

## SOCKET-URI

- comunicare între procese de pe calculatoare diferite (se poate și din același calculator)
- model client - server
- pt. a comunica în rețea se folosesc 2 tipuri de protocole
  1. UDP (User Datagram Protocol)
    - ↳ permite transferul fără conexiune a informațiilor
  2. TCP (Transmission Control Protocol)
    - ↳ permite transferul prin conexiune a informațiilor, cu o de încredere
- se pot folosi la:
  - transfer de date
  - comunicare în timp real
  - descărcare de fișiere
  - chat
  - jocuri online

## TIPURI DE SOCKET-URI

### Socket stream

- serviciu orientat către conexiune
- date receptate în ordinea transmisiei
- protocol TCP
- analogie aparat telefonic

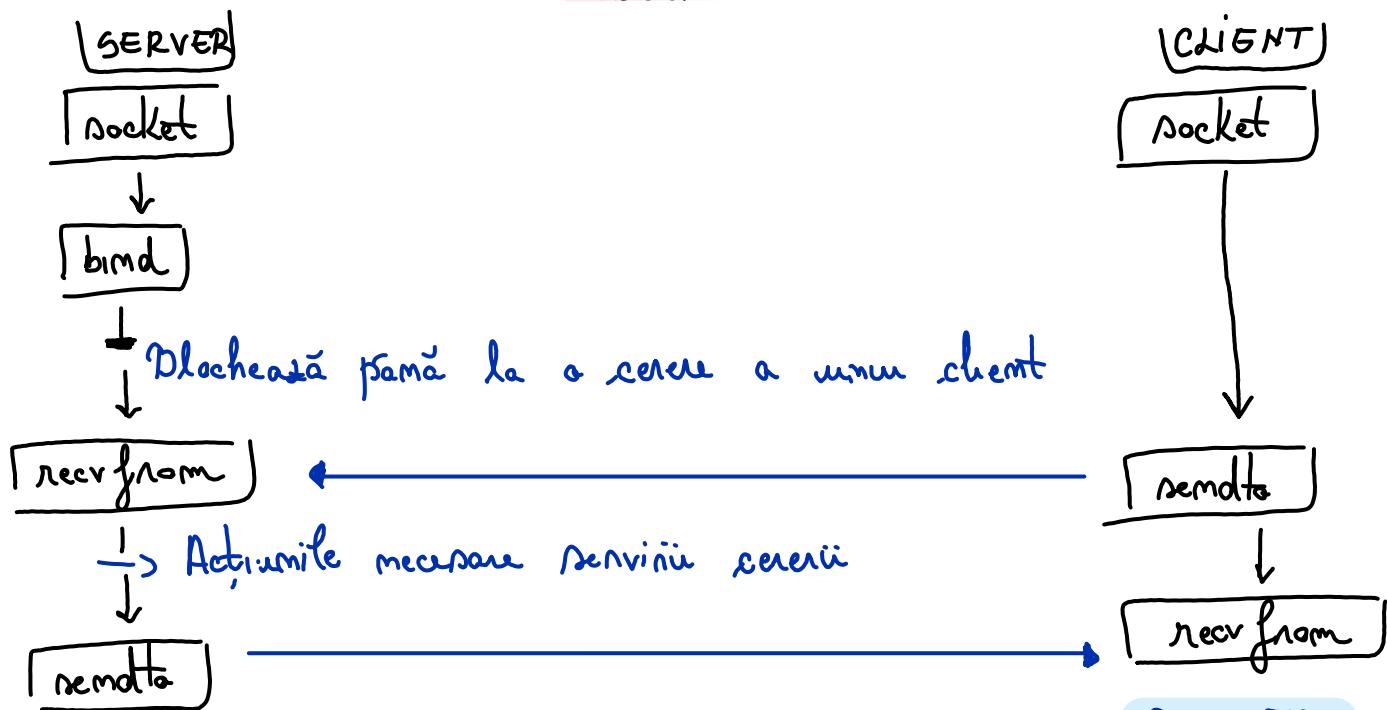
### Socket datagram

- serviciu fără conexiune
- nu garantează receptarea datelor
- datele pot ajunge în altă ordine decât cea în care au fost transm.
- protocolul UDP
- analogie: cutia poștală

Pentru a stabili o conexiune folosind socket-ură, fiecare dispozitiv sau proces are o adresa IP și un port asociat. Adresa IP identifică dispozitivul, în timp ce portul indică un serviciu sau o aplicație.

ex IP → clădirea  
port → camera

### SOCKET DATAGRAM Protocol UDP



ip = '0 0 0 0'

port = 8888

s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

s == -1 => fail

s.bind((ip, port))

mesaj = "salut"

s.sendto(mesaj, (ip, port))

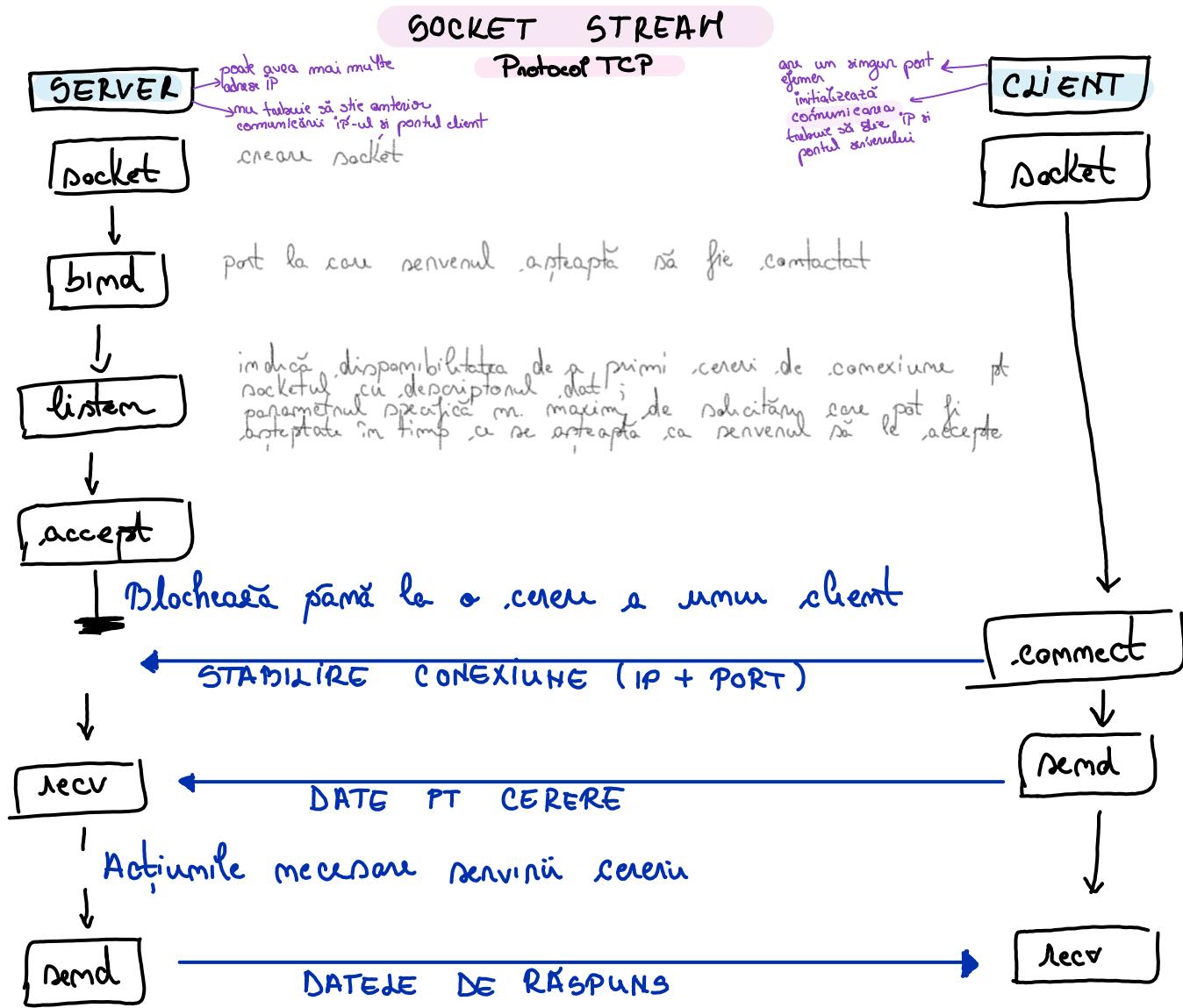
data, clientAdr = s.recvfrom(buffer)

datele primite de la client (socket) tipul <ip, port>

octeti

= zonă de mem. temporară utilizată pt memorarea și manipularea datelor

- discul, retea, disp hardware



### Apeluri sistem

`IP = '0 0 0 0'`

`port = 8888`

`s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)`

`s == -1` → fail

`s.bind((IP, port))`

`s.listen(n)`

`s.connect((IP, port))`

`clientSocket, clientAddr = s.accept()`

mai tip de socket care reprezintă conexiunea formată cu clientul

`AF_INET`

↳ Address Family Internet

`PF_INET`

↳ Protocol Family Internet

→ indică utilizarea adr.

IPV4 în comunicarea cu socket-uri

## Când eşuează socketul?

- 1 Portul sau IP-ul e deja folosit
- 2 Lipsa permisiunilor
  - nu sunteți permisiți să creați un socket
  - port < 1024 nu este folosit de preceșe de rutare
3. Rutarea prea multor conexiuni simultană
  - depășirea limitelor de resurse ale sistemului (limitele de descriptori de fizici)
- 4 Setările incorecte ale firewall-ului
  - se blochează conexiunea
- 5 Probleme de rețea
  - server indisponibil
- 6 Deactivarea protocolului necesar
  - ex: Încercarea utilizării unui socket IPv6 pe un port IPv4
- 7 Depășirea limitelor de resurse ale sistemului
- 8 Erori de memorie sau depășirea bufferului

## MEDIU DE TRANSMISIE

### 1. Fir de cupru

- cablu coaxial
- cablu UTP (Unshielded Twisted Pair)

### 2. Fibra optică

- Single mode - distanțe lungi și viteză mare
  - un singur mod de propagare
- Multimode - distanțe mai scurte și mai ușoare
  - mai multă moduri de propagare

### 3. Wireless Media

- unde radio
- infraroșu (lumini de vedere directă)

### 4. Satellite Communication

- sateliți geostacionari
  - înălțime fixă deasupra Pământului
  - comunicare la nivel global (ex. tv)
- sateliți non-geostacionari
  - se află pe orbită
  - comunicare mobilă și internet

## Cabluri

Cablu direct - dispozitive diferențiate

- ,alb portocaliu
- portocaliu
- alb verde
- albăstru
- alb ,albăstru
- verde
- alb maro
- maro

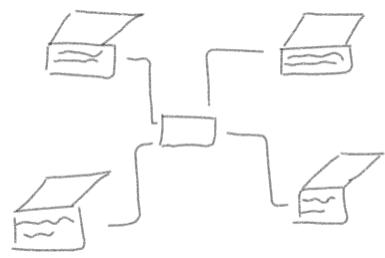
Cablu cross-over (se schimbă portocaliu cu verde) - dispozitive similare

- ,alb verde
- verde
- alb portocaliu
- albăstru
- alb ,albăstru
- portocaliu
- alb maro
- maro

## TOPOLOGII

### 1 Star topology

- ↳ toate dispozitivele sunt conectate punct-um mod central (comutator / hub)
- ↳ dispozitivele sunt conectate în mod ind. prin modul central



### 2 Extended star topology

- ↳ se folosește un dispozitiv central care e conectat la mai multe alte dispozitive



### \* 3 Bus Topology

- ↳ toate dispozitivele sunt conectate la un cablu comun
- ↳ datele de la un dispozitiv la altul sunt transmise prin magistrală și fiecare stație decide dacă să proceseze sau să ignore datele destinate altor stații



### 4 Ring Topology

- ↳ datele circulă de la o stație la alta pînă ajunge la determinatie



## NETWORK

- = conexiune de dispozitive sau sisteme interconectate care permit schimburi de informații, date sau resurse
- poate fi  **fizică** (componente hardware reale, calculatoare, switch-ură, routere, cabluri de rețea, sisteme Wi-Fi, imprimante, hub-ură)
- poate fi  **logică** (modul în care dispozitivele comunică, deosebit de important cum se conectează: IP, subretele, tabele de routare, regulile de firewall, protocoale de comunicare)
- cuprinde: calculatoare, telefoane, servere, imprimante, routere

## TIPURI DE REȚELE

### 1. LAN (Local Area Network)

- rețea locală, neînțără dintr-o casă / birou / clădire / campus
- dispozitivele din același locație se pot conecta și să partajeze informații
- adresa sunt gestionați de un router sau un switch pe o linie tronșcul de date între dispozitive

### 2. WAN (Wide Area Network)

- acoperă o zonă mult mai extinsă (oraz / ţară / planetă)
- mai lent decât LAN
- se bazează pe infrastructura publică de telecomunicatii: linii telefomice, cabluri de fibră optică, sateliți sau conexiuni de internet
- internetul în sine este un WAN

### 3. MAN (Metropolitam Area Network)

- **LAN < MAN < WAN** (zonă metropolitana, oraș, oraș mare)
- scop de a conecta dispozitive din diverse locații sau clădiri din același oraș
- viteză de transfer: LAN < MAN < WAN

### 4. PAN (Personal Area Network)

### 5. WLAN (Wireless Local Area Network)

## PROTOCOALE DE COMUNICARE

### 1. TCP/IP (Transmission Control Protocol / Internet Protocol)

- asigură transmiterea fiabilă a datelor și controlul fluxului
- rapid

### 2. UDP (User Datagram Protocol)

- mai simplu, mai puțin fiabil
- livrare neasigurată
- viteză redusă

WWW

(World wide Web)

