

CURS 1

- NOTA FINALĂ
 - 40% lab.
 - 60% exam. scris (grile)

Cum funcționează internetul?

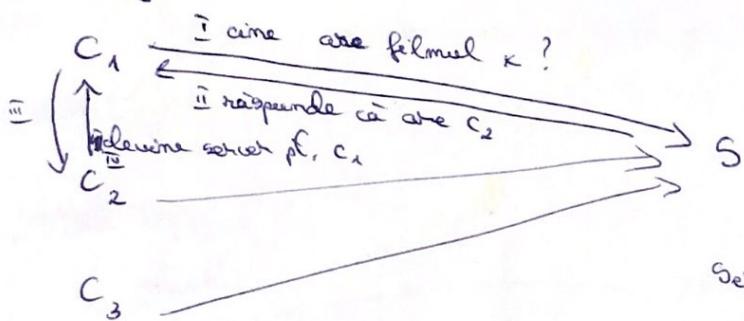
- Principiul adresării IP

- adrese IPv4
- adrese IPv6

- Mecanismul de dirijare (Routing)

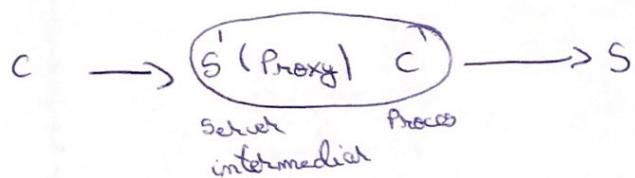
• DNS (Domain Name System) = mecanismul prin care un NUME DE CALCULATOR se translatează într-o ADRESĂ IP

- Paradigma client - server

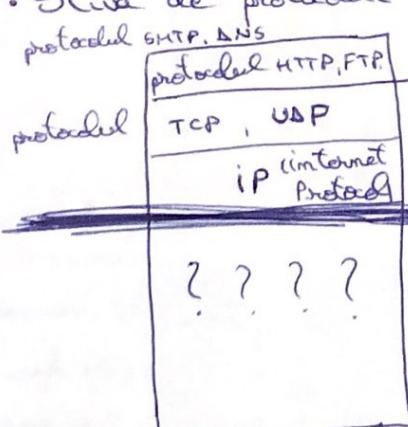


client de Torrent

• Server Proxy



- Stiva de protocoale TCP / IP



Nivel aplicatie

Nivel transport

Nivel rețea

Nivel date Link + Nivel fizic

RFC-uri

(Request For Comments)

WEB

Aplicatie (aici se află protocoalele HTTP, FTP, DNS, etc.)

- protocoale → RFC (publice, specifică documentație)
proprietary (private, nespecificat)

- protocoale de la acest nivel se bazează pe cele de mai jos

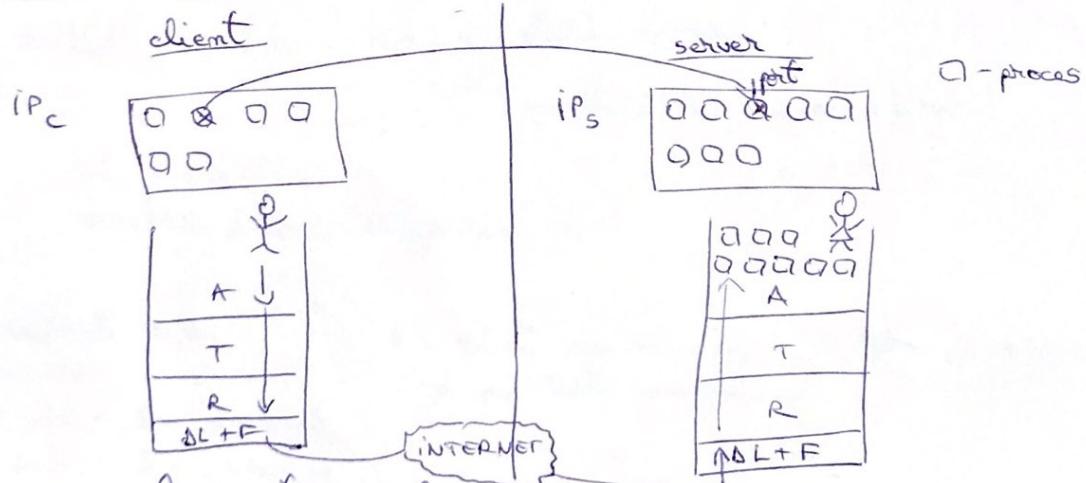
Transport (aici se află protocoalele TCP
UDP)

Retea (Nivel Internet) (aici se află protocolul IP)

Data Link + (cabluri, switch, router,
Fizic)

- Socket = canal de comunicare BIDIRECIONAL care rulează pe 2 sisteme
între 2 procese care rulează pe 2 sisteme diferite conectate

- Socket → TCP
UDP

Socket TCP

• server-ul primește primul

• clientul trebuie să cunoască ip-ul serverului (sistemul/calculatorul)
deschis

C = socket

connect (C, -ip, -port)

send (C)

close (C)

S = socket

bind (S, -port) - apel de rezervare al
-INADDR_ANY unui port

listen (S)

C = accept (S)

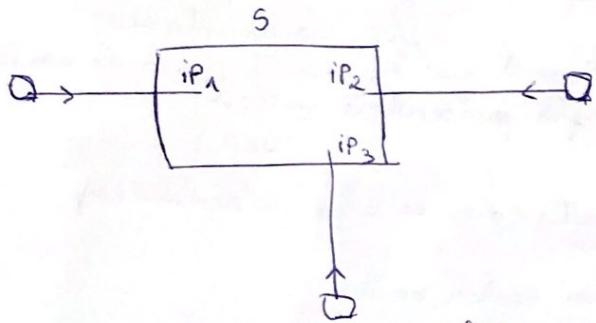
read (C)

include close(C)-f. C include close(S)-f. toti

1

! verificare ce returnează connect
bind

! la connect, ip-ul și portul se pun într-o structură care se dă ca parametru



! bind (s, -5000 ← portul
-INADDR_ANY ← + ip cu portul = 5000

. netstat -an (în cmd) - af. toate conexiunile device-ului la servere

CLIENT:

`uint16_t a; // 16 biti (2 octeti) fără semn 0...65535`

`a = 1234; // 0x0204 (hexad)` // `02 04` în memorie (LittleEndian)

`a = htons(a);`

`Send(c, &a, sizeof(a), ...)`

— — — — — 04 — — — 02 — — — →

SERVER:

`recv(c, &x, sizeof(x), ...)`

→ `02 04` ! Dacă arhitectura serverului = LittleEndian ⇒ interpretare corectă

x = ntohs(x); ! altfel (arhit. server = BigEndian ⇒ interpretare PE săș

! nr. octeti trimisi de client = nr. octeti primitive utilizati de server

! IDEAL: arhit. client = arhit. server. Dacă me e aşa, asig. transform. corectă ce se face!

CLIENT

htons

htonl

SERVER

ntohs

ntohl

! Dacă se rulează pe arhit. LittleEndian:

htons } inversează cei 2 octeti ⇒ BigEndian
htonl

Altfel, me face nimic (me strică nimic)

! For (`i=0; i<m; i++`) { → se strică din cauza ntohs
`x[i] = ntohs(x[i]);`
`send(c, &x[i], sizeof(x[i]), ...);`
 }

! `uint16_t m, x;` // 1 octet, nu mai necesită folosirea htons pt.
`m=100; x=110;` // ceea ce trebuie să fie inversat
`send(c, &m, sizeof(m), ...);`
`send(c, &x, sizeof(x), ...);`

! Dacă `uint16_t m, x` ⇒ htons NECESSAR.

! Dacă trimitem ceva de corect:

`char s[100];`

`sprintf(s, "%s", m);` // s = "100"

`send(c, s, strlen(s))`

```
sprintf(s, "%hu", x); // s = "110"  
send(c, s, strlen(s), ...);
```

- - - - - '1' - - - '1' - - - '0' - - - '0' - - - '1' - - - >

! Nu se pot delimita cele 2 nr. la primire =>
diferenta

=> ~~nu~~ trimitem un delimitator (de ex. "\") între nr.

SAU

trimitem strlen(s)+1 (se va trimite și "\0") între nr.

