr de adrese IP și netmask cu căt mai puțini biti de 0)

$$80 = \boxed{0101\ 0000} . 81. 82. 83$$

$$82 = \boxed{0101\ 0100} . 83. 84. 85$$

$$\begin{aligned} \rightarrow \text{netmask: } & 1111\ 1100. 0.0.0 && \begin{cases} 6 \text{ biti de 1} \\ 26 \text{ biti de 0} \end{cases} \\ & / 252.0.0.0 \\ & / 6 \end{aligned}$$

\rightarrow dacă facem & pe biti \Rightarrow clasa 80.0.0.0 \leftarrow adresa de rețea
83.255.255.255 \leftarrow adresa de broadcast

Răspuns: 80.0.0.0/6

② În cîte clase au netmask 255.255.255.252 se împarte clasa de dimensiune maximă, care are adrese de broadcast cu clasa de dimensiune maximă, care tragește la adresa 193.244.0.0?

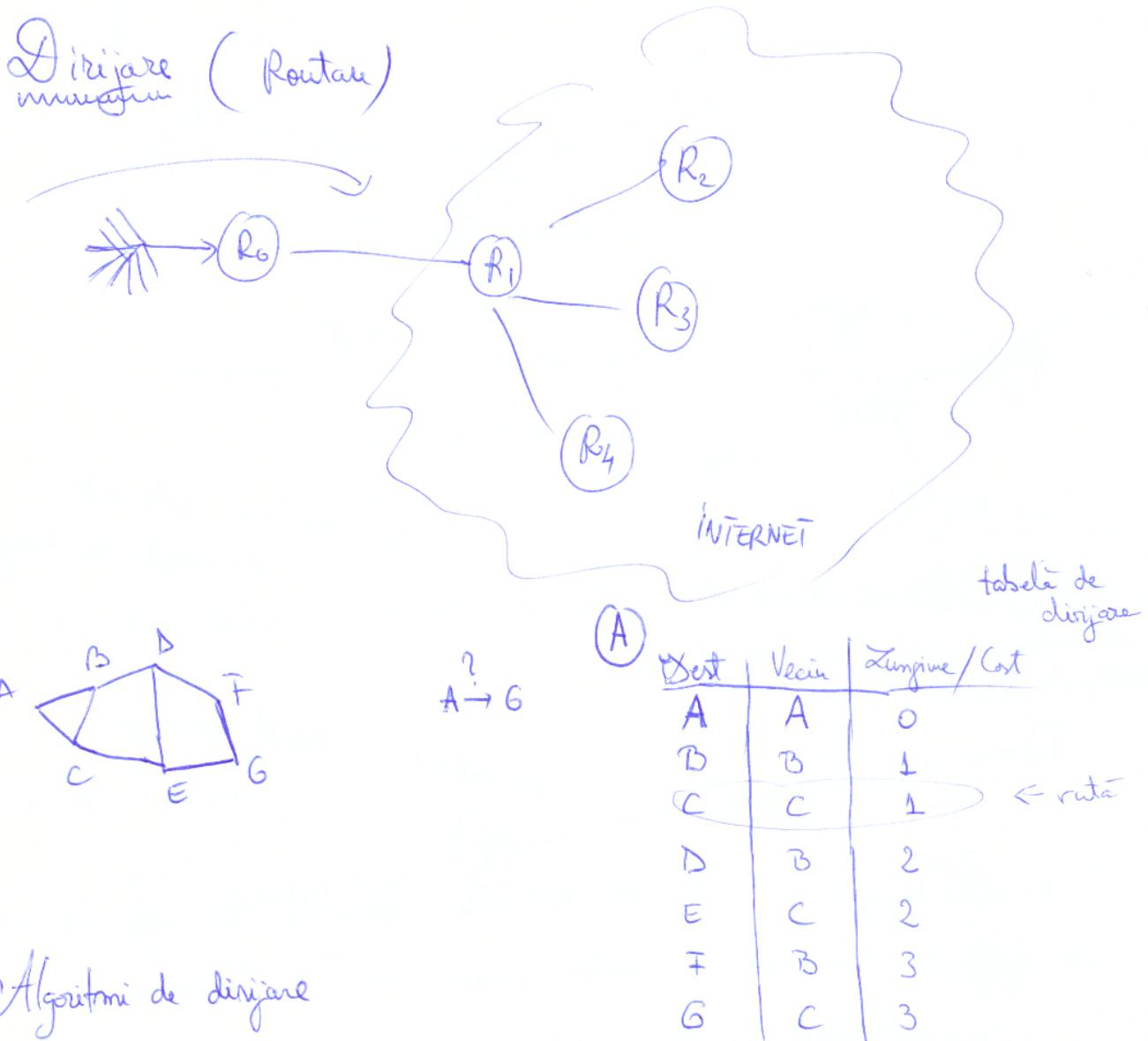
\rightarrow 193.244.0.0 \leftarrow adresa de rețea

\rightarrow 193.244.255.255 \leftarrow adresa de broadcast

193.248
 \hookrightarrow multiplu 8 \rightarrow clasa de 8 agregată

Cursul 10

Dirijare (Routing)