= rețea globală de informații  
accesare și partajare  
prin internet

### 3. HTTP (Hypertext Transport Protocol)

- transfer de pagini web și resurse asociate pe internet

### 4. HTTPS (Hypertext Transport Protocol Secure)

- doar dacă datele sunt criptate

HTTP și HTTPS TCP

↓  
port 80

↓  
port 443

### 5. FTP (File Transfer Protocol)

- transfer de fișiere între calculatoare

### 6. SMTP (Simple Mail Transfer Protocol)

- transmisie și receptare de mailuri

### 7. POP3 (Post Office Protocol, vrs. 3) și IMAP (Internet Mail Access Prot.)

- folosita pentru accesarea și stocarea mailurilor
- POP3. descarcă mail-ul pe dispozitivul curent
- IMAP. le păstrează pe server și permite acceseul de pe mai multe dispozitive

### 8. DNS (Domain Name System)

- asociază numele de domeniu (ex: www exemplu.com) cu IP-ul

### 9. DHCP (Dynamic Host Configuration Protocol)

- atrage automat adresa ip și alte informații de configurație a rețelei dispozitivelor care se conectează la o rețea)

## 10. **SNMP** (Simple Network Management Protocol)

- monitorizarea și gestionarea dispozitivelor de rețea (ruter, switch-uri, servere)
- permite administratorului să monitorizeze starea dispozitivelor și să facă modificări

## 11. **SSH** (Secure Shell)

- conexiune securizată și criptată între 2 dispozitive

## Poșta electronică (e-mail)

- = serviciu de trimis și primire mesaje prin intermediul internetului
  - ↳ ex. Google, Yahoo, Microsoft etc
- utilizatorul primește și adrese de e-mail sănătoase (poate trimite mail unicun ic în lume, independent de serviciu)
- se utilizează **SMTP** (Simple Mail Transfer Protocol)
  - ↳ utilizarea **TCP**
- beneficiu viteză
  - urgența de utilizare
  - cost redus
- dezavantaje spam
  - phishing
  - riscul de a avea emailul compromis

## SPF (Sender Policy Framework)

- tehnica de autentificare pentru e-mail care permite receptorului să verifice autoritatea serverului de poștă electronică
- permite proprietarilor de domeniu să specifice serverele de p.e. care sunt autorizate să trimită mesaje de e-mail în numele domeniului lor
- poate reduce spamul și atașurile
- receiverul server trimite DNS query-uri pentru a face verificări, dar e limitat la 10
  - ⇒ > 10 query-uri → ↳ există "extensiuni" pentru o mai bună funcționare

## Protocol

- set de reguli pe care trebuie să le respecte 2 parteneri care comunică
- Există **RFC** (Request For Comments). descriu de ce trebuie să facă comunitatea implementator, de client și sunul de server

## Tipuri de trafic

- Uunicast**
- comunicare 1:1 emițător - receptor (deobicei din același LAN)
  - conexiune TCP clasica

- Broadcast** - metodă de transmitere a informațiilor către toate disponibilele dintr-un LAN
- comunicare 1:toti
  - servicii ARP, RARP, DHCP (prin UDP)
    - ↳ nu se poate prim TCP (ar fi mai mult pe același descriptor de socket)

- Multicast** - comunicare 1:m (nu primește totuști meșterat)
- ex. 2 pere dintr-un LAN între pe un rețeaua  $\rightarrow$  2x trafic unicat între device în rețea
  - ABCD
    - ↳ A = maxim 223
    - ↳  $\rightarrow$  ABCD validă

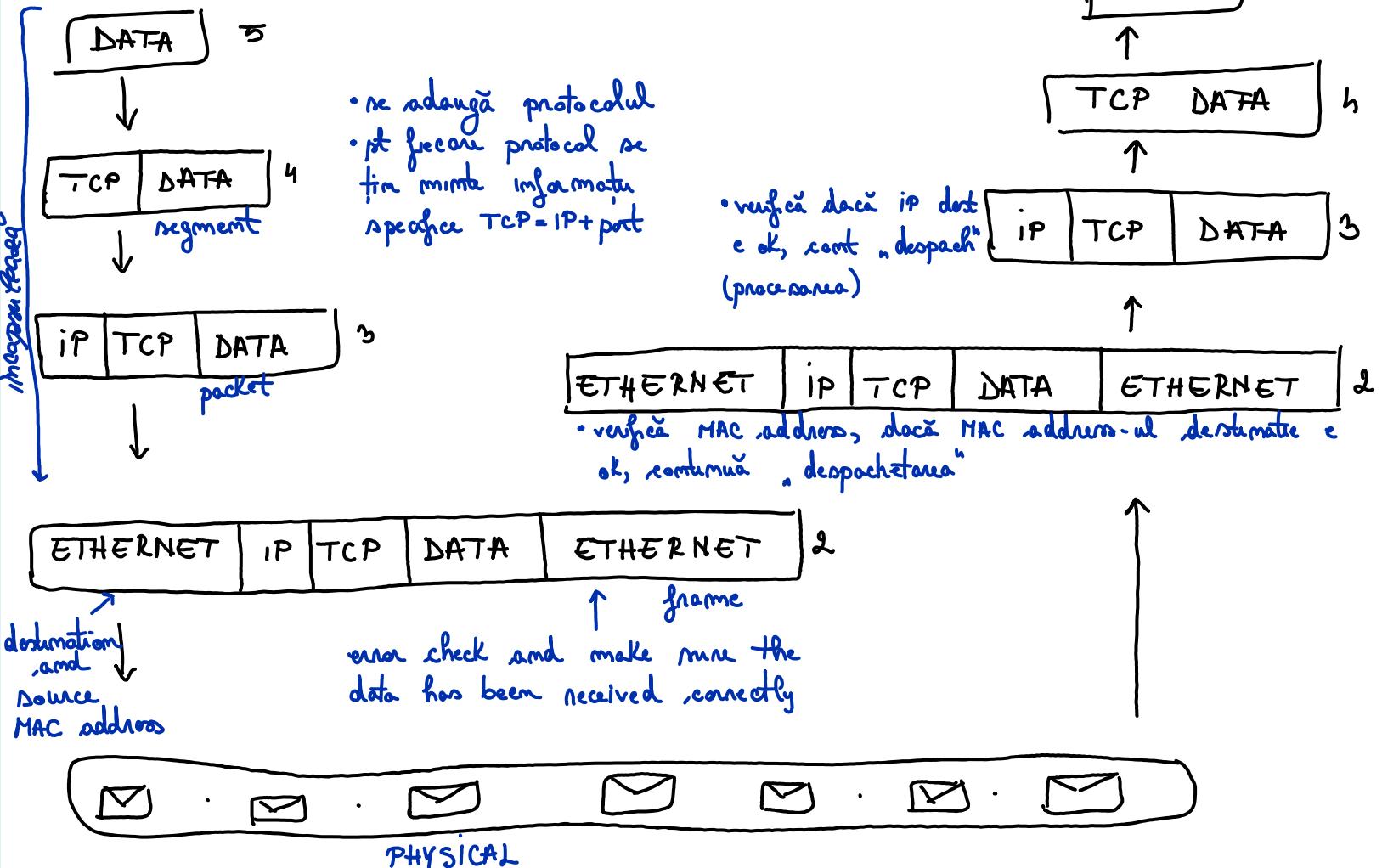
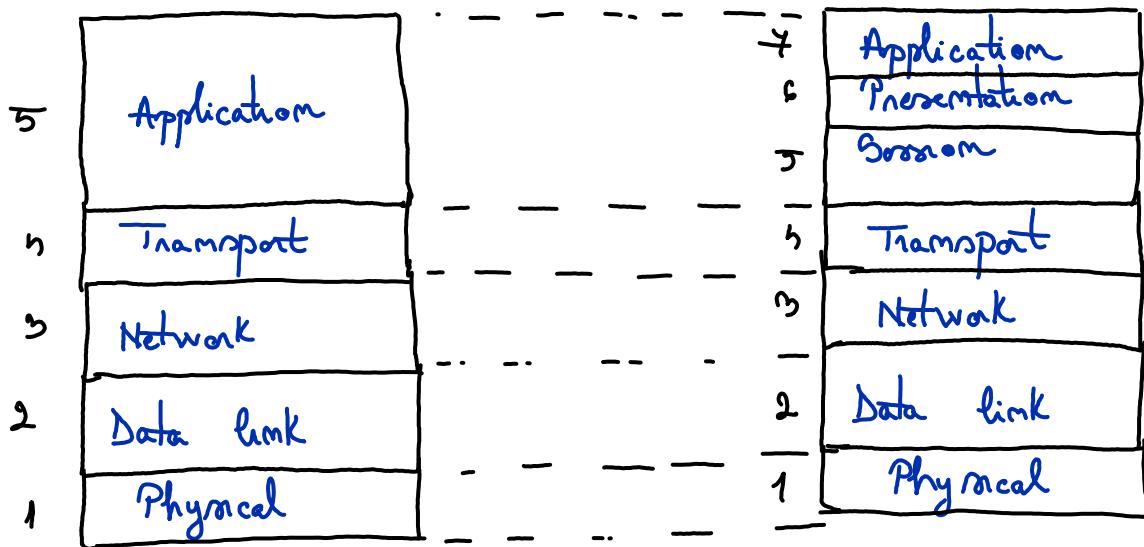
- Anycast** - comunicare 1:cel puțin 1 din mai multe
  - ex. într-un LAN sunt mai multe servise DHCP. e important ca cel puțin 1 să răsp. cererii unui client

FF FF FF FF FF FF

= MAC de broadcast

- utilizat pentru comunicarea cu toate echipamentele dintr-un LAN

## TCP / IP model



## OSI model

OSI (Open System Interconnection) este un cadru conceptual utilizat pentru a descrie în întregime funcționalitatea rețelelor și nu e folosit în prezent.

<p>Acesta e cel mai apropiat strat de utilizator. În loc interacțiunile cu aplicația: navigare web, clientul de e-mail, server web, etc.</p>	<p>SMTP, FTP, Telnet</p>	<p>Application (7)</p>
<p>Se ocupă cu formatarea datelor și le face compatibile cu dispozitivele în aplicările din rețea. Compresie, criptare și alte operații de prelucrare a datelor.</p>	<p>Format Data, Encryption</p>	<p>Presentation (6)</p>
<p>Stabilire, menținere și închidere semnificative de comunicare între dispozitive</p>	<p>Start &amp; Stop Session</p>	<p>Session (5)</p>
<p>Asigură comunicarea fizică și controlul flexibil între dispozitive (protocole)</p>	<p>TCP, UDP, Port Numbers</p>	<p>Transport (4)</p>
<p>Routarea datelor între rețele dif. Utilizarea adrese IP și a unui pește de către destinația corectă.</p>	<p>IP Address, Routers</p>	<p>Network (3)</p>
<p>Transmiterea datelor între dispozitive conectate la același LAN. Se add. MAC addresses și se efectuează verificarea de erori și asigură transmiterea fizică.</p>	<p>Mac Addresses, Switches</p>	<p>Data link (2)</p>
<p>Componente fizice ale rețelei (cabluri, conectori, seminoduri electrice sau optice). Se ocupă cu transmiterea bruto a datelor.</p>	<p>Cable, Network Interface Cards, Hubs</p>	<p>Physical (1)</p>



1	Application
2	Presentation
3	Session
4	Transport
5	Network
6	Data link
7	Physical

Prin incapsulare, pachetul este "îmvelit" de primul dispozitiv în traius spre al doilea, care prelucrează pachetul în ordine inversă (pe rând fiecare nivel)

4	Application
5	Presentation
6	Session
7	Transport
1	Network
2	Data link
3	Physical

## Port numbers

adresa URL → adresa IP → server

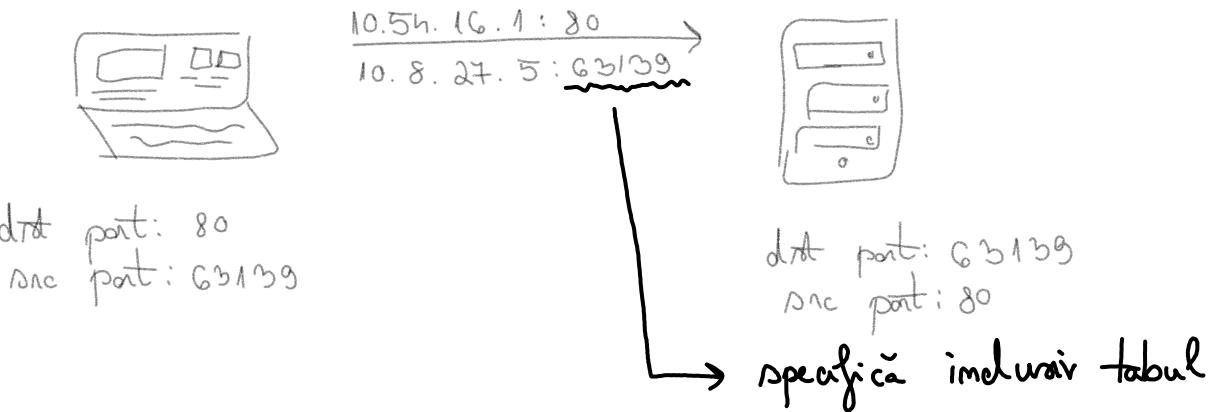
se adaugă portul  
specific în funcție  
de aplicație

ex: 10.54.16.1:80

dst port: 80 (HTTP)

src port: 65139 (generat random)

Adresa IP date către computer  
Pat date către aplicație



## Porturi

0 - 1023 → well known

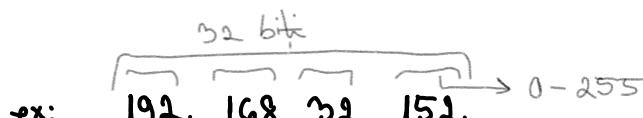
1024 - 49151 → registered

49152 - 65535 → dynamically assigned

## Adrese IP

- identificator unic assignat fiecărui dispozitiv conectat la o rețea de calc

IP v4



(țarada)