! GRESIT: recv(c, r, strlen(n), ...); // nu corectam lungimea
// sirului trimis

CORECT: recv(c, r, l, ...); // l = lung. preluari, trimisă înainte de
// sir

! strace ./client // rulează + af. apelațiile sistem

ltrace ./client // _____ mai puțin detaliat

! LA SERVER IMPORATIV:

listen(s, 5); // acceptă conexiunea a 5 clienti deodata, DAR ei vor fi
// "semiconectați" și vor fi deschise
// după ce s-a conectat primul client, ceilalți sunt "semicon".
// și vor fi deschise după deschiderea primului

! LA SERVER CONCURENT:

1 se face un label

2 în FIU: se deschide clientul și se dă exit(0);

3 părintele va continua să accepte ceilalți clienti și să creeze alți fiți (se repetă inst. de mai sus 1 + 2)

• Relare pe calculatoare diferite:

- `gethostbyname()`. (benzii)
- `ipconfig` (comanda Windows + Linux)
- `ipconfig /all`
 - af. si `Default Gate`
 - `DHCP Server`
 - `DNS Server`

de laice sunt aceleasi

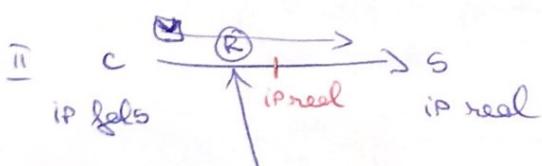
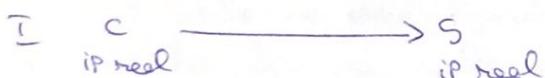
- `ipconfig` (pe Linux) - af. - adresa IP false - prefixate cu 192.168.0 sau
 (private) 192.168.2 (clasa tip C), 172.16
 private 172.31 (clasa tip B), 10.
 (clasa tip A)
- adresa IP reala
 (publica)

- Conectare client la server care nu e local (server-ul de la loc)
- folosire adresa IP reala
- relare ping: www.scs.ulbmfj.ro (apare adresa serverului după

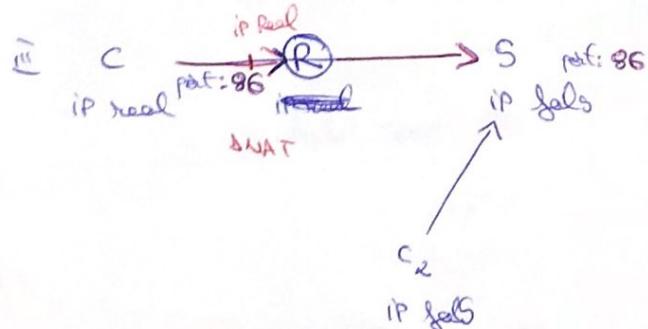
Pinging www.... [...]
 ↑
 ip server

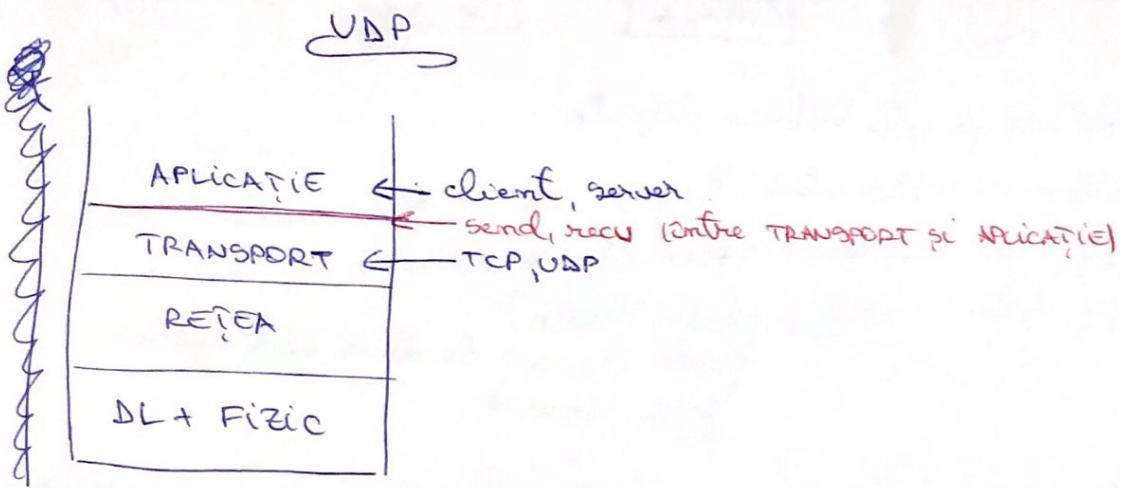
- folosire port intre 10000-50000

CAZURI CONECTARE:



Router (modifică IP false în IP real); serverul va vedea IP-ul routerului (real)





emittor (client)

c = socket

~~connect~~

~~send~~ → sendto(c, ..., &server, sizeof...)

↑
in loc de ~~struct. servet~~

de la TCP, ~~se~~

(se pune adresa struct.)

receptor (server)

s = socket

bind

~~listen~~

~~accept~~

~~recv~~ → recvfrom(s, ..., &client, sizeof...)

↑

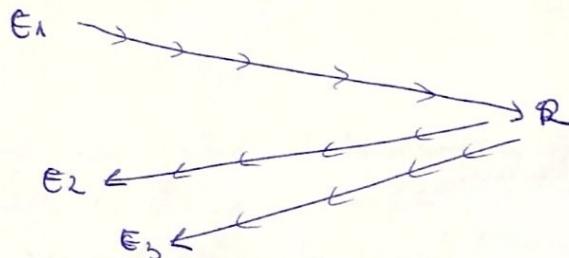
in loc de struct client

de la TCP

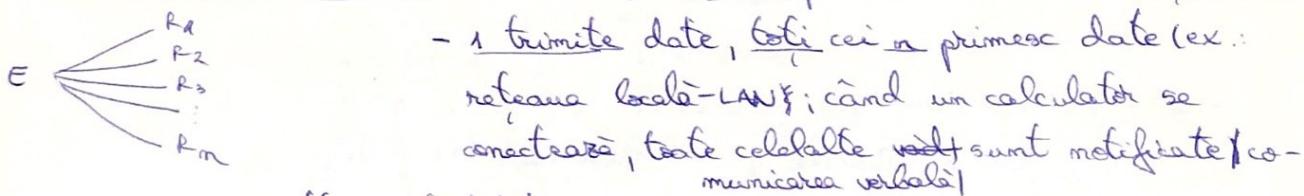
(se pune adresa struct)

Tipuri de trafic

- Trafic unicest -> la 1, între 2 parteneri (emisator-receptor) / client/server)



- > trimite date, & primește date (ex. $E_1 \rightarrow R$, apoi $R \rightarrow E_2, R \rightarrow E_3$; NU $E_1 \rightarrow E_2, E_1 \rightarrow E_3$)
- comunicarea TCP: unicest

Trafic broadcast -> la m (cei m receptori trebuie laci. în rețea locală)

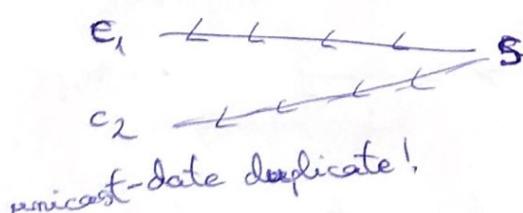
- patterns
 - { - LAN
 - DHCP
- comunicarea UDP: broadcast

? Poate un proces client să se conecteze la mai multe servere?

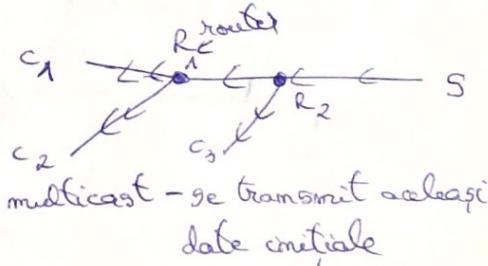
R: Da, prin mai multe socket-uri

Pentru TCP, NU se poate.
Trafic unicest

Pentru UDP, DA.
Trafic Broadcast

Trafic multicest -> la m (cei m receptori NU trebuie laci. în rețea locală)

- HTTP
- RTSP



- arbore de acoperire
- nr. red. 9

- Trafic anycast = 1 trimit, 1 primeste (nu conțineu cine)

Topologii de rețele - arhitecturi

• LAN (Local Area Network)

- ex.: rețea de acasă, rețea facultății din clădirea centrală
- de dimensiuni mici
- dacă se extinde pe dimensiuni mari, e bine să fie împărțită logic în rețele mai mici VLAN (Virtual Local Area Network)
- rețea locală cu difuzare (mediul de transmitere este comun și public)

• MAN (Metropolitan Area Network)

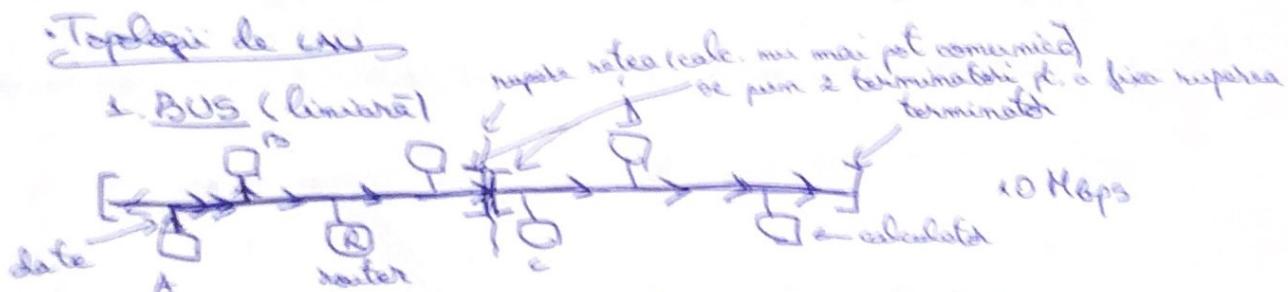
- ex.: toate rețelele publice din UBB Cluj
- de dimensiuni mari
- încopierea mai multe LAN-urilor interconectate

• WAN (Wide Area Network)