Algoritmi de dirijare

↓

Protocoale de dirijare

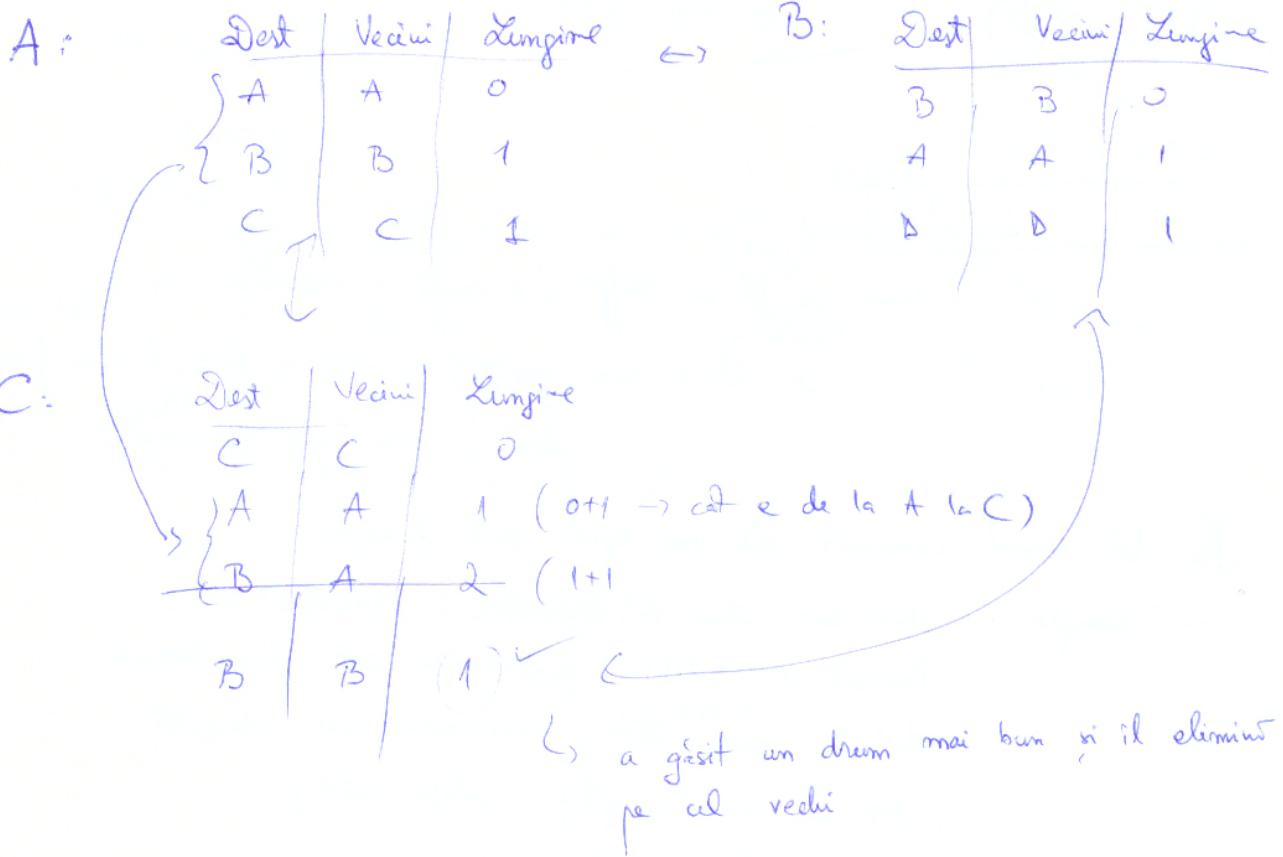
(implementate și rețele de Rute)

- a) vectori distanțe (Bellman-Ford)
- b) stocuri legăturilor (Köigkstra)

Protocol
 ↗ RIP (Routing Internet)
 { BGP (Border Gateway Protocol)

o stare initială (primul rând)

a) Schimb de informații între tabelile de direcție ale routelor, primul rând

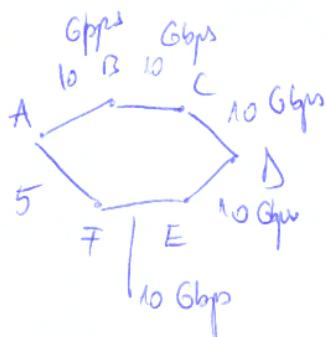


o Tabelele de direcție se actualizează în cazul în care :

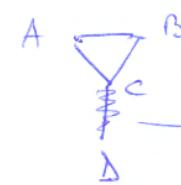
- se găsește o rută mai bună
- se pierde o legătură
- se notifică noii vecinii, pentru a exista sincronizare între tabelele de direcție ale routelor

↳ problema numărării la infinit (suplementar)

Ex:



3GP → ia în calcul și viteza de transmitere, și se formează rută



pe că legătură,
 $A \rightarrow B : 3$
 $C \rightarrow D : 4 \dots$

Dest	Veciui	Lungime
F	?	

Rip de fizică:

F 1

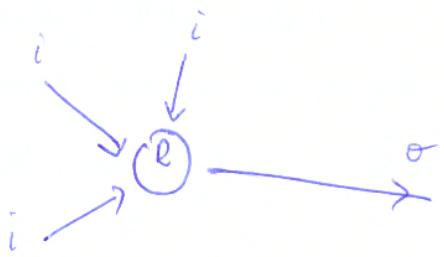
folosind doar didactic

3GP ar trebui:

B 5

folosind în practică

viteze mai mici de transmitere a datelor



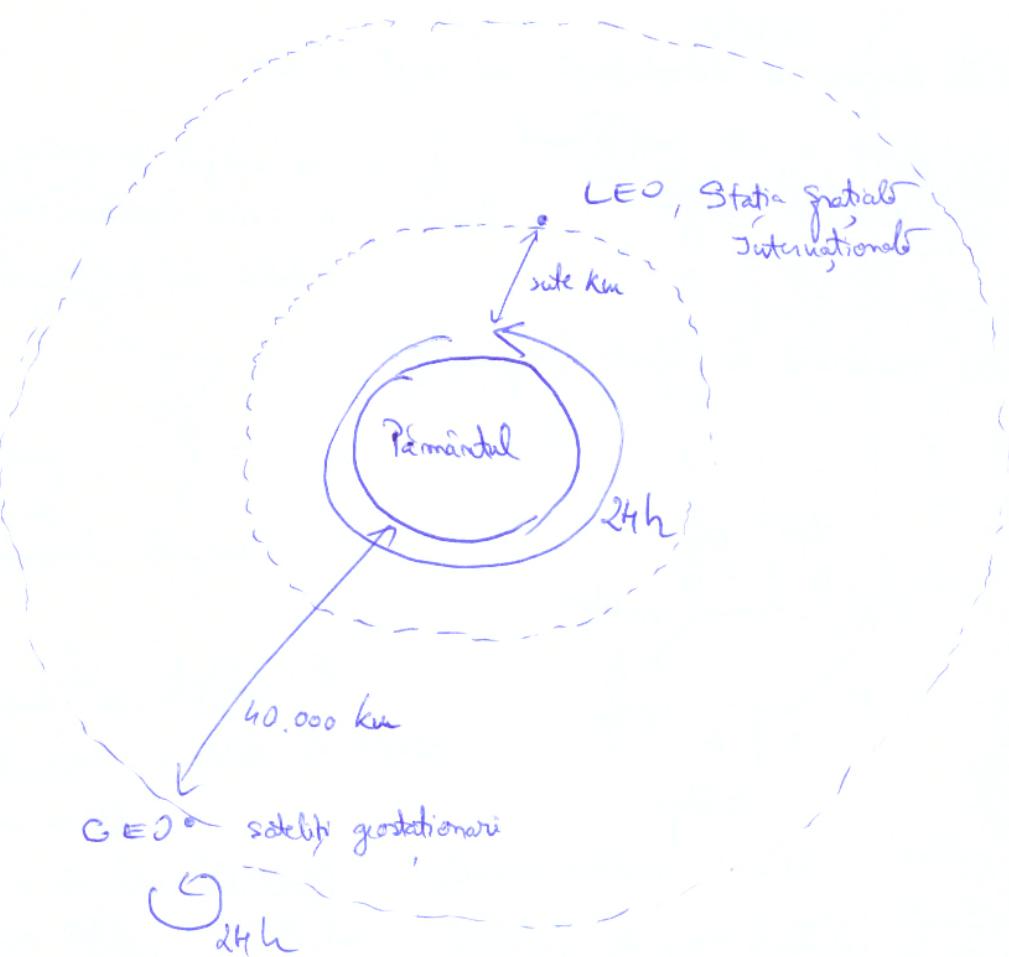
II. III

↳ un fel de buffer / o coadă a rețelei

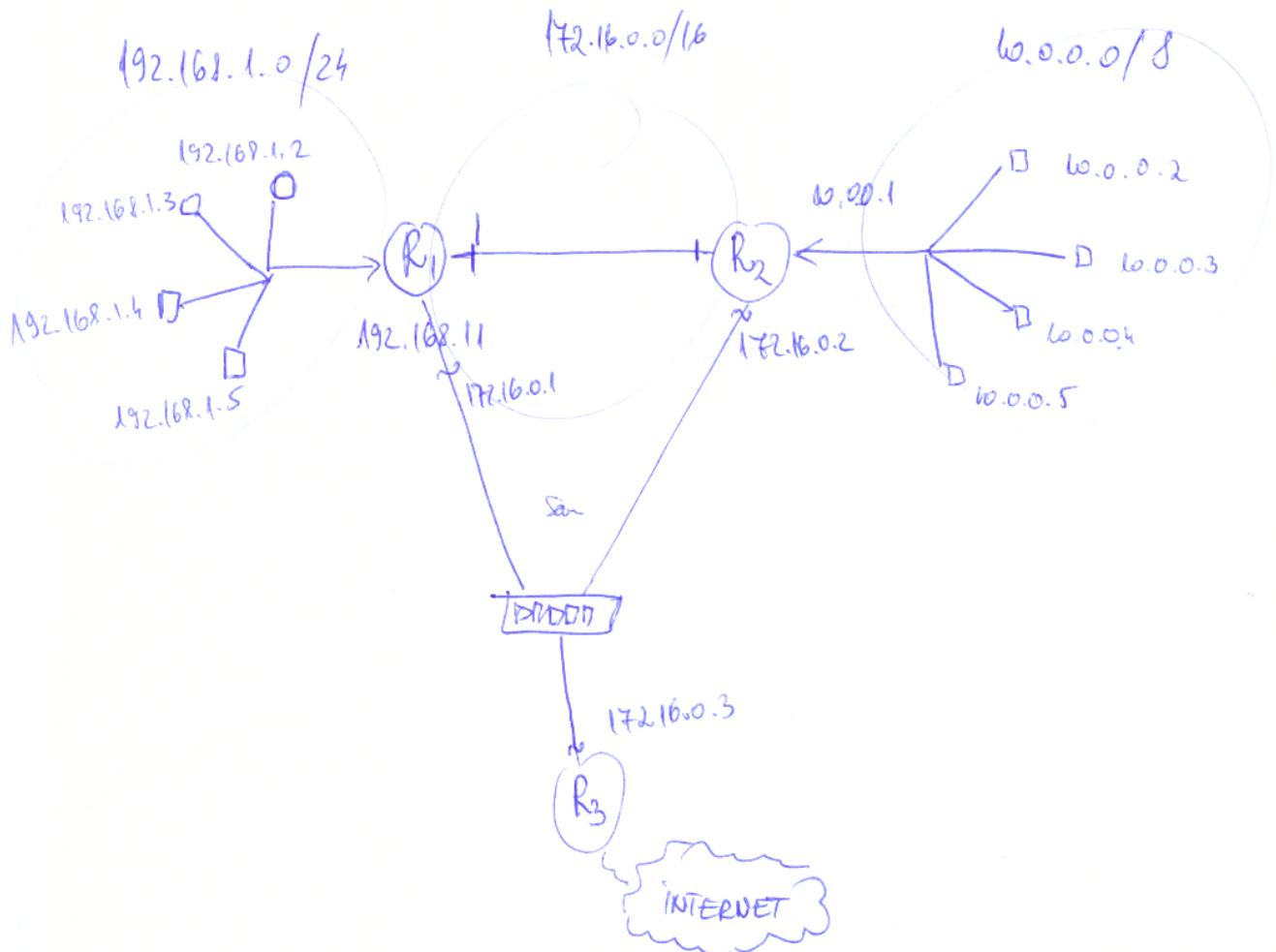
- cu cât e mai goată, cu atât e traficul mai fierb
plină, -u- trafic aglomerat