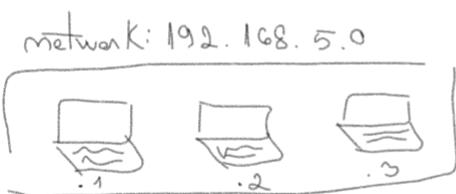
(m. carei)

- împărțită în 2 prima parte NETWORK, și două părți: HOST

Subnet mask 255.255.255.0

validă doar pentru  
NM /24

ex:



192.168.5.0 → host  
255.255.255.0

**CLOSE**      **PUBLIC**

A      1000 - 126.255.255.255  
subnet 255.0.0.0 → hosts: 16777216 hosts

**PRIVATE**

10000 - 10.255.255.255

B      128000 - 191.255.255.255  
subnet 255.255.0.0 → hosts: 65536

142.1600 - 142.31.255.255

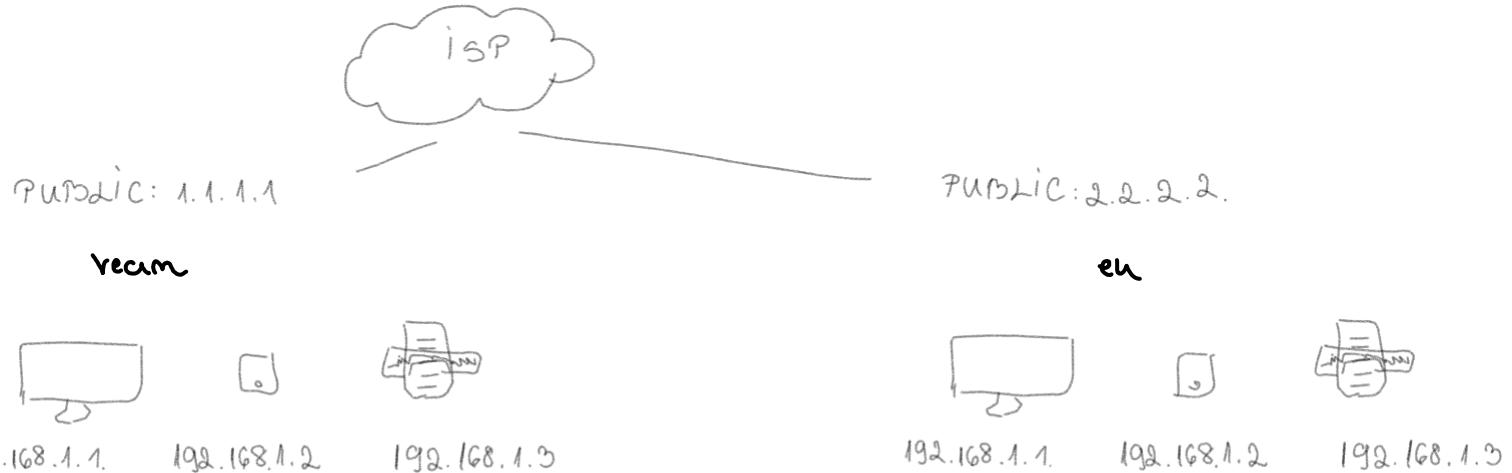
C      192.000 - 223.255.255.255  
subnet 255.255.255.0 → hosts: 256

192.16800 - 192.168.225.225

D - multicast addresses

E - experimental use

Public - doar prin internet, trebuie să fie sunrice



### Adrese IP private (false)

- trebuie să fie urmice doar în propria rețea  
( $\hookrightarrow$  nu și în vecinul patern arec același adresa IP)
- nu pot fi folosite prin internet (s-ar face un ghicire de dupăcat)
- In general, la crearea contractului pentru internet primești o adresă IP
- avantaje permit economia de clase de addrs. ip reale  
securitatea
- dezavantaje: trebuie SNAT ca să meangă internetul  
nu se pot ruia servicii pe care să fie accesibile din alte  
părți din internet fără DNAT
- nu sunt mutabile

NAT nu înlocuiește adrese  
IP false cu urmăre reale  
(se poate fals cu fals / real  
cu real)

## Bimany

ex: 192 168 0.1

1 - em  
0 - of

10000000 10101000 00000000 0000 0001

128 | 63 | 52 | 16 | 8 | 6 | 2 | 1  
0 | 0 | 0 | 0 | 0 | 0 | 0 | 0

1

$$192 - 128 = 64$$

128	65	52	16	8	5	2	1
0	0	0	0	0	0	0	0

$$168 - 128 = 40$$

$$A - A = 0$$

$$h0 - 32 = 8$$

$$8 - 8 = 0$$

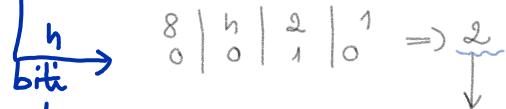
Punem 1 în 0 astfel  
încât , atunci , cînd  
adunăm valoarele din  
căsuinile la care am  
pus 1 , să ne dea  
mr. dorit. Începem  
de la ~~7~~<sup>7</sup> și 1 , cu cea  
mai mare valoare.

## IPv6 Addresses

IPv4 - 32 bits (4 octets)

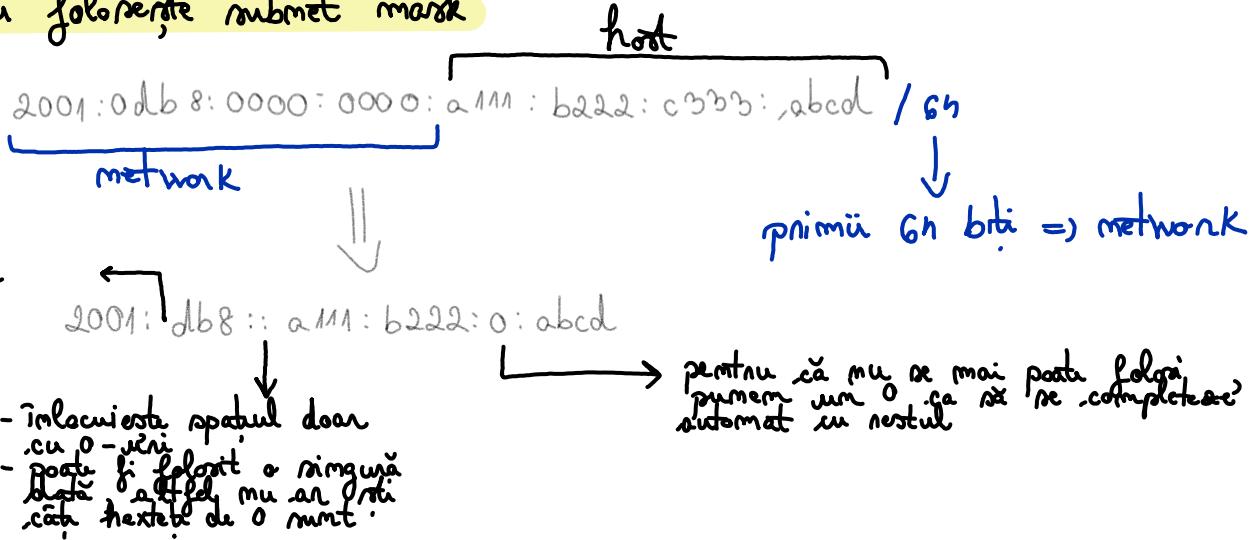
IPv6 - 128 bits (8 hexadeci)

- format din caracter (cifre 0-9 sau litere A-F)



2001:0db8:0000:0000:a111:b222:c333:abcd

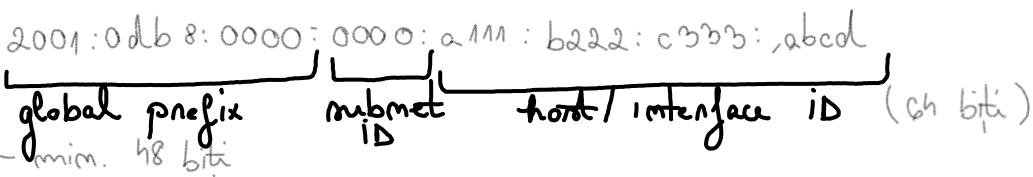
- nu se mai folosesc subnet mask



### Tipuri

#### 1. Global Unicast

- ca în una publică v4 (pentru că sunt sute mii de rețele, nu mai avem nevoie de adrese private)



2000::/3 Publicly routable (începe cu 2 sau cu 3)

#### 2. Unique Local

- ca în una privată v4

F000::/7 Routable in the LAN (începe cu F, urmat de c sau D)

A	10
B	11
C	12
D	13
E	14
F	15

### 3. Link Local

- comunică între nicio rețea a unor rețele
  - 169.254.x.x. Cand dispozitivul nu se poate conecta
- FE80::1/10 Not routable (începe cu FE)

### 4. Multicast

- se trimite unui grup de dispozitive care așteaptă în mod special același adresa (broadcast)

FF00::1/8 Addresses for groups (începe cu FF)

### 5. Anycast

- angajarea unei adrese IP mai multor dispozitive
- informațiile sunt trimise celui mai apropiat dispozitiv cu adresa respectivă

2000::1/5

## MAC Addresses (Media Access Control)

- identificator unic assignat unei interfețe de rețea (NIC) (Network Interface Card)
- adrese fixe
- nu pot fi schimbate → x pot schimba dar nu e recomandat
- coduri (48 biti)

### 5. APPLICATION

#### 4. TRANSPORT

#### 3. NETWORK

#### 2. DATA LINK

#### 1. PHYSICAL

→ tehnologie pentru conectarea dispozitivelor la LAN

veloare unică atribuită de vendor

08 - 00 - 27 - EC - 10 - 61

OUI (Vendor)  
(organizationally unique  
identification)

Vendor

= codul de identificare  
al furnizorului

- compania / producătorul  
dispozitivului

## Tipuri

### 1 UNICAST

↳ particulară, unică (ex. de răs)

### 2. MULTICAST

01 - 00 - 5E - 00 - 00 - 05

multicast prefix

- pentru aplicație / protocol  
- se trimite tuturor dispozitivelor,elor care sunt conectate la același lan / rețea

### 3 BROADCAST

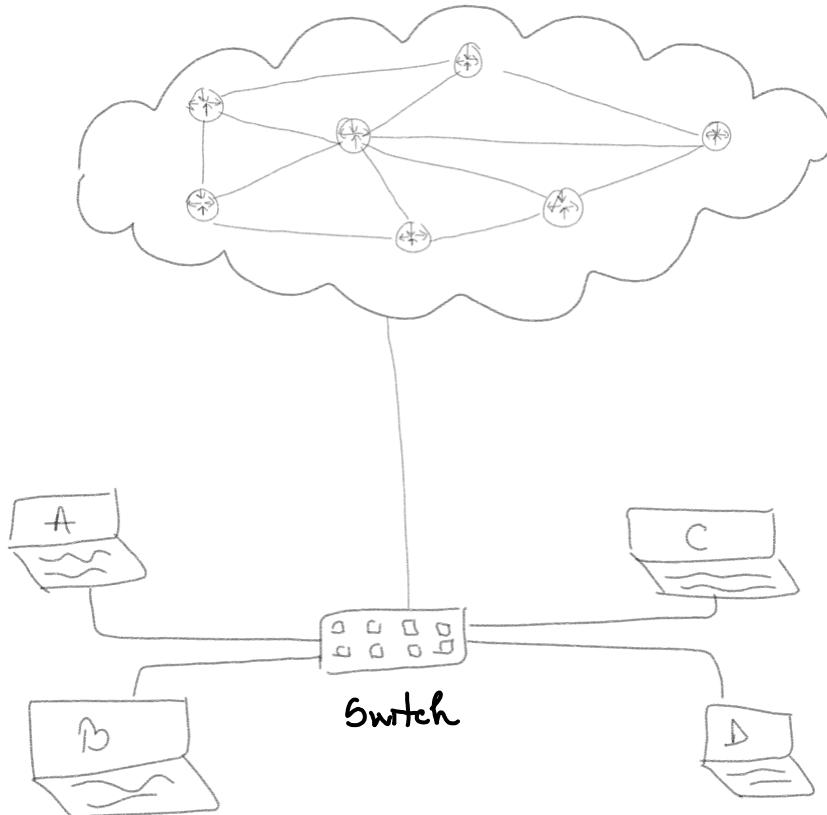
FF - FF - FF - FF - FF - FF

↳ se trimite tuturor dispozitivelor dintr-o rețea

## Moduri de rețea

dimux / Apple 08 00 27: EC: 10 . G1  
 Microsoft . 08 - 00 - 27 - EC - 10 - G1  
 Cisco 08 00 27 EC 10 G1