- ex.: toate rețelele de la nivelurile 2 și 3 din lume
- de dimensiuni mari ($>$ MAN)
- rețeaua Internet

Topologii de LAN

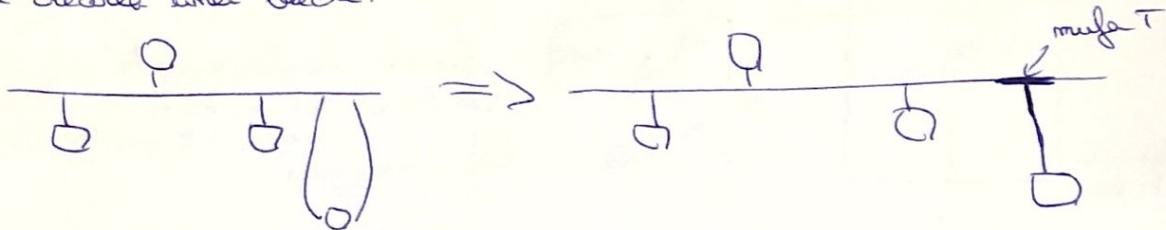
1. BUS (liniar)



- Dacă A transmite date către B, datele sunt accesibile din toate celelalte calculatoare (Broadcast) caracter cu difuzare
- Dacă A comunică cu B și C comunică cu D, doar către A se aplică

- Ce fizică perioche de călătoare care comunică, viteză se înjumătățește

- Dacă se mai dăște conectarea unei calc., se pierde o rețea pt. a evita crearea unei bucle.

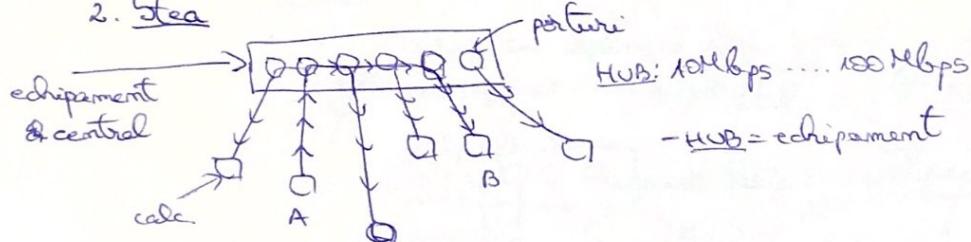


- Dacă se repește mediul de transmisie, picătoarele refacă. Se fixează cu terminatori în stânga și dreapta rețelei

- Router-ul = dispozitiv la nivelul LAN-ului care asigură comunicarea cu alte rețele

- păstrează traficul de broadcast la nivel local și deasupra rețelelor locale
- implementează politici de filtrare = firewall

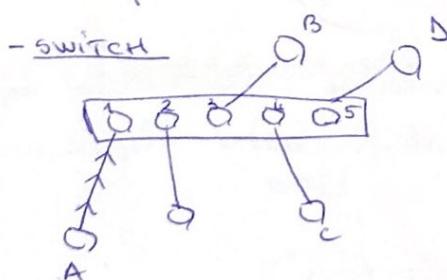
2. Stea



- ~~HUB~~ se menține caracterul de difuzare

- ~~switch~~ = ~~HUB - echipament~~

- Dacă se repește mediul de rețea, este efectuat numai device-ul conectat pe liniul respectiv



- adrese MAC/Ethernet / fibrice

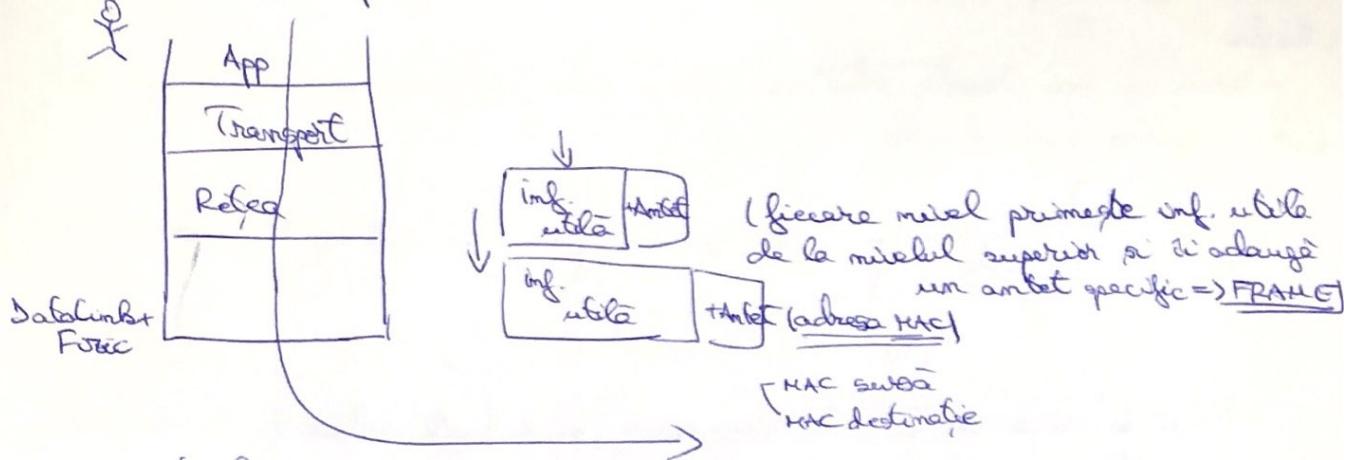
- switch = echipament

- păstrează numărătoarea porturilor

- are memorie și procesor propriu (ex.: $(1, \text{MAC}_A)$, $(3, \text{MAC}_B)$)

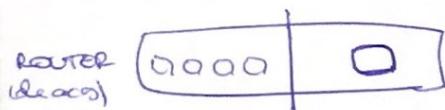
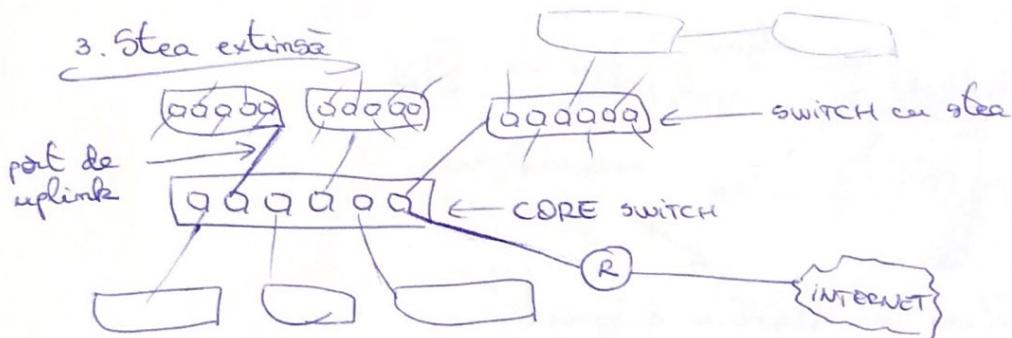
- initial, dacă A trimite, primește boli. Când B răspunde, comunicarea continuă doar între A și B (min. coliziuni)

- adresa MAC opere la nivel Datalinks



APP protocol

- ! · arp -a]- af. teste pe echipe (adresa IP, adresa MAC
(pe Windows))
- ! · arp -m
(pe Linux)
- MAC-ul de Broadcast = ff-ff-ff-ff-ff-ff

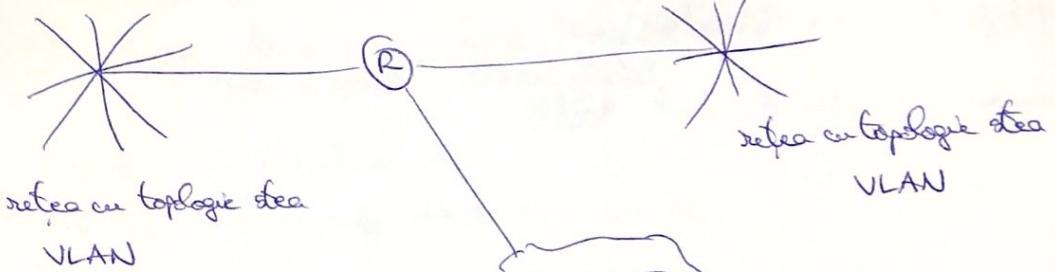


2 placi de rețea : LAN WAN - conectate la nivel Rețea (nu Datalinks !) =>
= > porturi disponibile de pe portul de uplink!