! Nu tot tipurile conexiunile cele mai rapide sunt favorizate

→ exemplu: chiar dacă viteza de transmitere a informației este
Până în sateliți geostationari este viteza luminii,
distanța mare face comunicarea mai întâzată
(față de comunicarea a sateliților LEO)



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



192.168.1.2 —————> 10.0.0.3

(R1):	Dest	Viaj
S	10.0.0.0/8	R ₂ 172.16.0.2
	0.0.0.0/0	R ₃ 172.16.0.3

→ rețea face să pe fie
toate adrese ip unde trebuie
nu ajungă date și tot
netmask -urile din tabela
de dinigore

(R2):	Dest	Viaj
S	192.168.1.0/24	R ₁ 172.16.0.1
	0.0.0.0/0	R ₃ 172.16.0.3

↳ rută statică

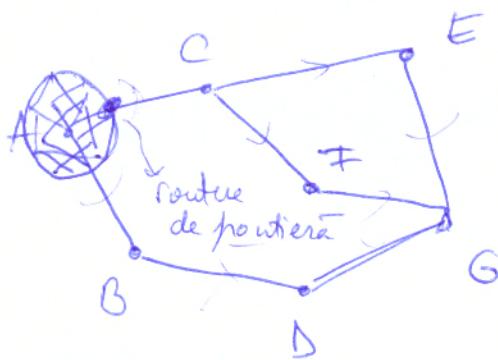
Dest	Viaj
0.0.0.0/0.0.0.0/0	192.168.1.1

/0 ↙

Rută Default

Comanda: route print
pe Windows

Cursul 11



A	Destination	Viaj	Metrică
rate \rightarrow			
0.0.0/	192.168.1.1		
0.0.0.0			

route default (\hookrightarrow spre origine) (default gateway)

adresa routierului

(continuare curs 10)

b) starea legăturilor

$(A, C, 1)$ timestamp
 $(A, B, 1)$ timestamp

peste cunoșterea
celor mai recente
informații

informatii trimise atât lui B, cât și lui C \rightarrow acesta trimis mai departe

\hookrightarrow tehnica inundării

\rightarrow în realitate, în interior rețelelor provider-ilor sunt

folosiți algoritmi IGP, iar în exterior (mai general) algoritmi a)

\hookrightarrow între provideri

\rightarrow BGP

Border Gateway Protocol

\hookrightarrow protocol de schimb inf. (tabele de digăz)

între routere de frontieră

Sisteme autonome (AS)

nr. AS \rightarrow fiecare provider are un nr. AS, pe care se identifică între providerii din eșaf

public route servers
/ routers servers

SSH / Telnet
PuTTY \rightarrow

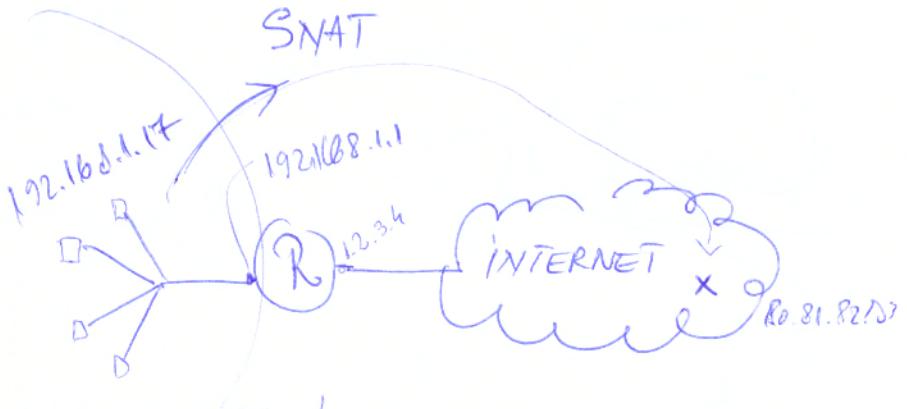
VoceRoute

mai intens computațional

a) vs.
fiecare R face
toate calculele
pentru stabilire
tabele de digăz

entă probleme
numerice
de infinit

\downarrow
nu se
protejează pe
rețele foarte
mare



$192.168.1.0/24 \rightarrow$ Iată false (non-routabile)

$80.81.82.83/24$

Căse false/private

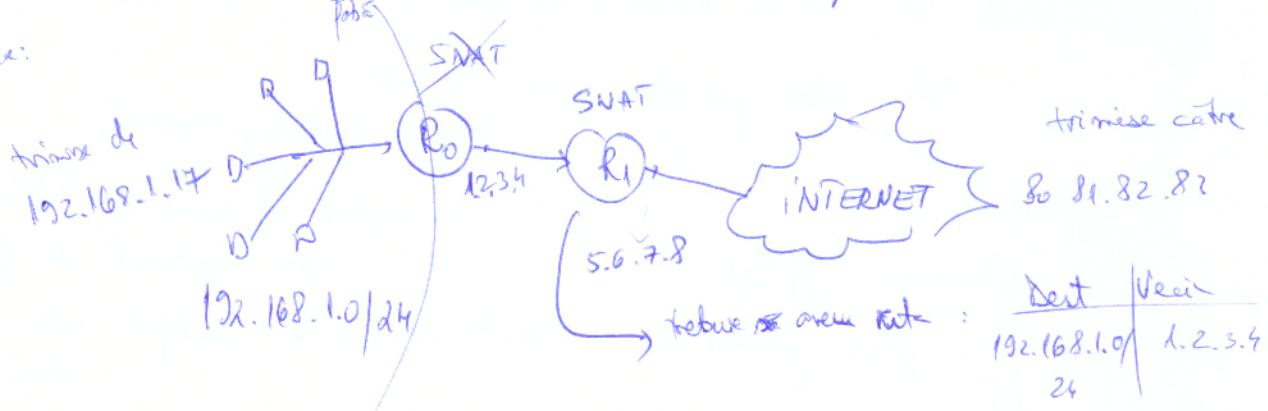
$6.0.0.0/1$ } căse A

$172.16.0.0/16$ }
 $172.31.0.0/16$ } 16 căse B

$192.168.0.0/24$ }
 $192.168.1.0/24$ } 256 căse C
 \vdots
 $192.168.255.0/24$

[SNAT] → adresa ip privată este schimbată în adresa reală pe care o are routerul pe interfața externă