Router

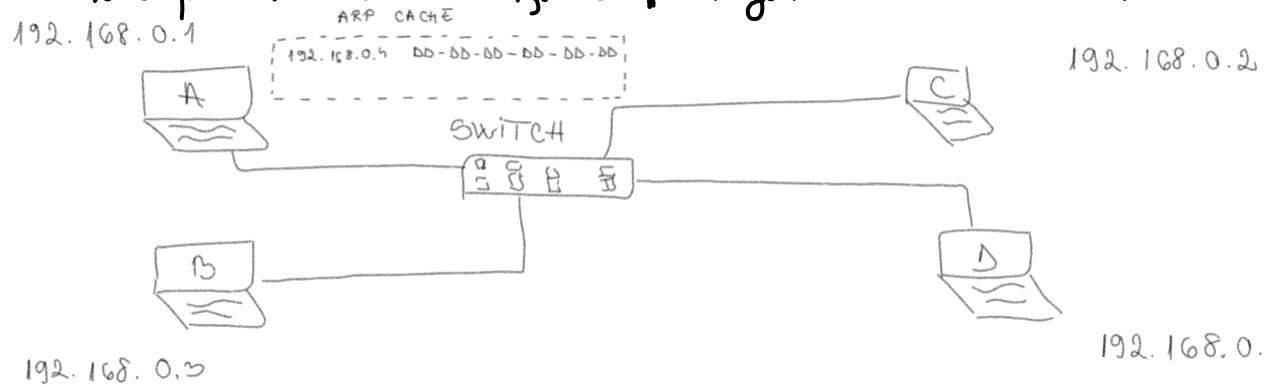


layer 3  
 IP Addresses  
 → global communication

layer 2  
 MAC Addresses  
 → local communication

## ARP Address Resolution Protocol

→ descoperă adresa MAC și le „transformă” în adresa IP



A vrea să comunice cu D

Trimit un mesaj broadcast să vadă unde e IP 192.168.0.4

B și C nu au înțelesă

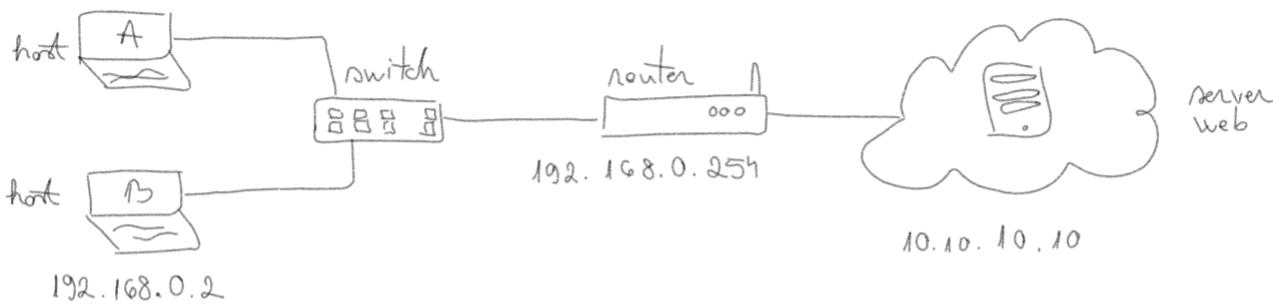
D îl trimite înapoi lui A adresa MAC

Se va salva în ARP cache pentru viitoare utilizări

### SWITCH

- layer 2
- doar adr. MAC

192.168.0.1  
D/G: 192.168.0.254



A vrea să comunice cu serverul web

A verifică IP-ul D.G - ului și vede că nu face parte din aceeași rețea

A trimite cerere ARP în rețea: B negrescă, routerul îl răspunde cu adresa MAC (ca A să poată ieși din rețea)

A acum A poate să trimită date cu MAC adresa-ului routerului, dar IP-ul serverului (mai departe se ocupă routerul).

Cererea ARP  
se realizează  
doar în  
layer 2 !

Default gateway = „ună” spie întreținută din rețea (în funcție de IP)

ex: În acest caz, e routerul

### RARP

( Reverse Address Resolution Protocol )

- protocol utilizat pt. atribuirea adreselor IP dispositivelor care nu pot stoca propriile adrese IP
- mod de functionare: dispozitivul trimite adresa MAC si solicită una IP
  - un server RARP răspunde cu adresa cerută
- ⇒ află adresa IP pe baza adresei MAC
  - ↳ doar pe rețea o sumoară
- se face înlocuit de DHCP

### Broadcast

- metodă de transmitere a unui pachet de date tuturor dispozitelor dintr-o rețea
- totă bitul gazdăi număr 1

## VLAN

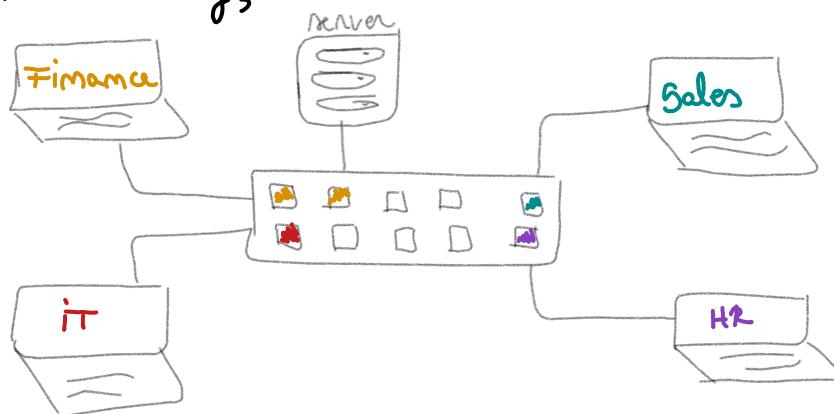
## Virtual Local Area Network

- separă rețea LAN-uri

De ce să folosim VLAN?

### Broadcast traffic

- se reprodă într-o rețea
- traficul de date se comportă ca și cum ar fi împărțit fizic (adăugarea unor switch-uri / routare)
- se atribuie interfețe

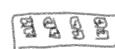


În realitate nu folosesc numere, nu culori!

- se poate comunica doar în „interiorul” acelorași VLAN (finanțe și server pot comunica, etc. că fac parte din același VLAN)

### Mod de funcționare

- VLAN initial VLAN1



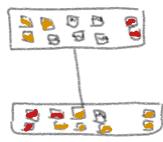
↳ toate interfețele pot comunica între ele

- se pot adăuga maxim 4094 de VLAN-uri



- VLAN 1 (df)
- VLAN 10
- VLAN 20

- putem să avem același VLAN-uri între mai multe switch-uri



→ necesită un trunk → tip special de interfață

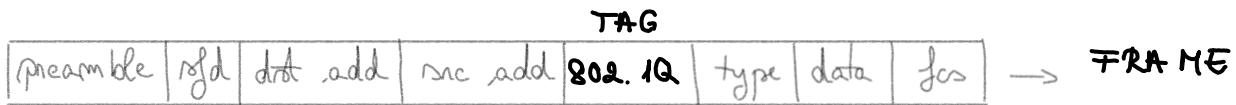
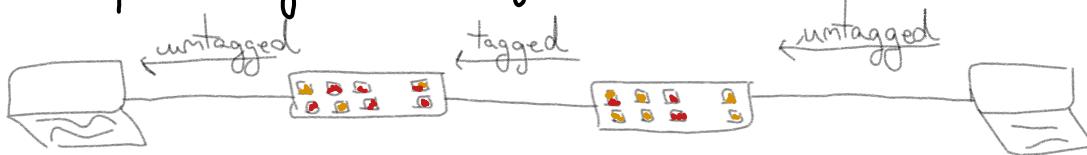
pentru a

tag

! astăzi switch

## Tag

- majoritatea dispozitivelor nu „stă” ce e un VLAN
  - ↳ gestiunea de switch
- ⇒ computerele gestionă frame-uri

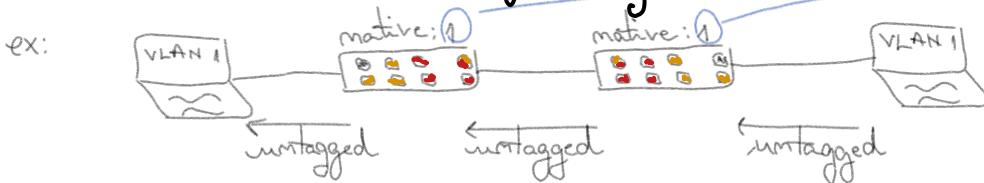


- 4 octeți · TPID (tag protocol identifier)
  - ↳ permănuște să identifice frame-ul ca fiind 802.1q tagged
- TCI (tag control information)
  - ↳ 3 biti · 1 priorităție
  - 2. DEI (drop eligible indicator)
  - 3. id-ul VLAN-ului

## Native VLANs

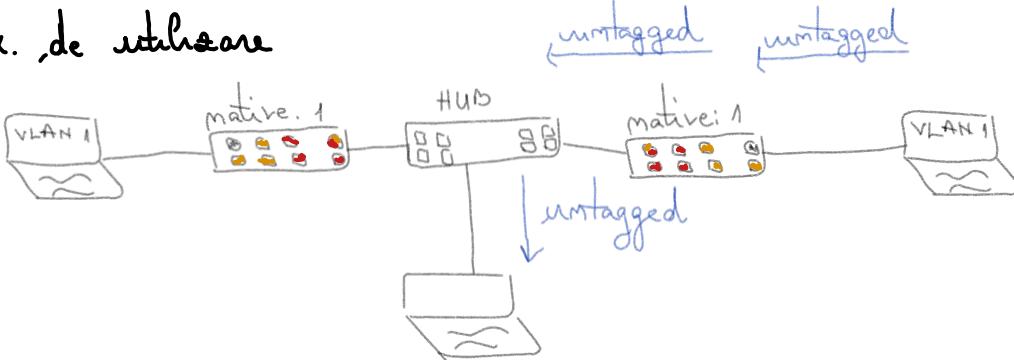
- trunk interface

- traversarea unui trunk fără tag



nu pot fi definite,  
informația nu-are mai  
ajunge la destinație

- ex. de utilizare



### HUB

- nu pot scrie / citi tag-uri
- doar transmit frame-uri

## STP Spanning Tree Protocol

### Tipuri de STP

- STP / 802.1D - original
- PVST+ - imbunătățire Cisco a STP prin adăugarea VLAN
- RSTP / 802.1w - imbunătățire STP cu o convergență mult mai rapidă
- Rapid PVST+ - imbunătățire Cisco a RSTP prin adăugarea VLAN

### Utilizare

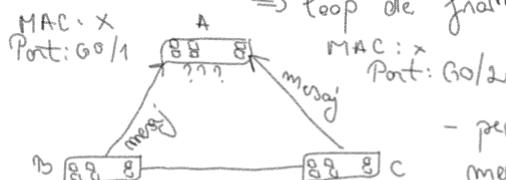
- prevene loop-urile, când sunt utilizate 2 sau mai multe switchuri

→ broadcast storm



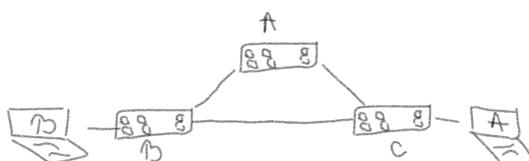
- de fiecare dată când se trim. un mesaj broadcast, se trimit către toate switch-urile  
⇒ loop de frame-uri

→ unitable MAC Address Tables



- pentru același mesaj de la switch dif., se vor actualiza mac-tablele

→ duplicate frames



Host B → host A

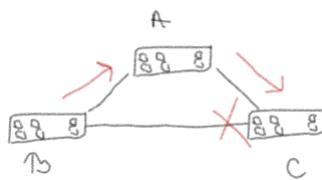
sw. B nu știe adresa lui h A

sw. C știe locația lui h A și îi trimite

sw. A știe că nu e pețnul el, deci trimite mai departe ⇒ ajunge la sw. C care îi trim. iar lui h A

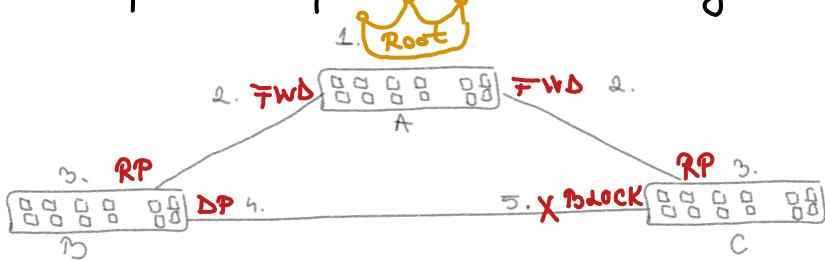
(⇒ trimite la toată lumea)

- mod de prevenție: pentru a nu se creea loopuri, sw. care trimite informația va bloca anumite porturi



## Mod de functionare

1. Elect a Root Bridge
2. Place root interfaces into a Forwarding state
3. Each non-root selects its Root port
4. Remaining links choose a Designated Port
5. All other ports are put into a Blocking state



## Roles

Root Ports - the best port to reach the Root Bridge

Designated Ports - port with the best route to the Root Bridge on a link

Non-Designated Ports - all other ports that are in a blocking state

## States

Disabled - a port that is shutdown

Blocking - a port that is blocking traffic

Listening - not forwarding traffic and not learning MAC addresses

Learning - not forwarding traffic but learning MAC addresses

Forwarding - sending and receiving traffic like normal

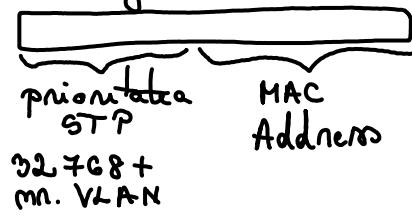
## STP

### 1. Root Bridge Election

- forward switch are use BPDUs

→ root cost, route in local RIB

→ Rbridge ID



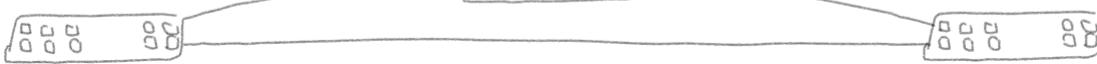
- devine root bridge switch-ul cu cel mai mic RIBD pe total

- la început, fiecare zw. se consideră rădăcina

Root Cost: 0  
My BID: 32769 bbbb: bbbb: bbbb  
Root BID: 32769 bbbb: bbbb: bbbb

Root Cost: 0  
My BID: 32769 aaaa: aaaa: aaaa  
Root BID: 32769 aaaa: aaaa: aaaa

Root Cost: 0  
My BID: 32769 cccc: cccc: cccc  
Root BID: 32769 cccc: cccc: cccc

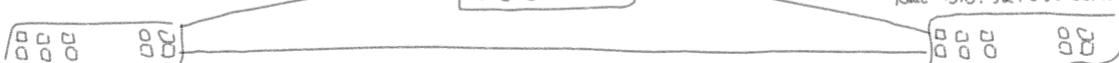


- apoi își trimit BPDU-urile între ele, iar cele care au BID mai mare se "conformează"