REȚEA ȘCOALĂ (LOCALĂ) CU TOPOLOGIE STEA EXTINSA) - LAN

ELEVI

PROFI



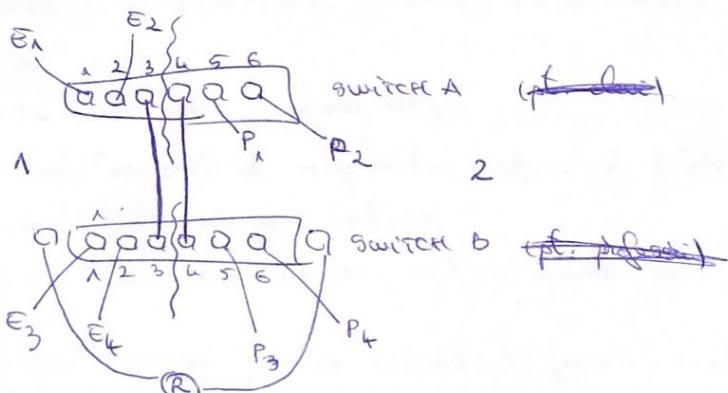
INTERNET

NU
ASA

ETAJ II (LAN ET. II)

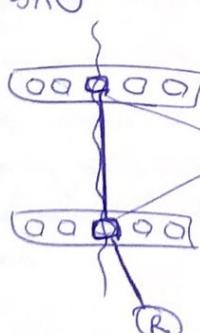
ETAJ I (LAN ET. I)

rețea cu topologie stea extinsă



"se împart" cele 2 switch-uri fizice în 4 switch-uri virtuale

"se împarte" rețelele locale LAN în 2 rețele VLAN (înălțate)
prin Router R
2 switch-uri = 1 VLAN



$1000 \text{ Mbits/sec} = 1 \text{ Gbit/sec}$

$1024 \text{ Mbytes/sec} = 1 \text{ Gbyte/sec}$

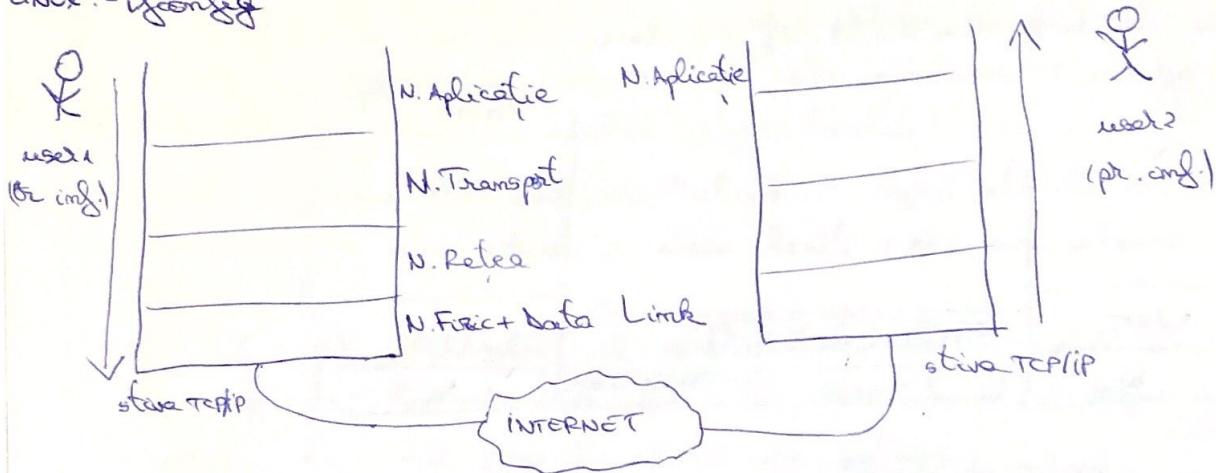
Praederie de internet vând Mbits.

Legătură adresa MAC - adresa IP
 (adresa fizică) (adresa internet)

3-2-1
 2-0-0
 0-3-0

{ ipconfig - af. adresa IPv4
 { ipconfig - all - af. adresa MAC (fizică)

UNIX: - ifconfig



Între N. Retea și N. Data Link e nevoie de o mapare.

La N. Retea: adresa IP

N. Data Link + Fizic: adresa MAC

Partea de la nivel Transport al receptorului arigură că datele se transmit procesorii pe care le-a primit (dorit) dintre cele care reusește.

Fiecare nivel al stivii rețelei comunică cu fiecare nivel al stivii rețelei 2.

Datele se transmit „în jos” pe stiva rețelei 1; fiecare nivel își transmitează datele și le adaugă un ambet (-N).

Protocol arp: ip → mac
rarp: mac → ip

translație comandă
 -arp -a - af. leg. existente între
 adresa IP și adresa MAC

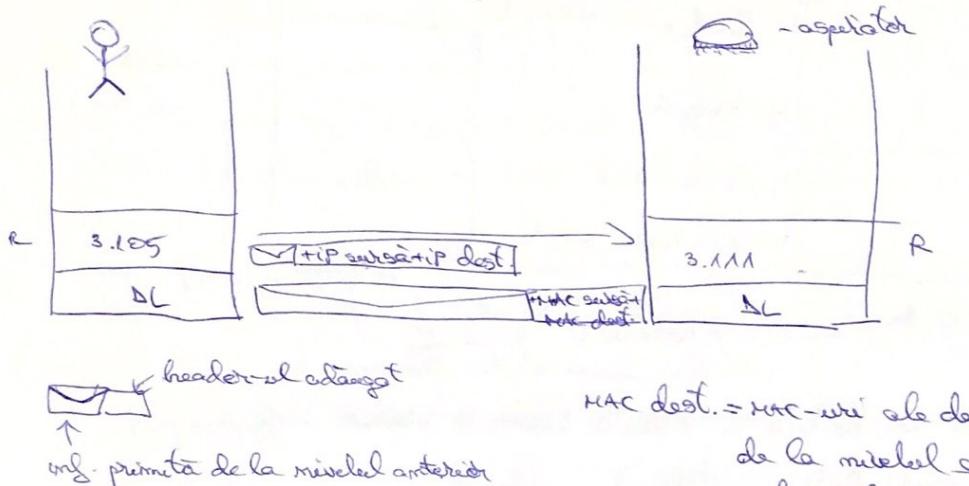
Default gateway: adresa IP a routerului conectat la rețeaua locală (asigură conectivitatea cu exteriorul rețelei locale).

Adresa IP de broadcast ff-ff-ff-ff-ff-ff se folosește p. a tr. date tuturor calei, din adresa locală.

- ping adresă IP device - face leg. device-ului de pe care reținem comanda ce device-ul cu ip-ul dat

\Rightarrow -arp -a - va săptă leg. crește în starea ping-ului
 ↑
 address resolution protocol (ip device ce reținem - adresa MAC device legat) (adresa IP a device-ului legat \rightarrow adresa MAC prin protocol ARP).

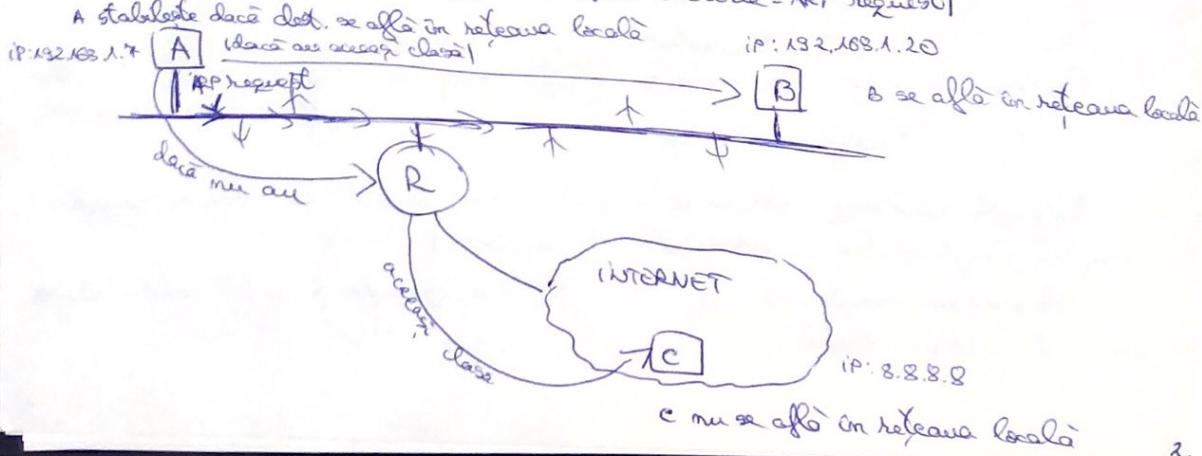
La nivel Schelma: ambtelecădantă conține cele 2 adrese MAC (ale device-ului de pe care se tr. traf. + a device-ului care pl. traf.)



Dacă se trimite date în afara rețelei locale, e nevoie de router (datele se trimet la router, și routerul le trim. mai departe).