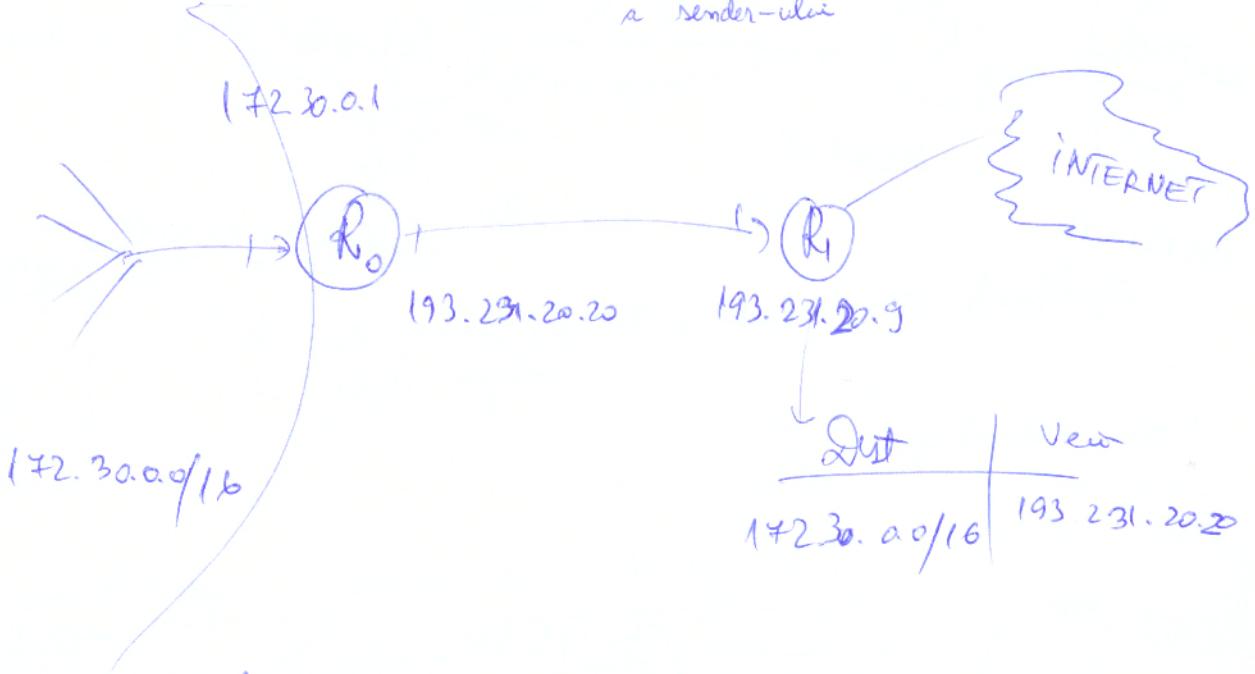
Pb. exemplu ex:



- Routerul R0 nu face SNAT. ? au calculatoarele acces la internet?

(calculatorul de poște face ca ele să aibă acces la net?)

Răspuns: R₁ trebuie să facă SNAT și să memorizeze ruta spre destinat față
 192.168.1.0/24 (↳ deci transformă răspunsul venit pe 5.6.7.8
 în adresa ip falsă)
 și să renvoie către
 a sender-ului



• nu are setat default gateway

videoproiectoare

172.16.0.1/16

172.30.0.0/16

Restul din campus

172.30.0.1

INTERNET

so adăugat ruta:

Dest	Via
172.16.0.0/16	172.30.0.4

172.16.0.2

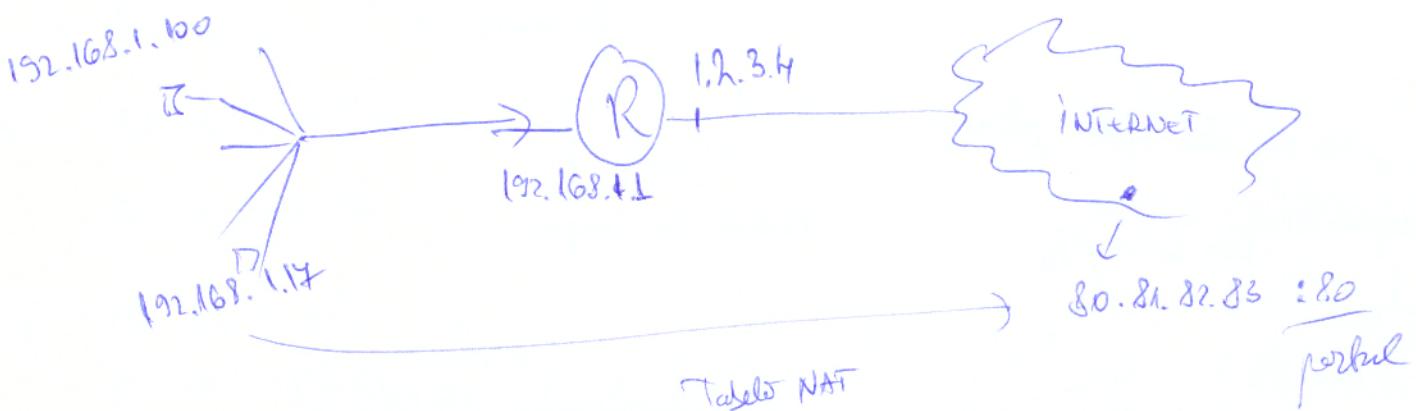
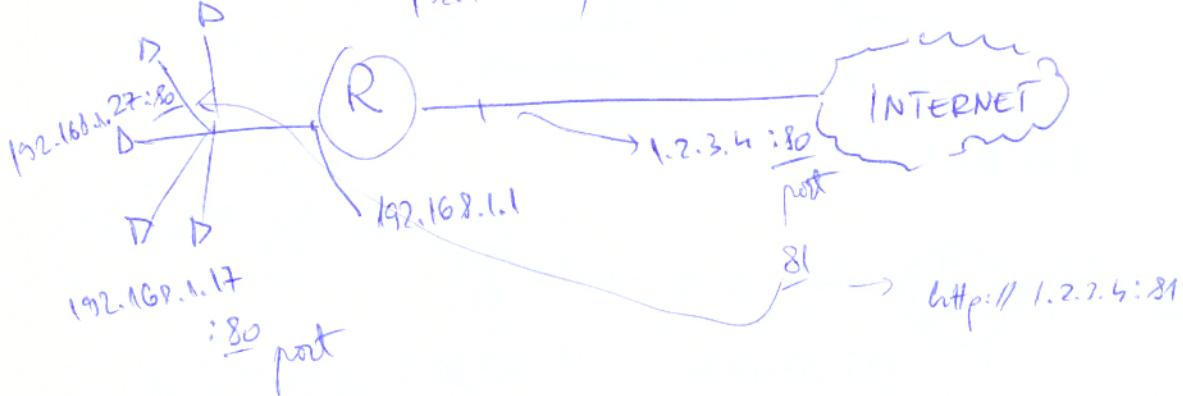
jocuri
multimedie