Root Cost: 0  
My BID: 32769 bbbb: bbbb: bbbb  
Root BID: 32769 bbbb: bbbb: bbbb

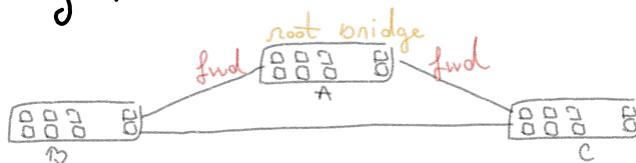
Root Cost: 0  
My BID: 32769 aaaa: aaaa: aaaa  
Root BID: 32769 aaaa: aaaa: aaaa

Root Cost: 0  
My BID: 32769 cccc: cccc: cccc  
Root BID: 32769 cccc: cccc: cccc



## 2. Root interface forwarding state

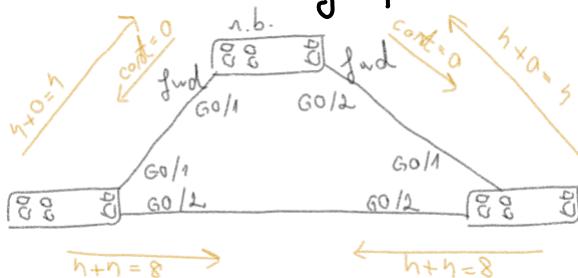
- toate porturile care se află în legătură directă cu rădăcina sunt în forwarding state



## 3 Non-roots choose the best path to the root bridge (reports)

- se bazează pe cantitatea porturilor,

nr. porturilor care merg spre rădăcina

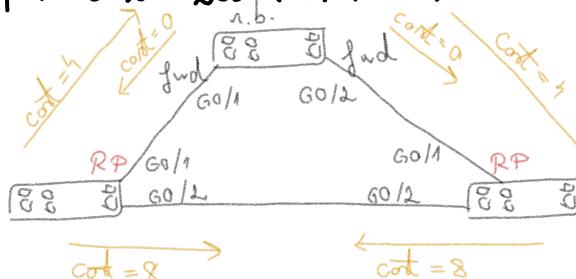


### PORT COST

Port Speed	Original	New
10 Mbps	100	2 000 000
100 Mbps	10	200 000
1 Gbps	4	20 000
10 Gbps	2	2000
100 Gbps	N/A	200
1 Tbps	N/A	20

- se alege portul cu cel mai mic cost

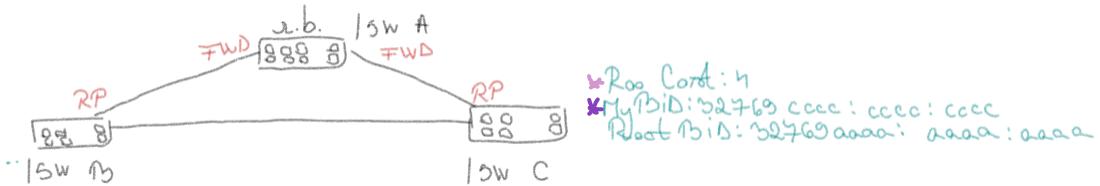
=>



- dacă nu fi fost același port pe mai multe porturi, atunci se alege vecinul cu cel mai mic BiD
- se verifică cea mai mică prioritate a portului
- în caz de egalitate, se verifică cel mai mic nr. de port

#### 4. Designated Ports

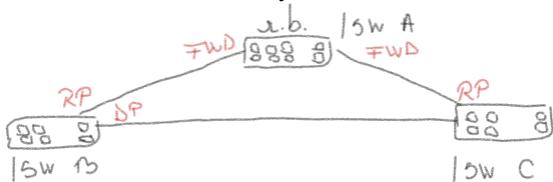
- se alege dintre cele care nu sunt repezzi



- pară (se trece la un numătorul în caz de egalitate).

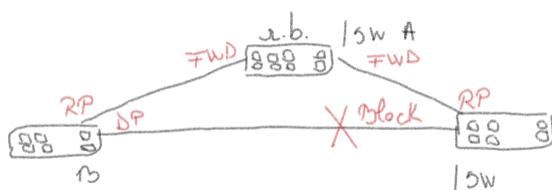
- se verifică cel mai mic port \*
- se verifică cel mai mic BiD \*
- se verifica cel mai mic neighbor port priority
- se verifica cel mai mic neighbor port number

- în ex. de mai sus, port B devine demgated port



#### 5. Blocking

- fiecare port care nu e rp (root port) sau dp (demgated port) este pus în blocking state



## Timers - Default

Hello	2 sec	→ intervalul de timp în care RBC crează și trimite hello messages (arașă rătie totă „lumea” că lucrările încă funcționează)
Max Age	$10 \times \text{Hello}$ (20 sec.)	→ atât timp, arăgădește switchul pînă reactualizează că ceea ce vă nu e ok
Forward delay	15 sec	→ timp între linklayer state și learning state

## STP states (condiția efectuării)

Forwarding state →  $\xrightarrow{\text{sec}}$  Blocking state →  $\xrightarrow{\text{sec}}$  Listening state  $\xrightarrow{\frac{15}{\text{sec}}}$  Learning state  $\xrightarrow{\frac{15}{\text{sec}}}$

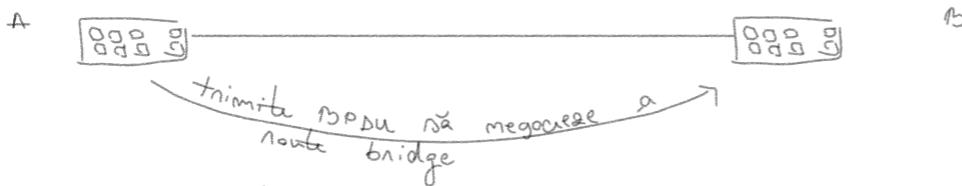
↳ durată multă → rezolvare rapidă splicing tree protocol

↳ în realitate, dacă avem un port conectat la un switch, portul va fi portocaliu către timp și abia apoi se face verde

Alt exemplu:  
Te conectezi la internet, nu merge, scot cablul și îl bag la loc (procesul durează multă, tu îl întrenui și apoi tib.  
Nă începă iar

## PortFast + BPDU Guard

STP - creat pentru evitarea loopurilor intre switch-uri



Dacă B nu are protectie, acceptă ceea cea = loop

Dacă B are BPDU guard enabled = B vede BPDU, realizează că e conectat la alt switch, blochează portul => Enabled

### Comenzi

spanning-tree portfast

spanning-tree portfast default (apoi se dezactivează de pe porturile medii)

show spanning-tree summary (nă vedem dacă portfast/bpdu e enabled / disabled)

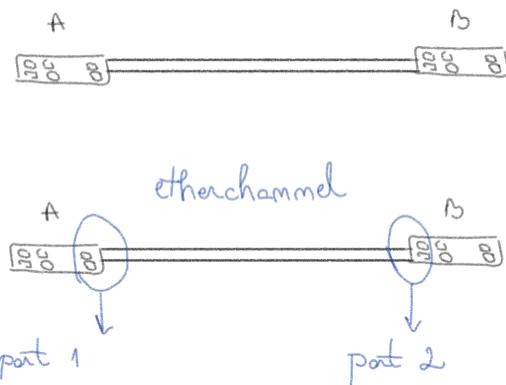
show spanning-tree interface fastethernet 0/1 portfast (dacă portfast e activat pe interfață respectivă)

spanning-tree bpduguard enable

spanning-tree portfast bpduguard default

show running-config

## Etherchannels



→ în mod normal, STP nu bloca unul dintre porturi

Se creează propria interfață logică și STP o primează direct în forwarding state

Se poate dubla capacitatea informațiilor  
Dacă un cablu e rros / nu mai funcționează,  
se utilizează cel rămas

### Proprietăți

1. Se comportă ca o singură interfață

- permite datele să "circule" în comunicație și în lipsa uneia dintre cabluri
- evitarea loopurilor (să nu se trimită informația pe un cablu și să revină înapoi pe celălalt)

2. Pot să fie până la 8 cabluri paralele

3. 3 metode de configurație statică, PAgP, LACP  
recomandate

4. Facilitarea traficului de date

PAgP - Port Aggregation Protocol

LACP - Link Aggregation Control Protocol

### Reguli pentru funcționare

Toate porturile în etherchannel - trebuie să aibă același

- duplex
- viteză

- același port / trunk port

↓                            ↓                                      ↓  
același VLAN                același allowed VLAN      deobicei

→ același allowed VLAN și matrice VLAN

- STP interface settings (ex: port priority)

## Keywords

0m / 0m	Static
Deminable / Demnable	PAgP
Deminable / Auto	PAgP
Active / Active	dACP
Active / Passive	LACP

retransmite información,  
debe ser creada un  
etherchannel

Channel Group = Port Channel = Etherchannel

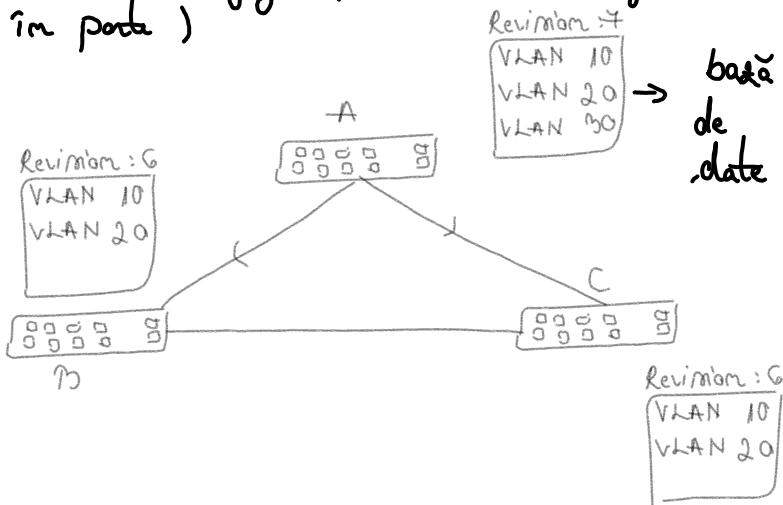
## VLAN Trunking Protocol (VTP)

- le permite switch-urilor să îmormătăsească configurațile VLAN (atât an într-o introducere de mărire la fiecare în portă)

### Summary Advertisements

- se trimit automat la fiecare 5 minute

- nume VTP
- parola VTP
- revision nr.
- followers\*



### Subset Advertisements

- nume VTP
- toată informația VTP

Se fiecare dată când e modificată baza de date  $\Rightarrow$  newrev + 1  
la fiecare 5 min se trimit de la fiecare sw. către toate restul summary advertisements

Se verifică nr. revizunii. Dacă un nr. are newrev mai mare, în funcție de followers\* se trimit celelalte nr. subset advertisements (adică acum se trimit informații despre VLAN - rămâne să fi fost trafic de date între).  
Celelalte sw. își actualizează baza de date și se trimit. Ian summary adv

## Moduri VTP

- Server - poate crea VLAN-uri  
- trimit update-uri în adv. către bazele de date a VTP

- Clienț - nu poate crea VLAN-uri  
- poate doar să trimită update-uri în adv. către bazele de date a VTP

- Transparent - poate crea doar VLAN-uri locale  
- nu dă update-uri sau adv.  
- poate doar să trimită mai departe update-uri în adv. între alte sw, dar el le ignore  
- nu are niciun impact asupra bazei de date VTP

## Reguli

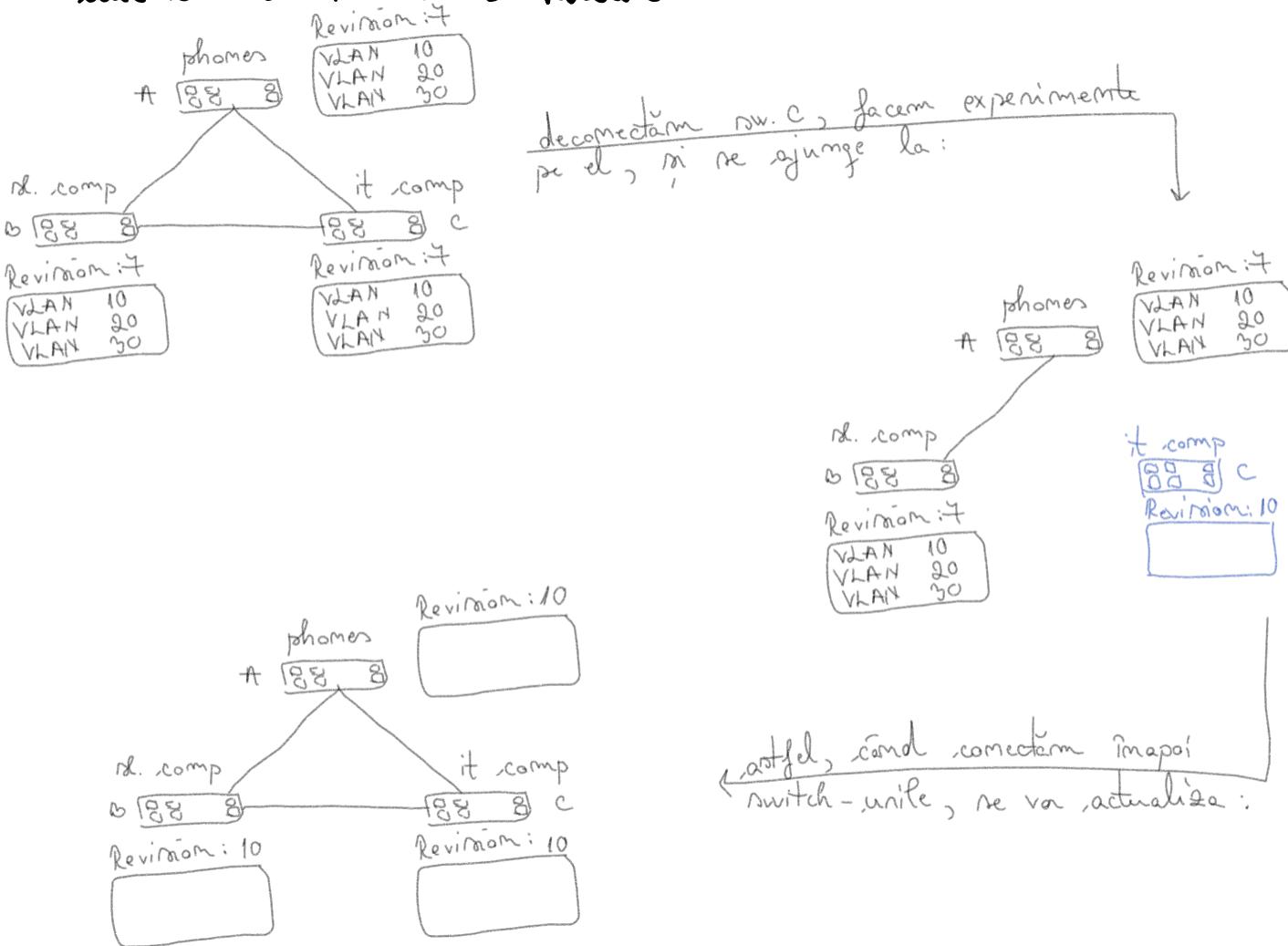
1. Link-urile trebuie să fie trunks
  2. Toate switch-urile din același VTP domain trebuie să aibă același VTP domain master
  3. VTP password să fie la fel la toate
- ↳ este optimă