- înaint. rețelei locale, NU e nevoie de router

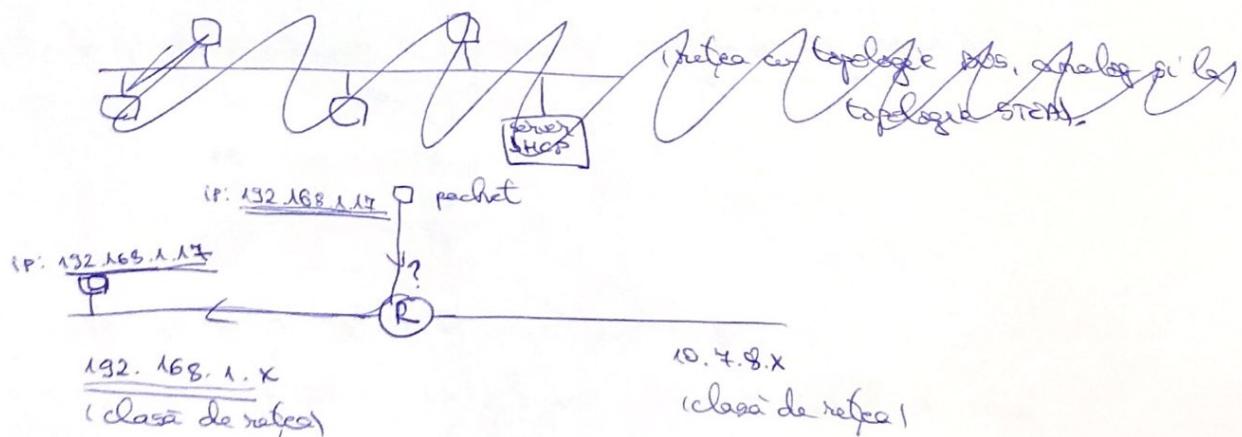
La nivel SL: se pune întrebarea "se face cerere 'Vreau adresa MAC a device-ului ce ip-ul flat de comenzi ping'". Cererea se face pe adresa de broadcast, deci tuturor device-urilor din rețea locală. Device-ul cu adresa IP cerută va fi adresa lui MAC. (Cerere = ARP request)



- RARP = Reverse Address Resolution Protocol (adresa MAC \Rightarrow adresa IP)
- ARP = Address Resolution Protocol (adresa IP \Rightarrow adresa MAC)
- tcpdump - comandă Linux care af. traficul de pe broadcast-ul rețelei locale
- arp -a : pe Windows } af. paralele adresa ip-adresa mac de la
arp -m : pe Linux } nivelul rețelei locale.
- arping - trimite un arp request către o adresa ip menționată if ~~ether~~ ~~arderif~~ (doar la nivelul rețelei locale)
- DHCP = Dynamic Host Configuration Protocol : oferă setările necesare conectării la rețea (oferă > setări decât RARP, care setează doar adresa IP)
- ipconfig -all (Windows) - af. serverul DHCP de la nivelul rețelei locale ifconfig (Linux) - af. plăcile de rețea, adresele IP, default gateway = adresa routerului
- netstat -lnp - af. procesele server care releasează



- Router-ul dă adrese prin DHCP la nivelul rețelei locale (LAN), nu și WAN.



Clase de adrese

- adresă IPv4: 192.168.1.x

$$x \in \{0, \dots, 255\}$$

primii 3 octetii sunt comuni tutelor adreselor IP din rețea locală
 ultimul octet identifică adresa unei anumite echipamente din rețea locală

192.168.1.x

parte comună - | parte specifică

$$255.255.255.0 = \text{netmask} = \begin{cases} 1 & \text{24 biti} \\ 0 & \text{8 biti} \end{cases} \Rightarrow$$

\Rightarrow 24 biti / 3 octetii = comuni tutelor devicelor din r.l.
 8 biti / 1 octet = specific fiecărui device din r.l.

\Rightarrow 192.168.1.0 - IP nefolosibil (identifică rețea locală) - adresă de rețea
 192.168.1.1 - primul IP folosibil (se dă la router)

:

192.168.1.254 - ultimul IP folosibil

192.168.1.255 - adresă de Broadcast

• Adresa de rețea / de Broadcast nu poate fi dată unei device.
 (prima din interval / ultima din interval)

• + IP din interval și netmask \Rightarrow adresă de rețea
 (și pe bitii)

Ex.: 192.168.1.114 și

255.255.255.0

192.168.1.0

• + IP din interval || ! netmask \Rightarrow adresă de Broadcast
 (să se calculeze pe bitii)

Ex.: 192.168.1.114 ||

0.0.0.255

192.168.1.255

• 255.255.255.0 $\stackrel{\text{net}}{=} 1/24$ (pt. că sunt 24 biti=1)

8 biti=0 \Rightarrow $2^8 = 256$ adrese IP posibile pt. clasa
 2^{32-24}

• 192.30.x.y
 $\{ \begin{matrix} \text{net} \\ = 16 \end{matrix}$
 $255.255.0.0$ } = netmask \Rightarrow adresele IP de la nivelul rețelei difere prin ultimii 2 octeti
 $\Rightarrow 2^8 \cdot 2^8 = 2^{16}$ adrese posibile
 192.30.0.0
 :

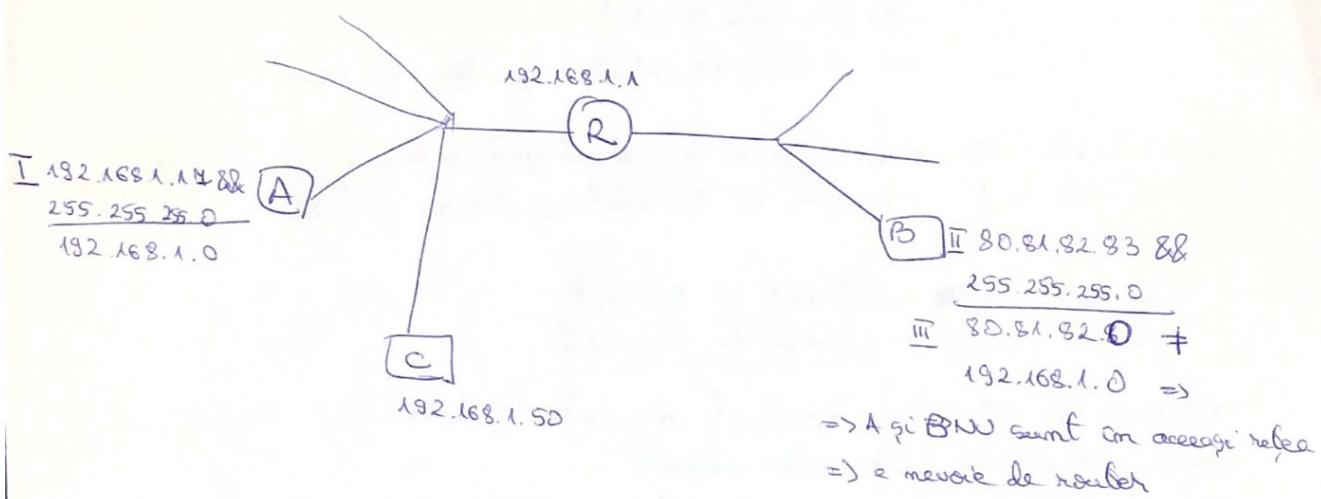
192.30.255.255

192.30.0.0 / 255.255.0.0
 unde încep adr. / netmask

SAU

192.30.0.0 / 16
 unde încep adr. / netmask

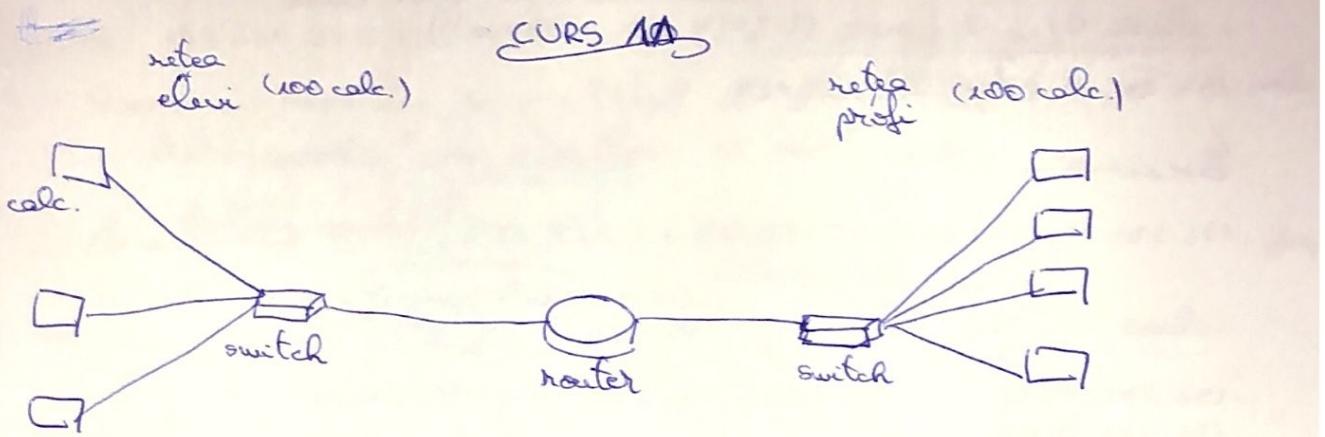
$32 - 16 = 2^{16}$ adrese posibile



CLASE DE ADRESĂ

- A. $\{ 0. \dots 127 \} / 255.0.0.0$
 - an 0 pe primul bit din primul octet
 - fiecare 2^7 val. pos.
 - intervale de
 - 127.0.0.1 adresa localhost
- B. $\{ 128. \dots 191 \} / 255.255.0.0$
 - an 10 pe primii 2 biti din primul octet
 - fiecare 2^6 val. pos.
- C. $\{ 192. \dots 223 \} / 255.255.255.0$
 - an 110 pe primii 3 biti din primul octet
 - fiecare 2^5 val. pos.