serverul

172.30.0.4

SNAT → acest puter să trimite videoproiectoarelor date影像,
 și să redirecționeze adresa 192.16.0.2 ca fiind din
 același rețea ca el și serverul poate converti
 adresa în adresa a sender-ului

port forwarding
D-NAT (virtual servers) - traducerea adresei destinație
 192.168.1.0/24



Packet imană & SNAT

ip src	Port src	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		

Table NAT

Packet după
SNAT

IP src	Port src	IP dest	Port dest
1.2.3.4	2179	80.81.82.83	80
1.2.3.4	5076	80.81.82.83	80

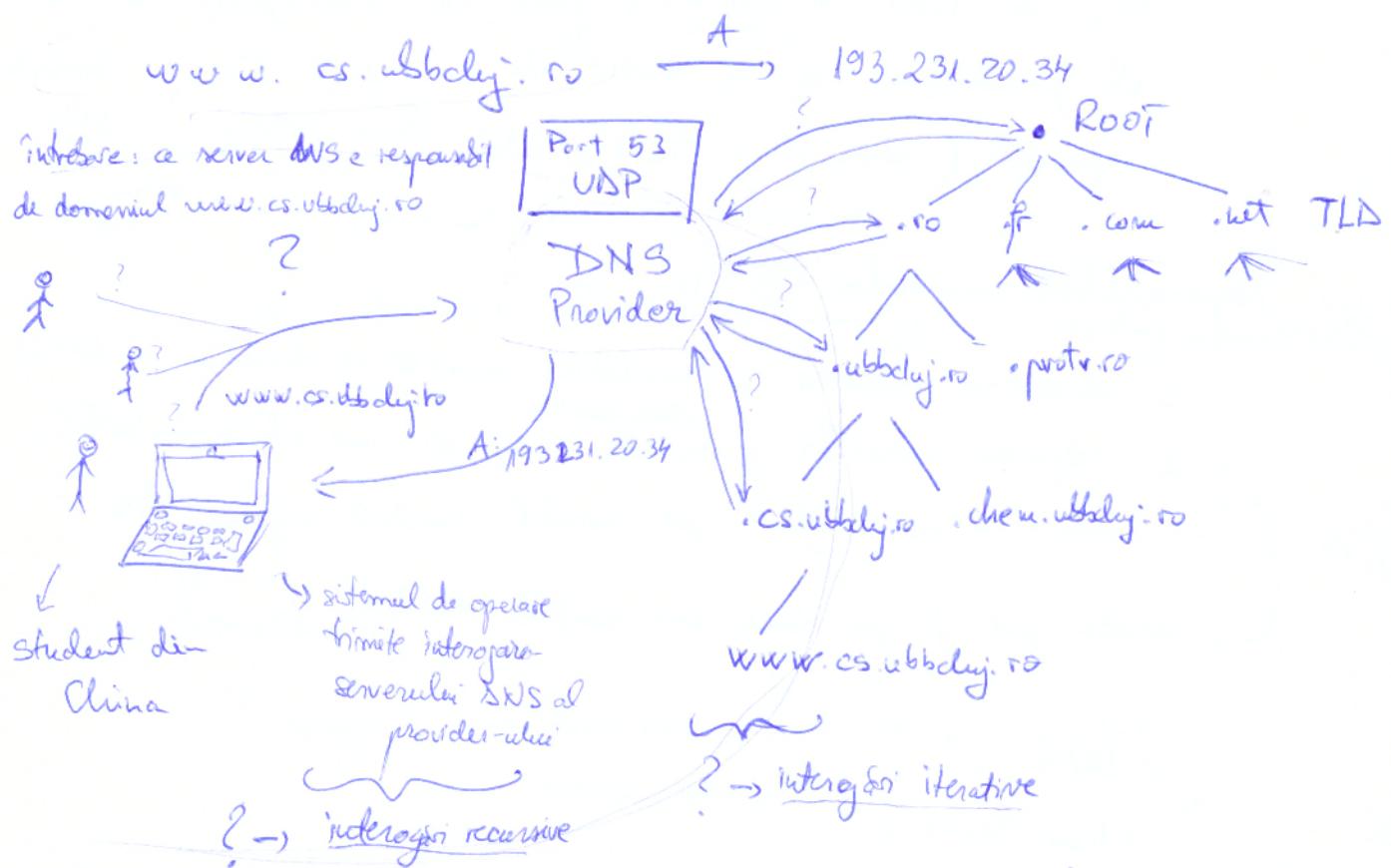
(2179 → în concurs
pe 80.81.82.83)