I) nu se trimite  
niciun mesaj pe  
porturi de acces

↳ important pt. SW, ca să stie ce  
mesaje să asculte și pe care  
nu le ignore

## DEZAVANTAJ

1. amintim că VTP-unile se actualizează după versiunea bazei de date cu cea mai mare valoare

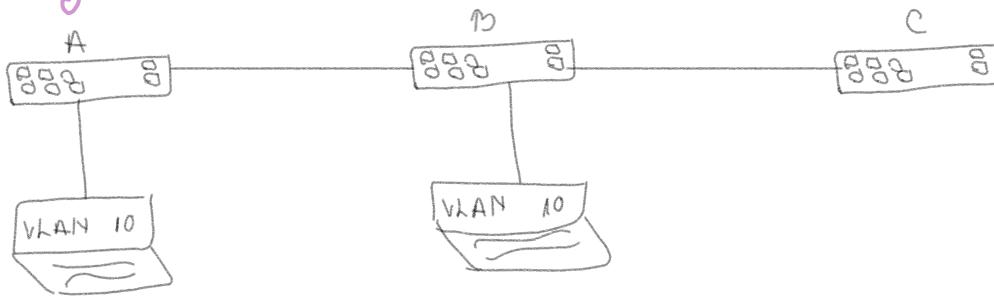


Exemplu comenzi

```
vtp mode server
vtp domain dama
vtp password dama
vtp pruning
interface fastEthernet 0/1
switchport trunk encapsulation dot1q
switchport mode trunk
```

enable  
configure terminal

```
vtp mode client
vtp domain dama
vtp password dama
interface fastEthernet 0/1
switchport trunk encapsulation dot1q
switchport mode trunk
```

Pruning

- permite să „zică” pt ce VLAN -uri său porturi

⇒ înloc ca A și B să îi trimită informații anunță lun C (care le-ar ignora), c „spune” din start că nu acceptă VLAN 10 ⇒ A și B nu îl mai trimit lun C → salvare de rezurse

## Rutare

= procesul prin care pachetele de date sunt redirecționate în rețele diferite

### Tipuri

#### 1. Rutare statică

- rute introduse manual de administrator  $\Rightarrow$  modificare și remunerare în cazul unei schimbări în rețea
- $\rightarrow$  posibilitatea controlului strict a dimensiunii tabelelor de rutare

#### 2. Rutare dinamică

- tabelele de rutare sunt construite automat cu ajutorul protocolelor de rutare: RIP, OSPF și BGP

permet noptenelor să comunice între ele și să schimbe informații despre rețeaua rețelei  
 $\Rightarrow$  noptenile pot lăsa deciziuni mai bune de rutare și pot adapta tabelele de rutare când se manifestă schimbări în rețea

## Agregarea rutelor

- practică de grupare a unor trasee între-unul sau mai multe specific  $\Rightarrow$  reduce complexitatea în dimensiunile tabelelor de rutare

## Rute implicite

- adresa IP a gateway-ului implicit

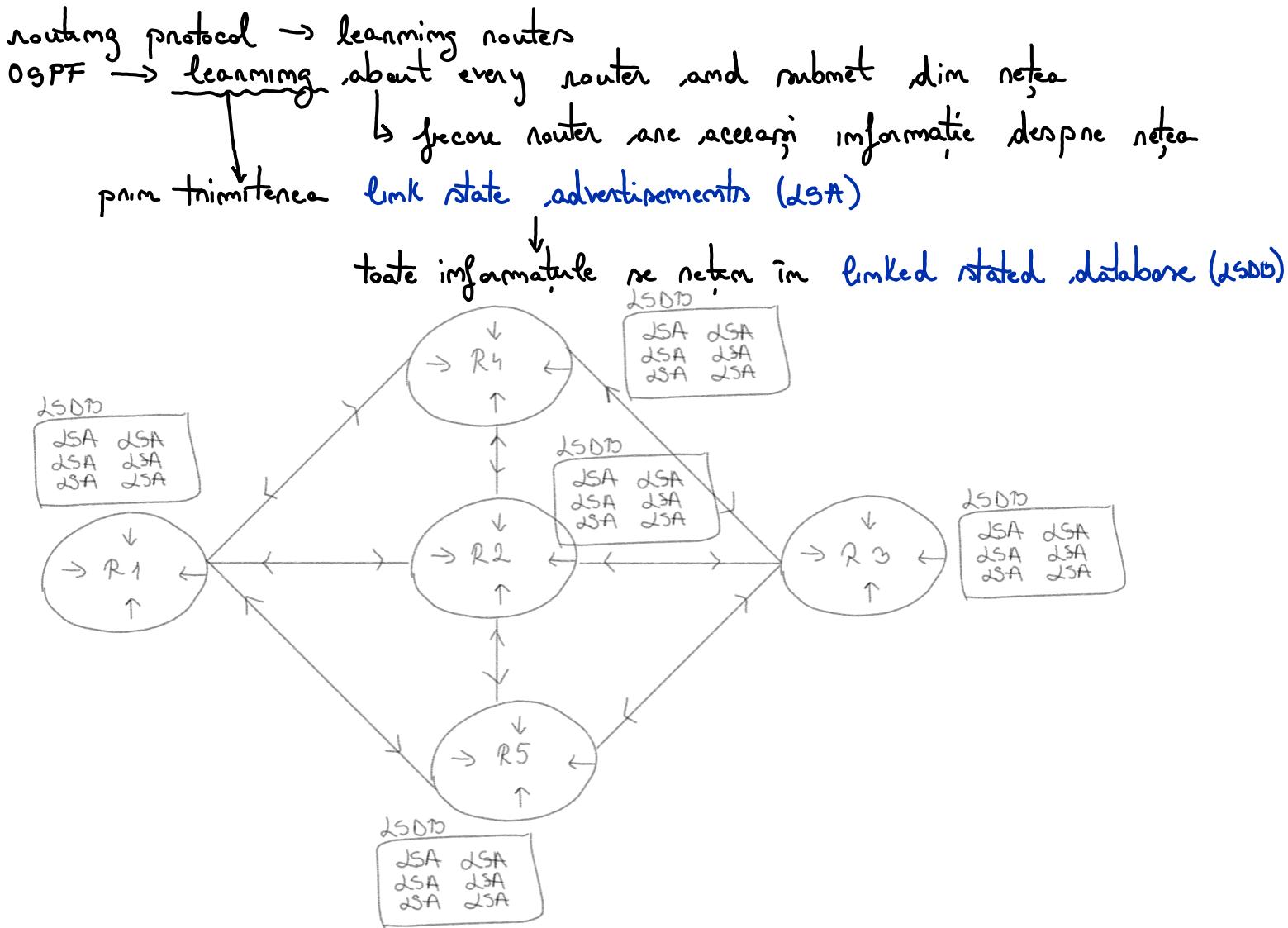
Tabelă de rutare = listă de rute disponibile peținu unui router

- informații despre rețele disponibile, adrese IP și metrice (=costurile) fiecărui nod

$\hookrightarrow$  minimă = conține toti rutele necesare și conectarea rețelei la Internet / altă rețea

## OSPF (Open Shortest Path First)

widely used and supported  
IGP (Interior Gateway Protocol)  
link-state routing protocol



### Pasi

1. **Become neighbours**
  - 2 ruteuri dim accezui rețea acceptă să creeze o relație de vecine (folosind protocolul OSPF)
2. **Exchange database information**
  - vecini fac schimb de informații dim LSDB între ei
3. **Choose the best routes**
  - fiecare ruteur adaugă în routing table cele mai bune variante din funcție de informațiile dim LSDB

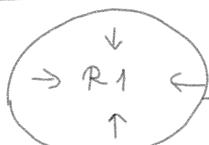
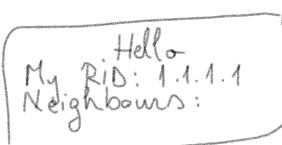
## Router ID (RID)

- nr. folosit pentru identificarea unui router individual
- de forma unei adrese IP, 4
- se poate seta manual / automat

oridice manually assigned

highest 'up' status loopback interface IP addn

highest 'up' status mon-loopback interface IP addn.

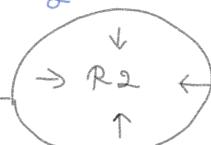


Down

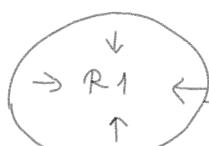
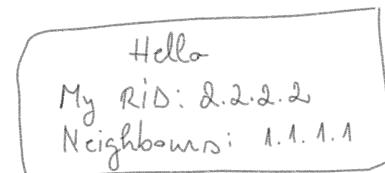
Requirements match?

intervalul pâră să crească  
că ceva nu e ok, by default  
e 10 sec. pînă în Hello

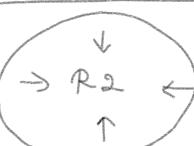
Area ID connecting links on the same subnet  
Subnet Hello and Dead interval  
Authentication (nă fie la fil)  
Sub area flag  
Unique router ID



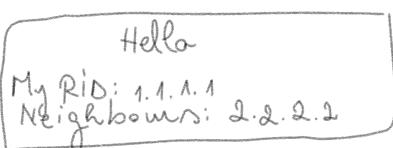
Down



Down



Ymit

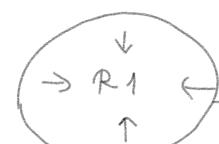


2-way

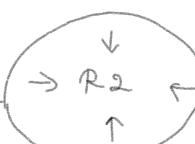


Ymit

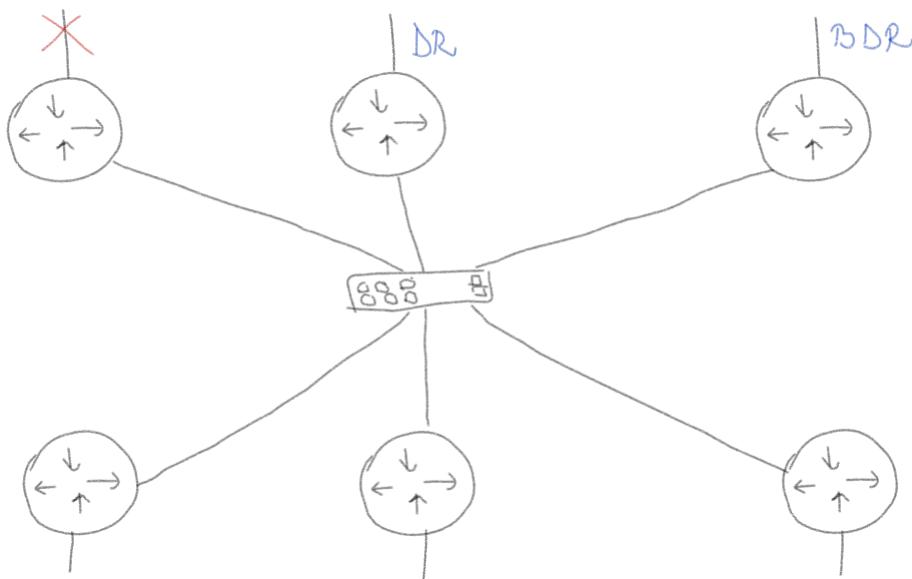
, acum sunt gata pentru schimb  
de informații



2-way



2-way



### Designated router (DR)

- router care devine responsabil pt. gestionarea actualizării OSPF

### Backup Designated Router (BDR)

- preia funcția de DR dacă acesta este înlocuit

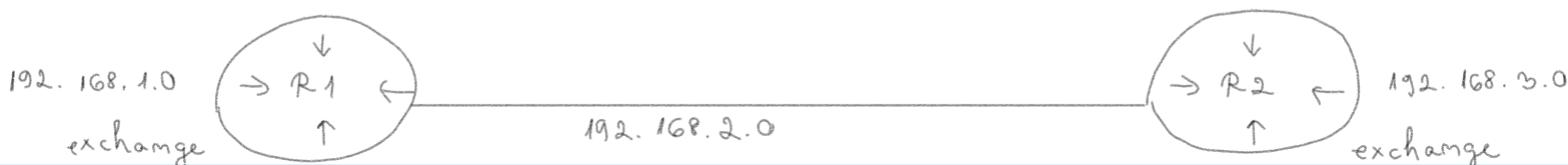
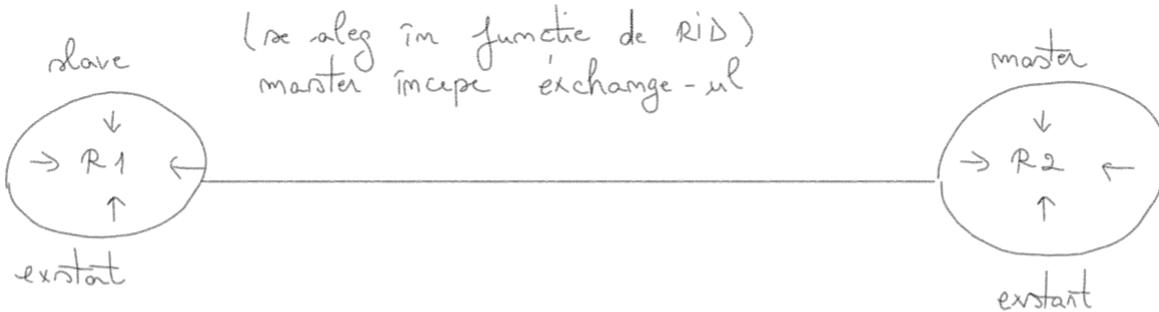
Precupunem că un router este rezervat 0 să se transmită modificării sau mesajele la toti vecinii și apoi de la fiecare la fiecare.  $\rightarrow$  avem nevoie de DR și BDR.