Ex.: clasa A, 80.0.0.0 / 255.0.0.0 se poate "rupe" în 256 de clase mai mici cu netmask 255.255.0.0 (clasa B)
 în 65536 de clase mai mici cu netmask 255.255.255.0 (clasa C)



$$193.231.20.0 \mid 255.255.255.0 = \begin{cases} 193.231.20.0 & \text{adresă de rețea} \\ \vdots \\ 193.231.20.255 & \text{adresă Broadcast} \end{cases}$$

Ex.: elev
193.231.20.3 trimite prof.
193.231.20.135

$$\text{netmask.} \\ 255.255.255.0 = 124$$

$$\begin{array}{rcl} 193.231.20.3 \& \& 193.231.20.135 \\ 255.255.255.0 \quad \quad \quad & \& 255.255.255.0 \\ \hline 193.231.20.0 & = & 193.231.20.0 \end{array}$$

\Rightarrow elevul va conchidează că
profila este în același clasa
de adrese (e în același rețea)

\Rightarrow elevul va face ARP req., care va șansa deoarece nu sunt în același
rețea și nu va trece de routerul său

Concluzie: & dintre ip prof. și netmask \Rightarrow adresă rețea + adresă
rețea elevi, deci se va face o altă emisie de adrese (în jurnalul):

Fixare: $193.231.20.0 = 193.231.20.00000000$

\vdots

$193.231.20.127 = 193.231.20.11111111$

2 biti comuni

netmask: $255.255.255.10000000 = 255.255.255.128$

* biti 0 = 125

2 adrese posibile

prof. $193.231.20.000 = 193.231.20.00000000$

\vdots

$193.231.20.255 = 193.231.20.11111111$

2 biti comuni

\Rightarrow

netmask: $255.255.255.10000000 = 125$

$= 255.255.255.128$

ad. rețea = 193.231.20.0

netmask = 255.255.255.128

adr. Broadcast = 193.231.20.128

netmask = 255.255.255.128

adr. Broadcast = 193.231.20.255

! Deci: 1 rețea 256 adr \leftarrow 1 rețea 128 adr. \leftarrow netmask = 255.255.255.128
1 rețea 128 adr. \leftarrow adr. rețea = 193.231.20.128
 \nearrow netmask = 255.255.255.128
 \nearrow adr. Broadcast = 193.231.20.255

Aceasta:

193.231.20.3 & 8

255.255.255.128

$\underline{193.231.20.0}$

!=

193.231.20.135 & 8

255.255.255.128

$\underline{193.231.20.128}$

\Rightarrow rețele diferite \Rightarrow netmask eronat!!!

\Rightarrow nu se face ARP req., și următoare la router

• clase C : 2 ^{1 adr. retea + 1 adr. host/ clase} jum (2²=4 adr. nefolosibile)

clasi: 193.231.20.0 | 255.255.255.128 = 125

193.231.20.

$\frac{125}{32} = 4 \Rightarrow 2^2$ adrese posibile

pagini: 193.231.20.128 | 255.255.255.128 = 125 $\Rightarrow 2^2$ adrese posibile

• clase C: 4 subclase ^{8+8+8+2=26 locuri comune}

193.231.20.0 ... 63 | 26 = 255.255.255.192

193.231.20.64 ... 127 | 26

193.231.20.128 ... 191 | 26

193.231.20.192 ... 255 | 26

! Acestea sunt netmase (\Leftrightarrow au aceeași dimensiune (aceeași nr. de adr. pos.)

• clase C: 8 optimi (2³=8 adr. nefolosibile)

193.231.20.0 | 24 = 255.255.255.224

193.231.20.32

:

• clase C: 16 posibile (2⁴=16 adr. nefolosibile)

193.231.20.0 | 30 = 255.255.255.252

193.231.20.1

193.231.20.2

193.231.20.3

• 193.231.20.14 | 32 = 255.255.255.255 = host only netmask

! 255.255.255.128

$256 - 128 = 128$ = dim. clasei fizicei subclase

doar pt. clasele < 256 adr. pos.

! Ex.: clase C: 3 parti

193.231.20.0 | 255.255.255.128 = 24

193.231.20.0 | 255.255.255.128 = 125

1 clase

193.231.20.0 | 25

193.231.20.128 | 26
1 clase

193.231.20.192 | 26
1 clase

Ex.: 193.231.20.0 / 255.255.255.0 = (24 adr. pos.)

Vrem să împărtim în 4 clase de 16 adr. (clasi)

adr. rețea adr. rețeb 1 clasă 25 adr. (pref)

(0 + 1 + 1 + 1 = 3)

adr. Broadcast

I clasi \Rightarrow clasă de 16 : 193.231.20.0 / 255.255.255.260

193....0 ... 193....15

adr. rețea adr. Broadcast

metivă: 1255.255.255.1110000 = 1255.255.255.260 = (23)

u Bitul 0 $\Rightarrow 2^4 = 16$ adr. pos.

25 + 1 + 1 + 1 = 28 \Rightarrow

II prefii \Rightarrow clasă de 32 :

193....16 ... 193....47
(consecvența de la înălț)

împărțire corectă:

193....0 } \in 193....15
193....32 } \in 193....47
193....64 } \in 193....47

GRESIT! ($\& \& \Rightarrow$ adresa de rețea dif. #)
cel 2 adrese nu vor aparține același
echipaj clasei

rețea prefii : 193.231.20.32

- Dimensiunea clasei = putere a lui 2
- O clasă tab. să încapă la o adr. = multimea de dim. clasei
 \equiv adresa de rețea a unei clase =

Ex.: Dacă împărtim în clase de dim. 128:0

....128

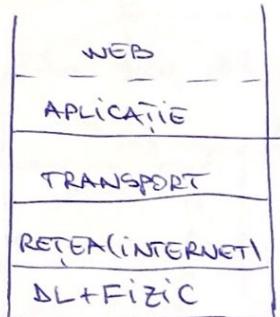
dim 64:0

....64

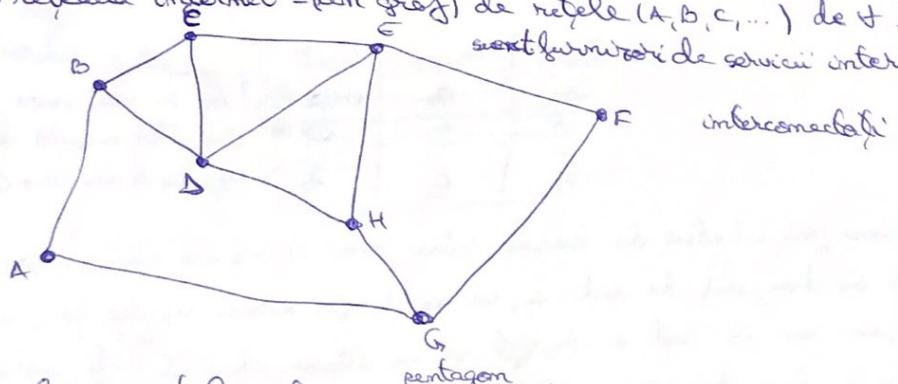
....128

....192

Avg

DirijareLa nivelul retea (internet):

- sunt localizate adresa ip
wrmat, netwerk
- retea internet = (un graf) de retele (A, B, C, ...) de tipul, ex. A, B, C, ...
o rete
sunt furnizori de servicii internet (orange, dtel, etc.)
- (default) gateway = adresa ip interna a routerului
(a device-ului conectat si la afara retelei)
- router
- clasa de adrese



- la inceputul anilor '60 a creat o retea aranjet, care ulterior a devenit baza a ceea ce noi numim azi retea internet.
- tabela de dirijare } \Rightarrow reguli de dirijare (det. dinamic)
- router
- alg. de dirijare = logica din spatele dirijarii, proces ce ruleaza pe modurile
 - dinamici (reguli stabilite on-the-fly)
 - stative (reguli stabilite manual, preconfigurare manuala)
- dirijare larcata
- pe inundare = mecanism prin care un nod replace un pacet pe toate retelele
 - lyricile
 - inefficienta, verificare multe pachete duplicate
 - folosita in retele de dim. mica

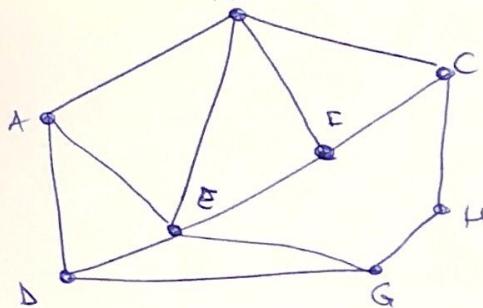
z mari
form.

de I dirijare bazată pe vectori distanță

de II dirijare bazată pe stocuri legăturilor

I - se bază pe alg. Bellman-Ford

tabel de dirijare din percep. nodului A ~~nu~~



= A:

Dest	Vecini	Cost (lungime)
A	A	0
B	B	1
D	D	1
E	E	1

- A va face limită tabelă fiecărui vecin,
analog, și pt. fiecare vecin va trimit lui A

B:

Dest	Vecin	Cost
:	:	:
C	C	1

\Rightarrow A va "invita" de la B să meargă (rest) de dirijare
pe care o adaugă în tabelă lui \Rightarrow regulă

\Rightarrow A:

Dest	Vecin	Cost
A	B	1+1=2
A	A	0
C	B	2

! Nu se mai pună decât
foarte regulă de a
afla A, cu cost < 0

- dacă în urma „schimbului” de tabele între A și B \Rightarrow un „drum” spre un
nod X/să regule de dirijare de cost <, se va șterge vecina regula spre nodul

X și se va pune cea de cost <. Astfel, se va obține „drumul” de cost minim

- protocol care implementează RIPv (Routing Information Protocol)
alg. de dirijare bazat pe vect. dist.