✓ 2180 ✓

→ se poate întâmpla ca să nu am altă conexiune pe portul 2179,
 deoarece routerul are folosit portul 2179
 (pentru o conexiune deschisă de către exterior)

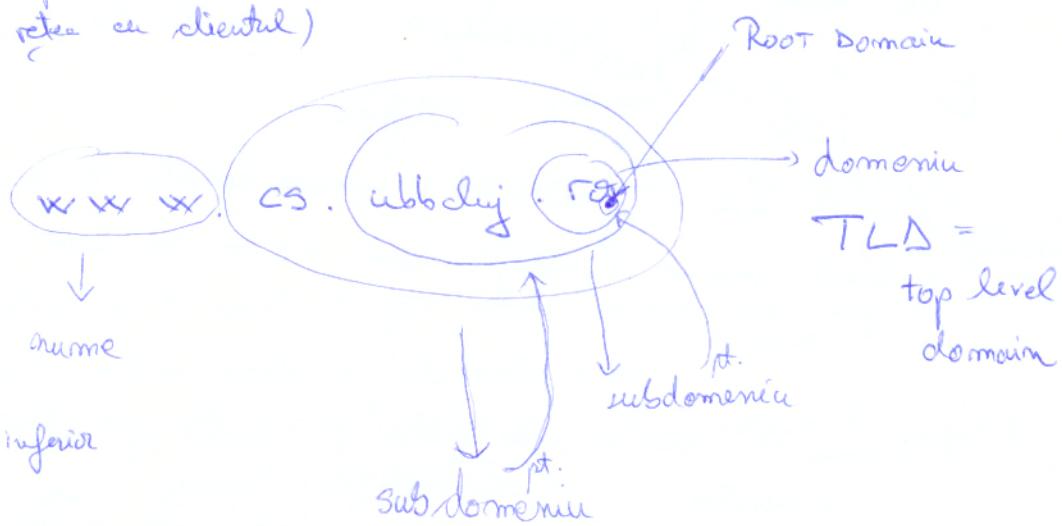
Curgul 12

DNS - Domain Name System



DNS → e plasat la nivel aplicație, ceea ce înseamnă faptul că serverul poate fi situat oriunde în internet (adica nu neapărat în același rețea ca clientul)

* fiecare nivel superior din arbore este responsabil să retină adresele serverelor de pe nivelul imediat inferior (fără direcți)



3600 s

TTL - Time To Live - mai multe inf. "refinute", "cache" ușoare

ipconfig / displaydns - arată adresa reținută, căt timp TTL > 0

/ flushdns - stergerea informațiilor

ea:

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

Responsabilitatea serverului DNS

- 1. deservirea clientilor → translatărea numelor de domeniu în adrese IP
- 2. reținerea adreselor (serverul DNS) de care este responsabil, adică a celor aflate pe nivelul imediat inferior ~~lui~~
- , acestea pot fi pe rând sau ambele deobicei

DNS 1.

Name Server 2.

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



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

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

↳ recursiv

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

↳ iterativ

cs.ubbcluj.ro

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

Tipuri de înregistrări DNS

- A → adresă
- AAAA → IPv6
- PTR → traduce adresa IP în nume de calculator
- NS → Name Server (cele de sus contin înregistrări despre rețele de jos - în arbore)
- MX → pentru mail
- CNAME → Canonical Name → alias-uri
 - ex: www.finomat7.ro este un alias pt. hosting.provider.ro
- TXT → orice