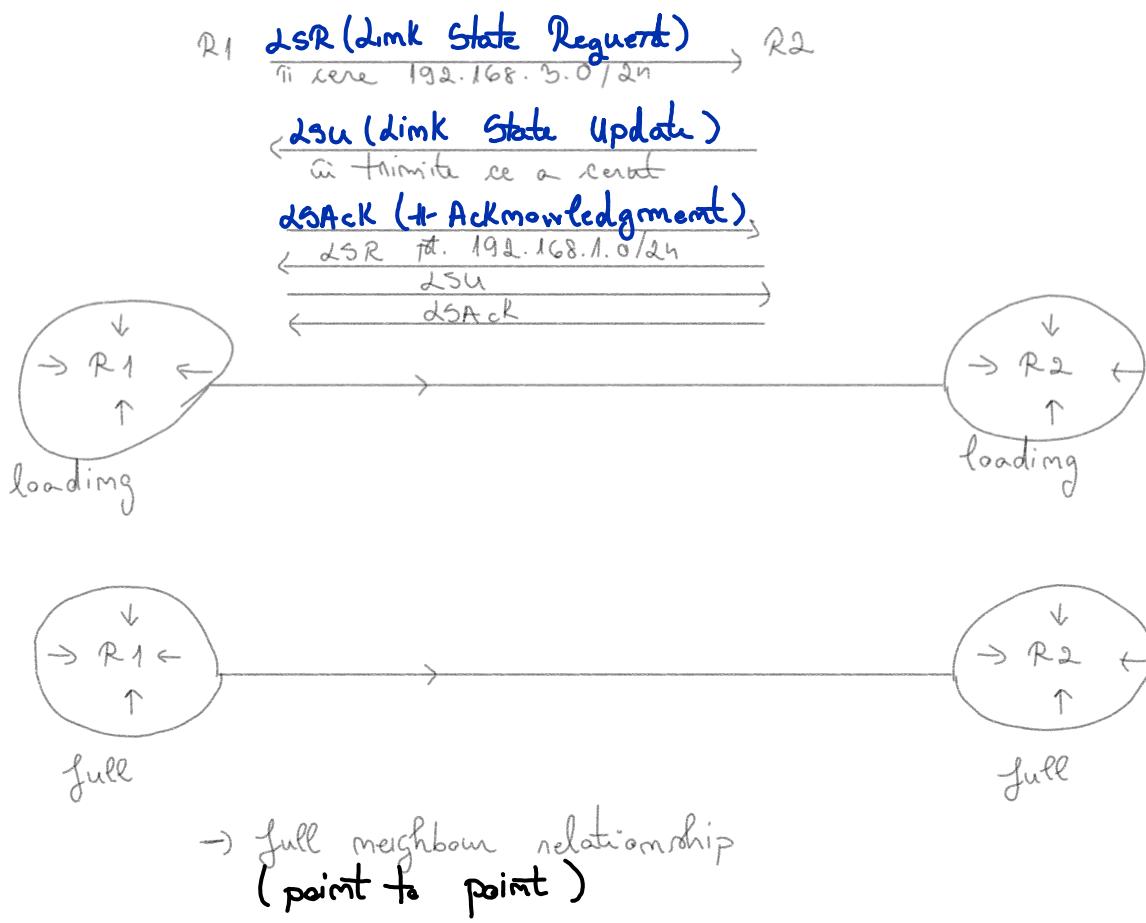
Această, dacă un router a șvadat, el transmite modificare la toti vecinii, dar toate routenele în afara de DR și BDR o ignoră. Astfel, se ocupă DR apoi să modifice routenele (care ascultă ce scrie DR și BDR)

Cum se aleg DR și BDR?

1. Prioritatea OSPF (cea mai mare (default e 1, dar se poate seta))
2. Router ID-ul cel mai mare

În același segment, routenele devin full neighbours doar cu DR și BDR. Restul vecinilor rămân în 2-way state (bagă în seamă doar updateurile venite de la DR/BDR)





### Adăugarea celor mai bune note la routing table

OSPF Cost = value given to a link based on the bandwidth of the interface

default:  
100 000 kbps ← Reference bandwidth / Interface bandwidth  
kilobits/second

Interface	Default bandwidth	Cost
Serial	15000 kb/s	64
Ethernet	10 000 kb/s	10
FastEthernet	100 000 kb/s	1

### Bandwidth

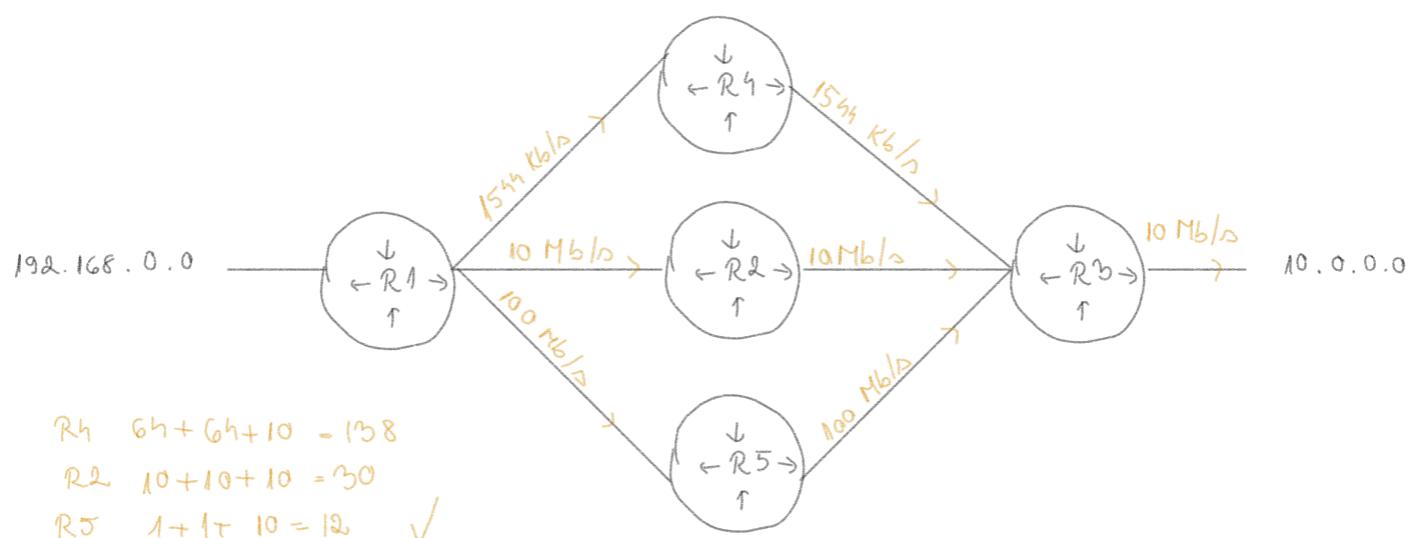
= capacitatea de transmitere a datelor printre-un mediu de comunicatie

- se măsoară în bit/s sau multipluri de bit/s / secundă

### link

= conexiunea fizică / logică, dintre 2 dispozitive / moduri între-o rețea - poate fi:

- cablu fizic
- conexiune fără fir
- legătură logică între 2 routere / switch-uri

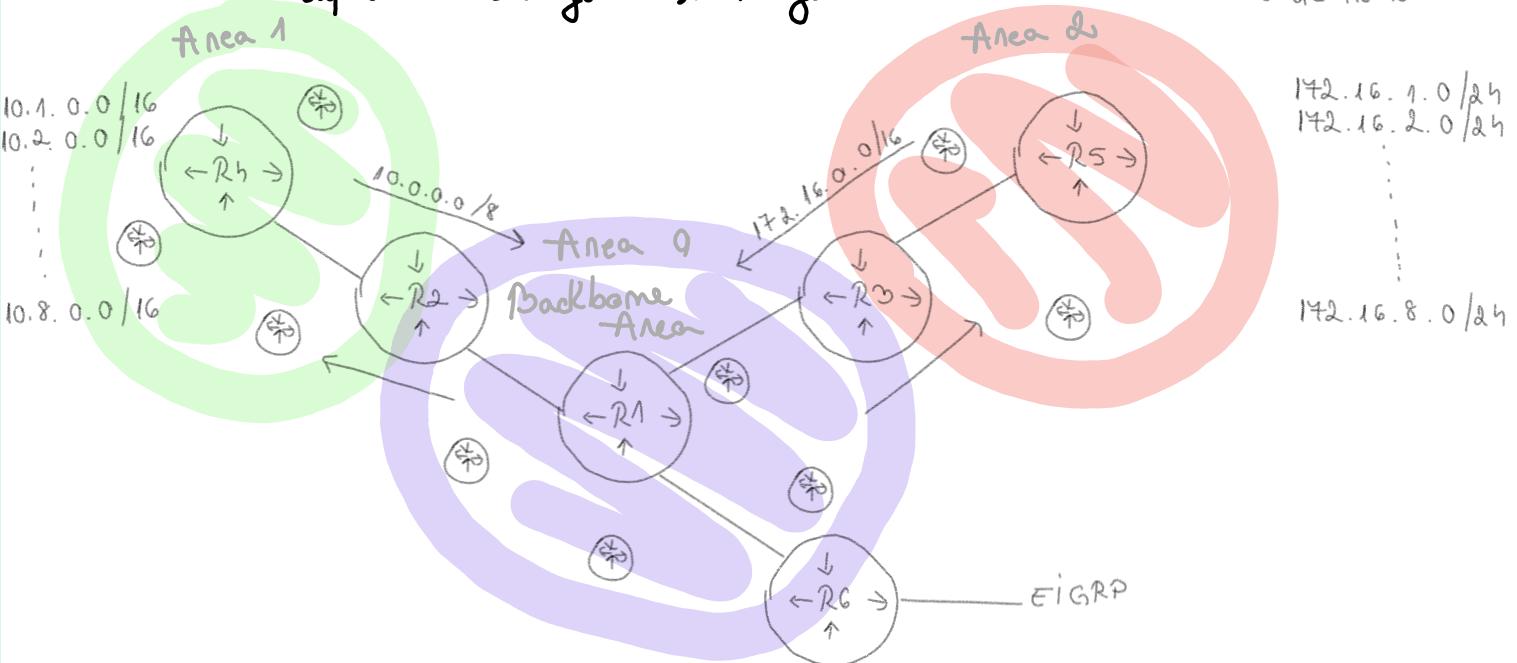


## OSPF Multi Areas

- Beneficii - reduce dimensiunea DSDTs  
 - "condensează" în routing tables  
 - update messages to a single area