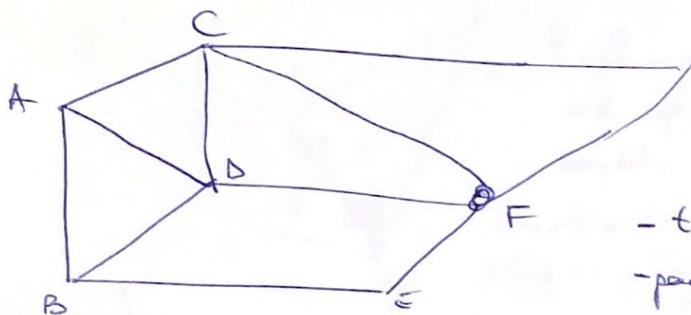
BGP (Border Gateway Protocol)

- alg. baz. pe vect. dist. \nearrow avantaj: sunt mai puțin ușor de calculat
 \searrow dezavantaj - un nod să fie ce se înt. doar în „celulă”
lui de rețea, nu să fie ce se înt. cu
pachetul trimis din restul rețelei internet
(probleme numărării la sa)

- TTL (Time To Live) - se determină că de fiecare router pe care îl primește
un pachet

pachetul; când a aj. = 0, se „strică la graniță”.

II - se bazează pe alg. Dijkstra (nu funcț). pe drumurile cu cost < 0)



pachete:

- Q (A, C, 1, timestamp)
- (A, D, 1, ts)
- (A, B, 1, ts)

- timestamp-ul reprezintă informație
- pachetele se tem. pentru înrandare

- avantaj: fiecare nod are o privire asupra întregii rețele, deci poate calcula drumurile minime folosind Dijkstra, și va emite prot. num. la o

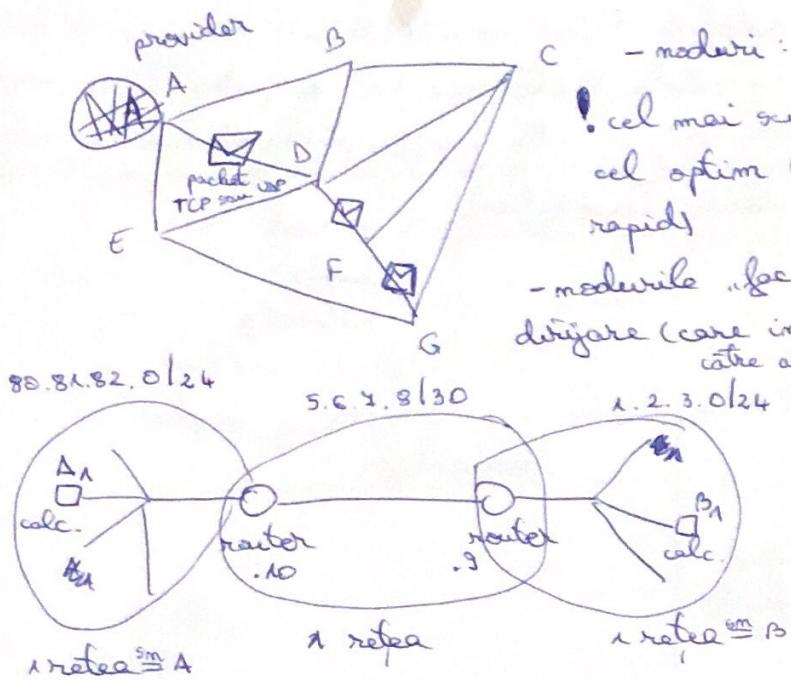
- dezavantaj: alg. intensiv computațional

I - se preferă unele rețele de dim. mici (ex.: rețea fizică furnizor de servicii)

II - se preferă unele rețele de dim. Δ (ex.: întreaga rețea internet, care este o rețea de rețele ale furnizorilor de servicii interconectați)

BGP

- Border gateway = ,



C - reteau: routere / provideri

! cel mai scurt drum nu e neaparat cel optim (poate sa existe altul mai rapid)

- reteurile "fac schimb" de tabele de dirijare (care indică legăturile, rutele) către alte clase de adrese

- comandă: `show ip route` - af. tabele de dirijare $s = \text{static}$

$B = \text{BGP}$
Border Gateway Protocol
route învățate de la "vecini"

www...

- tracert - pe Windows } af. rutele pt. a aj. la adresa specificată
- traceroute - pe Linux } af. rutele pt. a aj. la adresa specificată

- agregare de clase $\xleftarrow{\text{cond}} \text{clase consec.}$ se aj. la combina clase prim recibi $\xrightarrow{\text{sumă}} \text{pt. a lui 2}$ vecin

A : 90.81.82.0/23 $\xrightarrow{\text{aj. la A prim}}$ C \Rightarrow agregare între A și B
B : 90.81.83.0/24 $\xrightarrow{\text{aj. la B tot prim}}$

- clase false (private), ex 192.168.1.0/24

(nonroutabile) = când se face schimb de tabele de dirijare, NU se mențină date moi de pe rutele către clase private (deoarece ele se pot "repeta" în mezi multe rute) (NU sunt localizate în mod unic)

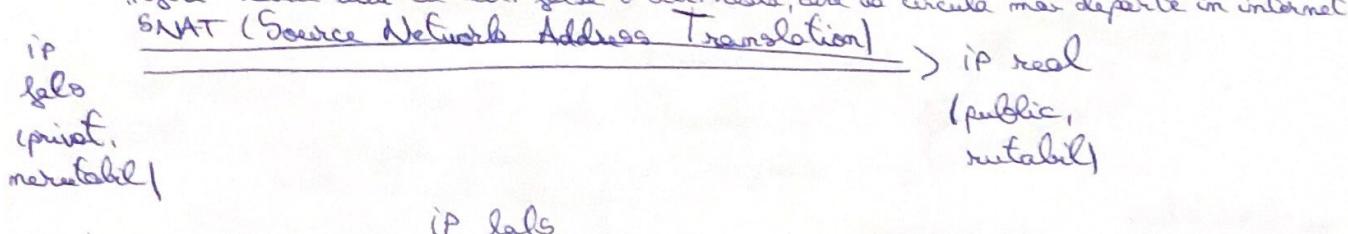
- SNAT

- clasă A de adr. false: 10.0.0.0/255.0.0.0 = 18

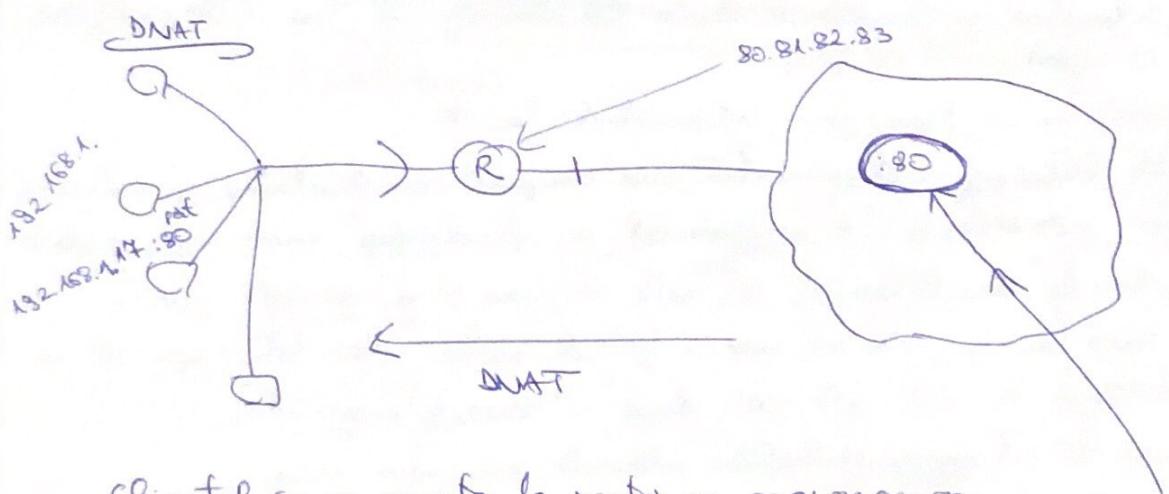
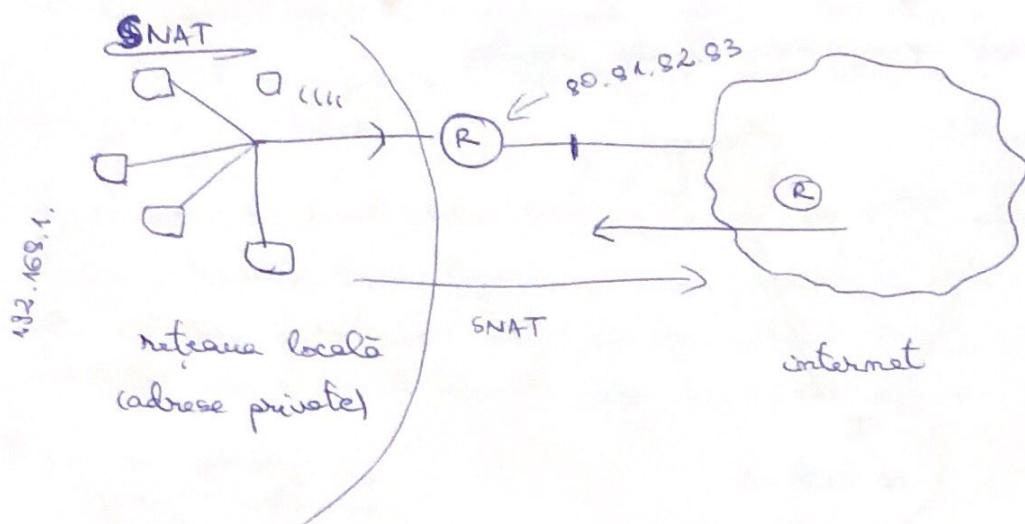
B de adr. false $\left\{ \begin{array}{l} 172.16.0.0/255.255.0.0 = 16 \\ \vdots \\ 172.31.0.0/16 \end{array} \right\} \xrightarrow{\text{agregare}} \Rightarrow 172.16.0.0/12$