finomat7.com

? merge → doar dacă ansele traduc, în final,
aceeași adresa IP

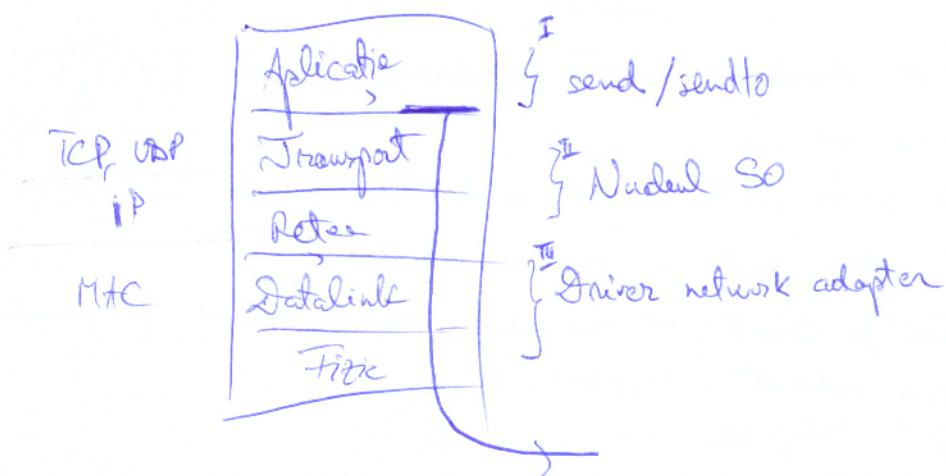
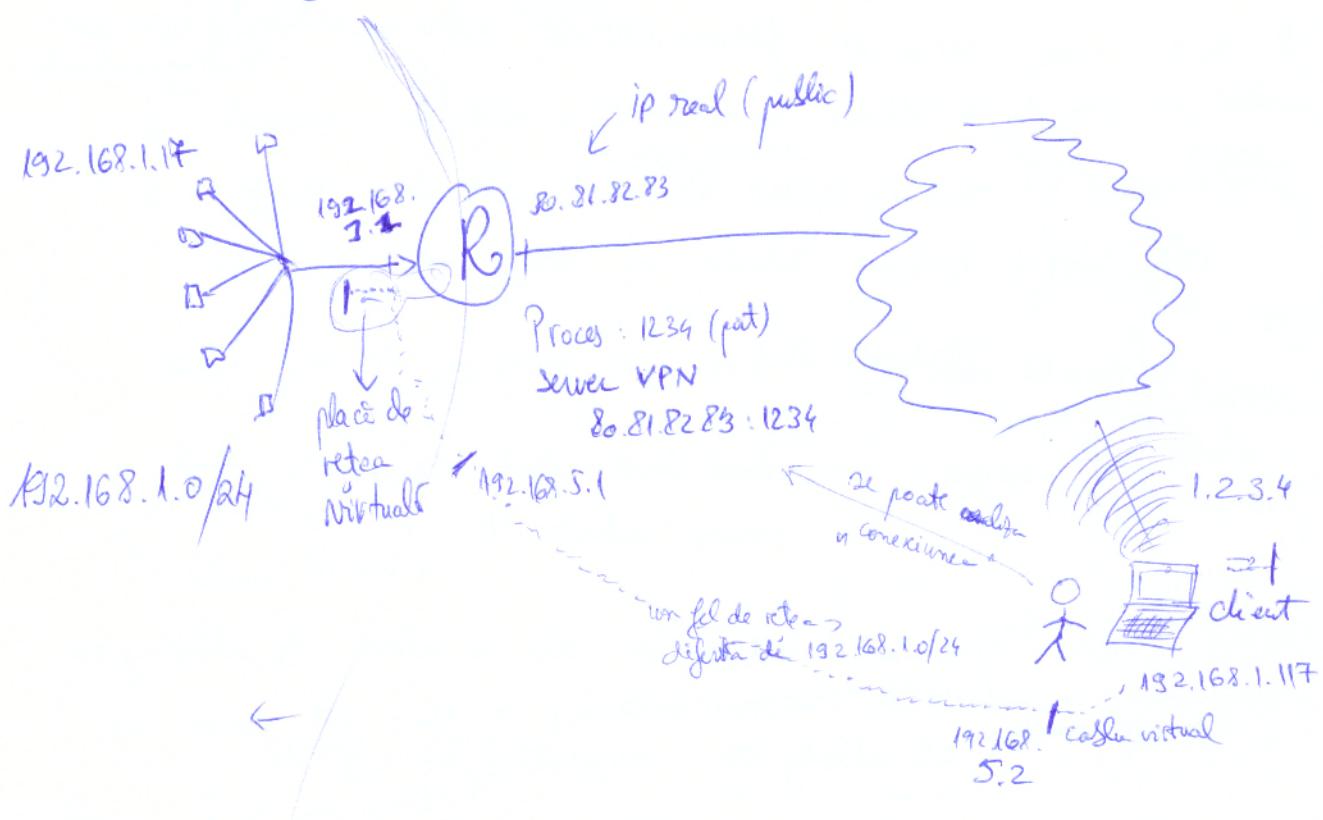
merg ansele ↗
www.cs.ubbcluj.ro are ca alias cs.ubbcluj.ro
cs.ubbcluj.ro → 193.201.20.34

→ să zicem că următoarea adresa IP a serverului fin 1.2.3.4.
ce face?

↪ trebuie să ţin cont că pot exista servere care nu au fin
mai multă vechie a serverului DNS
⇒ voi fi verificat și merg să verific adresa în paralel cu
noia adresa

Curgău 13

VPN (Virtual Private Network)

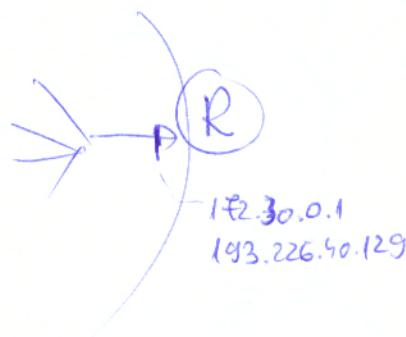


obi: atât client, cât și serverul VPN
trebuie să aibă
place de retea
virtuală

- De ce avem pe o placă de rețea mai multe adrese IP?
(2 interfețe ridicate pe aceeași placă de rețea)

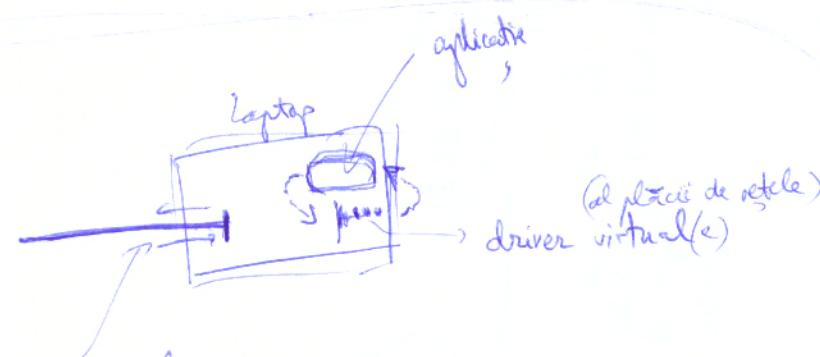
192.30.0.0/16

193.226.40.128/27



→ pot face asta, deoarece
după port, ce rol să
îndeplinească dispozitivul
(care nu poate fi și server
WWW, și altceva).

* la nivelul III se verifică
dăsă o adresa face parte
dintr-o altă plăcă de rețea
→ dacă => urcă în nivel
→ nu => se comportă ca
router și
trimite mai departe



So trimit date

driverul virtual

? = pe că
trimit datele ✓

↓ cu?

nu în exterior,
date pot să o

aplicatie
(la nivel proces)

192.168.1.0/24

12.34
80.81.82.83
192.168.5.2
192.168.5.1



Client de VPN

Host
192.168.1.0/24
0.0.0.0/0
80.81.82.83/32
default gateway (verde)

Gateway
192.168.5.1
192.168.5.2

192.168.5.4 → ! modifica default gateway - wlan