C de adr. false $\left\{ \begin{array}{l} 192.168.0.0/255.255.255.0 = 127 \\ \vdots \\ 192.168.255.0/24 \end{array} \right\} \xrightarrow{} 192.168.0.0/16$

- tipuri de NAT (Network Address Translation) : folosint pt a "traduce" adr. IP false, care sunt neruabile. Fără neruabile, un device cu IP false poate trimite pachete în afara rețelei locale, însă NN va putea primi răspunsuri unic din internet. Fără NAT, se evită această problemă, deoarece rețeaua "făcea" device-ului cu adr. false o adr. reală, care va circula mai departe în internet.



avantaje	dezavantaje
<ul style="list-style-type: none">- se economisește adrese IP reale- securitate (NN sunt direct accesibile din internet, nu va fi „atacată” direct)	<ul style="list-style-type: none">- router-ul are o tabelă NAT, prin care va identifica cării adr. false împotriva adr. reale, pt. a asigura că pachetul primit ajunge la destinatarul său. Acest proces se bazează pe proprietățile pachetului primit.

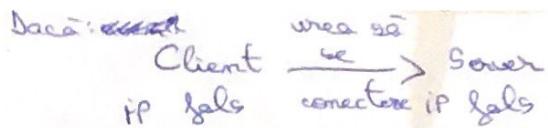


Cliențul se va conecta la router pe 80.81.82.83:80,
și trimite pachetul la router, care îl va redirecționa la
destinatariul cu IP privat 192.168.1.14:80 (care nu e unic).

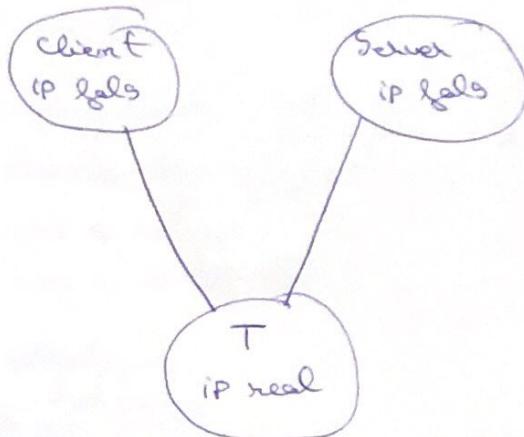
Cum? DNAT: Router-ul știe că tot ce se trim. din ^{in internet} client din exterior
exterior pe un anumit port (ex.: 80) se va
redirecționa către ^{un} ^{anumit} destinatar (ex: 192.168.1.14:80)

(aceea se trim. pachet
la 192.168.1.14:80
care nu e unică (privată))

! Portul din rețea locală NU e
obligaționă să fie același ca portul
din exterior



Nu e posibilă comunicarea directă. Soluție:



Clientul și Server-ul devin "clienti" al lui T (IP real), iar
 (IP fals) (IP fals)

comunicarea se va face prin intermediul lui T.

Default Gateway = echipamentul care asigură conectivitatea cu exteriorul

Serviciu DNS = server-ul responsabil cu "traducerea" nume calc. \rightarrow adr. IP

! Înainte de conectarea la un calc. la care îl se cunoscă numele,
 serverul DNS va fi "înțelept" căre e IP-ul aceluia calc. Abia apoi se va
 face conectarea la acel calc. (IP) (după ce serverul "răspunde").

Tipuri de interogări (întrebări adresate serverului DNS)

- tip A(Address) - Ce adresă IP are cunoscut server/calc?
- tip AAAA - Ce adresă IPv6 are cunoscut server/calc?
- interrogări inverse = PTR - Ce nume este asociat unei cunoscute adrese IP?
- CNAME (Character NAME) = alias-uri - un nume de calc. \rightarrow alt nume de calc
- MX (Mail exchanger) - se identifică serverele de mail yahoo.

Ex: Dacă scriem în browser "yahoo.com" \Rightarrow conectare la serverul yahoo $\xrightarrow{\text{HTTP}}$
 "user@yahoo.com"

• TXT (Text) -

- ! • NS (Name Server) - se poate localiza serverul DNS responsabil de un anumit domeniu (ex: albastur.ro), server-ul care resp. la interogările de mai sus pt. acel domeniu.

întrebare: De `www.ferrari.it` e rechizit de serverele DNS responsabile pt. el (bind, ping-ut), dar dacă scriem în browser nu facem?

Pt. că `www.ferrari.it` nu este un server web (nu se poate face conex pe portul :80 HTTP).

! Comunicare client - server DNS : prin UDP, pe portul :53.

! DNS = Domain Name System = sistem jerarhic de servere DNS conectate, între ele, care se întreabă între ele, dacă schimb de inf. pt. a răspunde clientilor care cere ip-ul asociat unei nume de calc.

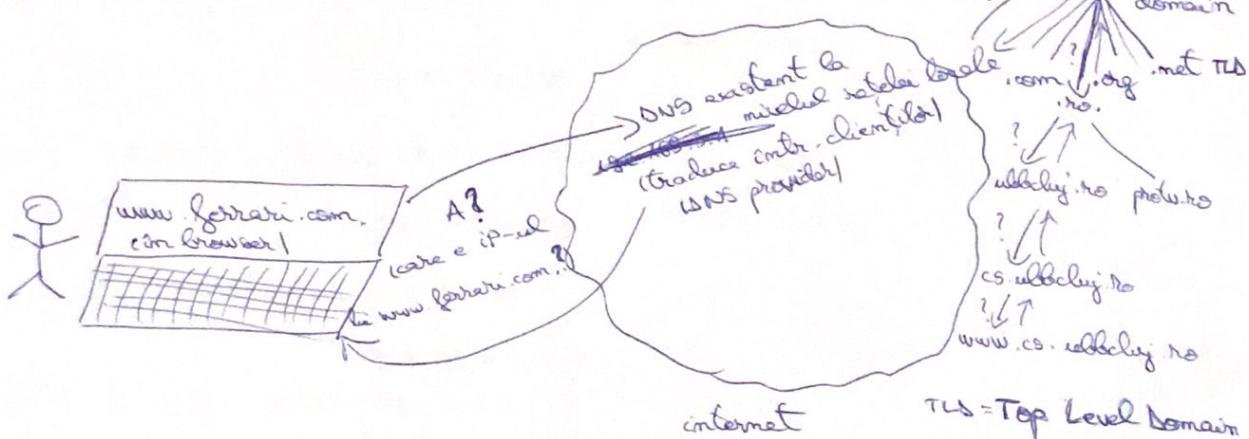
! `www.cs.ulbcluj.ro`.

`www` = hostname

`.cs` = domeniu

`.ulbcluj.ro` = domeniu (subdomeniu al domeniului `cs`)

• `ro` = _____
• = root domain



În root domain sunt localizate servere DNS care șiție ce servere DNS sunt responsabile pt. domeniul de pe urmat. Domeniile din arbore (urmat. nivel, ș.a.m.d. Astfel, întrebarea circulă de la serverul DNS local → serverele de root domain → → serverele DNS & TLD → ... → serverul DNS care ține ~~care~~ și ^{cum} `www.ferrari.com`. Cel din urmă răspunde, se transmite răsp. înapoi pînă la provider și apoi la calc. care a fostat în browser `www.ferrari.com`.

Întrebările au asociate un TTL (TimeToLive). În cache-ul local se regăsește prima întrebare de genul "`www.ferrari.com`", iar următoarele vor avea răsp. din cache (pt. a fi mai rapid, se evite repetarea întregului drum integral prin arborele DNS).

! Rețiri server DNS:

- preluare ? de la user și transmiterea ? mai departe în arbore + răspuns
- refinerea ? care are leg. cu un anumit domeniu din arbore (memorarea)

Interrogând un provider : repetitive (recursive)

 DNS e root : Iterative (aceste servicii să nu indică serverele de la nivelul următor, care să nu răspundă ? , nu răspund).