Area = grup de noutre

recomandat să fie max.  
50 de noutre



### 1. Area 0 (Backbone Area)

- toate celelalte noutre să se conecteze la ea

### 2. Subnetting

- să se grupeze să fie de același "tip"  
 ex: toate care încep cu 172.16. ceva

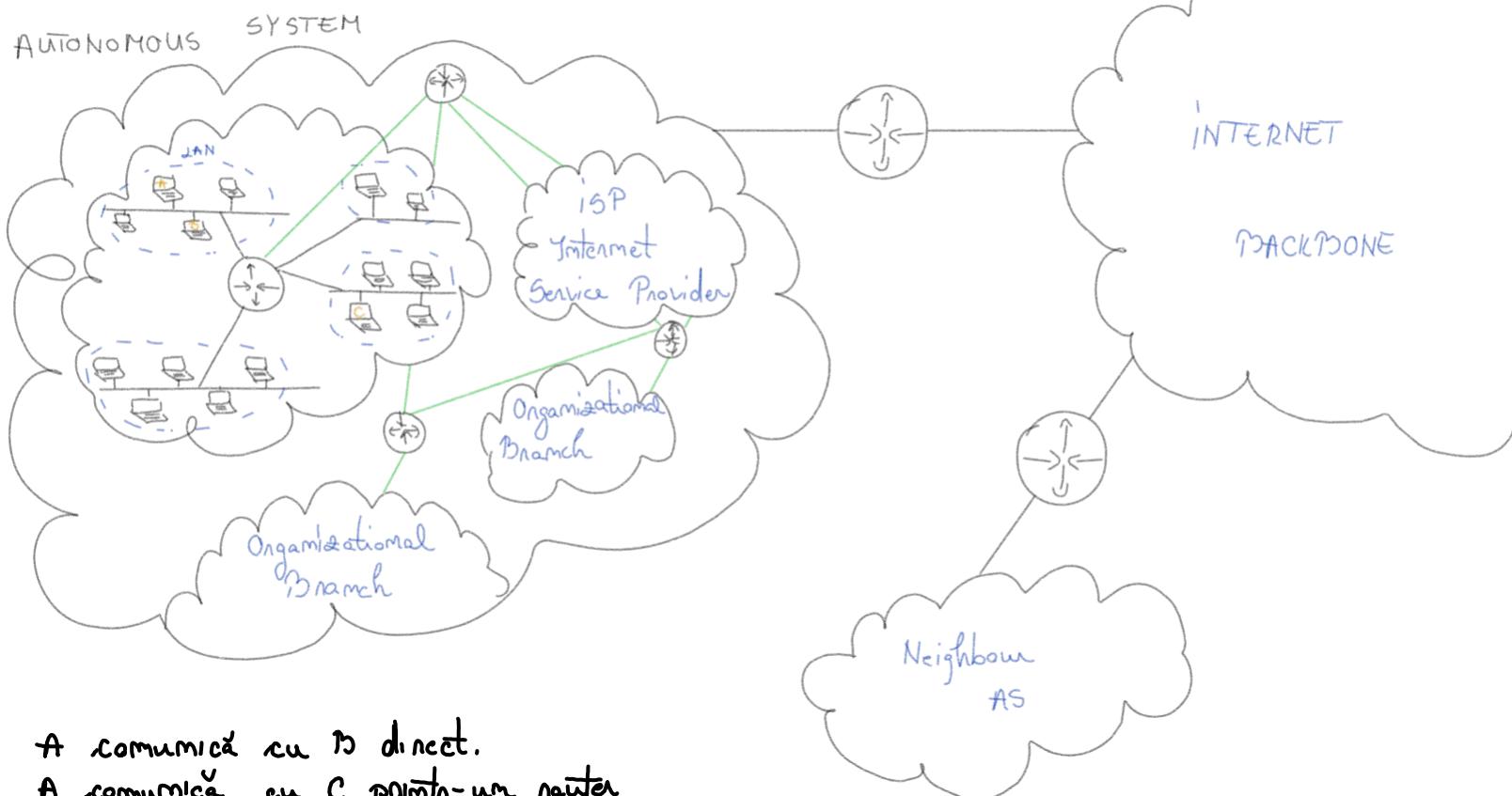
Backbone router R1, R2, R3

Area Border Router (ABR) R2, R3 → intermedioare

Internal router R4, R5 → mănuștează cu alte aree

Autonomous System Boundary Router (ASBR) R6

## BGP (Border Gateway Protocol)



A comunica cu B direct.

A comunica cu C printr-un router

↳ folosește diverse protocoale, nu poate include BGP

În același rețea  $\Rightarrow$  Internal BGP

Router conectat la AS, care se conectează la alt router conectat la alt AS  $\Rightarrow$  external BGP

sisteme autonome

vecini BGP (AS-URI)

comenzumi TCP între vecini

informații despre rute

actualizări periodice

Path Vector Protocol  $\Rightarrow$  informații se transmit împreună

- cu ruta în sine  $\Rightarrow$  evitarea buclelor în rutare BGP

Autonomous System

= colecție de rute și rețele care au același administrator și au același politici de routare

- se identifică printr-un număr unic

## RIP (Routing Information Protocol)

- = protocol de rutare cu vectori de distanță
- utilizarea "hop count" pentru alegerea rutelor

actualizări peste rețea UDP (RIP advertisements / updates)

hop count: hop = traseu pînă la un router

rute mai bune = cele mai puține hopsuri

primul actualizări prim broadcast și își actualizează tabelele de rutare

valoare maximă: 15 (RIP v1)

16 (RIP v2)  $\Rightarrow$  rute imacessibile  $\Rightarrow$  evitarea comunității la  $\infty$

split horizon: nu trimite informații despre rutele învăluite pînă la interfața primă aaceeași interfață

hold down timer: perioadă de timp în care routerul nu primește actualizări pe o rute, dacă aceasta a devenit imacessibilă

actualizări + alte caracteristici  $\rightarrow$  convergență rapidă

- RIP v2 - subredire
  - autentificare
  - adrese IP v6

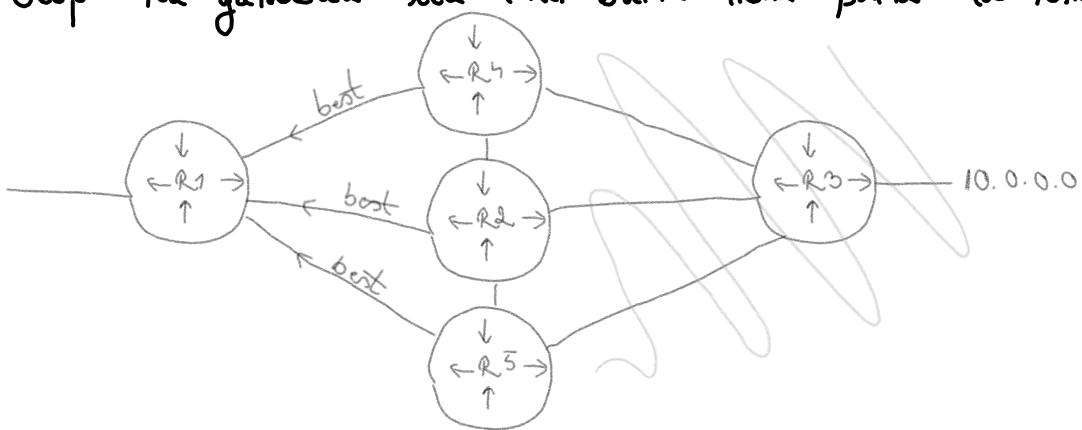
**Placa de rețea - device**  
 fizic prim care se conectează calculatorul cu extinderile. Pe o placă fizică de rețea se pot pune mai multe adrese IP  $\rightarrow$  interfețe

## EIGRP

(Enhanced Interior Gateway Routing Protocol)

- released as an informational RFC 7868
- used within a single autonomous system (retea independentă)
- distance vector and link-state like features

Scoap să găsească cea mai bună rută pâră la unicore router, din retea



R1 vrea să găsească ruta pâră la routerul 10.0.0.0

Pentru el, „există” doar vecinii lui în acel casă, fiecare vecin are o drum acolo. Fiecare vecin încearcă să trimită cea mai bună rută găsită de el pâră la destinație. Apoi, R1 tot amintește vecinii, din stemele în tot oraș.

### 1 Become neighbours

- Hello - 5 sec pt high bandwidth links
- 60 sec pt low bandwidth links
- se trim. non-stop de la unul la altul

Hold timer - 3 x intervalul lui Hello (15 sec)

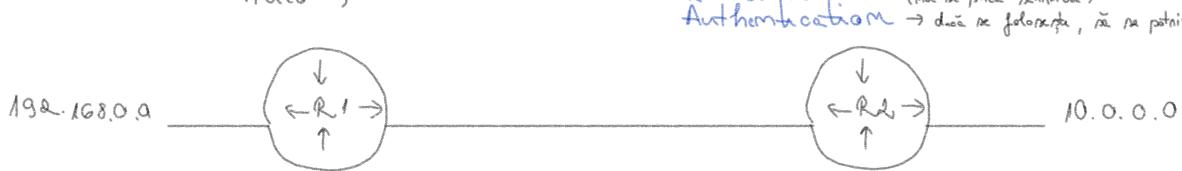
- urat se antreaptă pâră sănă că e „mort” celălalt, dacă nu trimit hello înapoi

Multicast - 229 0 0.10

Requirements match?

Hello și Hold intervalele nu trebuie să fie

B2: fie același  
Autonomous System → AS number  
Number (se configura la  
configurarea EIGRP pe rețea)  
Subnet → nu trebuie să fie același subnet  
K-values → nu trebuie să fie același K-value (nu se potrivesc pe rutele  
Autentication → dacă se folosesc, nu se potrivesc)



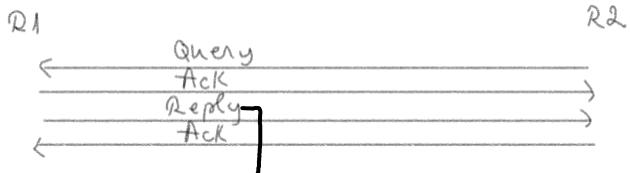
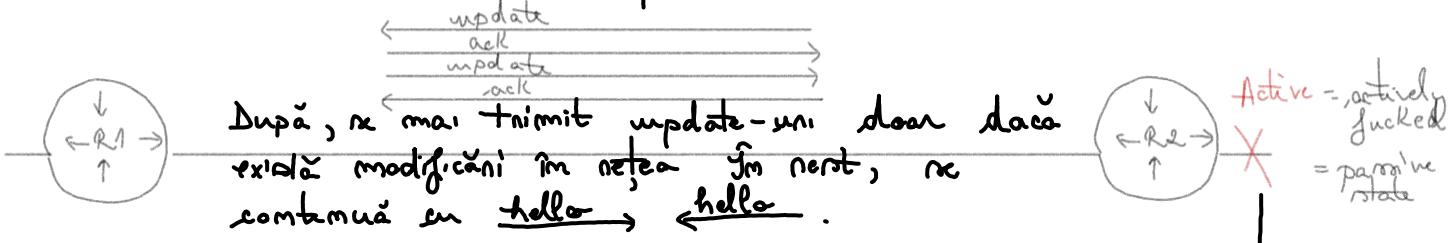
Dacă nu au îndeplinit condițiile ⇒ noutate vecine

## 2. Exchange routing information

-nu folosește UDP / TCP

-folosește RTP (Reliable Transport Protocol)

- folosește sequence numbers ca să identifice dacă mesajele au fost primite de vecin
- "duel" pt a evita loopurile



dacă găsește e ok, dacă nu, trebuie eliminată din routing table

Dacă se întâmplă asta, nu va petrece un Route Read Computation: routernul încearcă să cante o nodă liberă până la răbuit

Dacă nu găsește, îi întreabă pe vecini, dacă au ei o nodă

### 3 Choosing the best routes

#### • Metric Calculation Formula (Default)

$$((10^7 / \text{decat Bandwidth}) + \text{Cumulative Delay}) * 256$$

value given to each outgoing link (microsec)

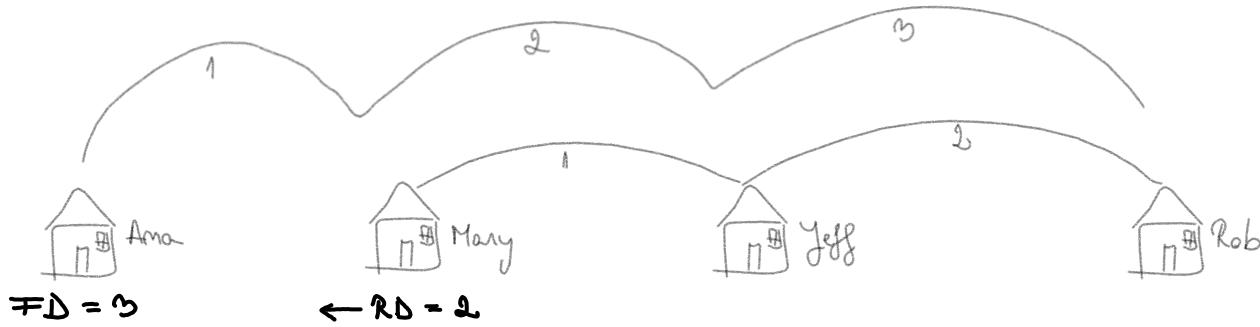


$$((10^7 / 100 000) + 10 + 10 + 10) * 256 = 33 280$$

#### • Reported Distance (RD) & Feasible Distance (FD)

the metric pt o nodă  
dpdv. a vecinului  
(advertised distance)

distanță raportată + distanță de la vecinul care me-a spus de nodă



Ama: Hei Mary, rău v-am de locuitoră Rob?

Mary: Da, la 2 case de mine!

Ama: Super, slăcă ești vecina mea, îmbeamă să la 3 case de mine!

### • Successor vs Feasible Successor

↓  
noda cu cel mai  
bun metric până  
la desfumare  
(pot fi mai mulți)

→ backup route în caz că este succesor  
→ trebuie să aibă  $RD < \text{Successor FD}$  (pentru evi-  
tarea loopurilor) (nu poate folosi și dacă nu se  
respectă imegalitatea, dar trebuie verificata loopurile)

### • Comenzi

bandwidth x

show ip route eigrp 1 b 10.0.0.0

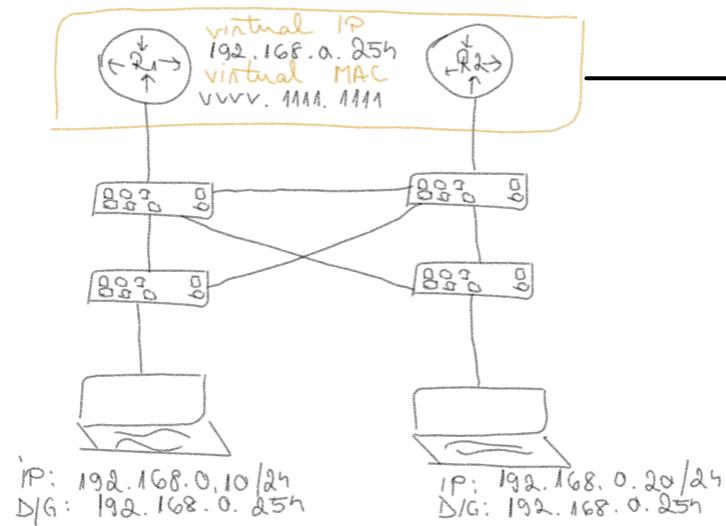
begin

→ schimbă valoarea în x

show ip eigrp topology

→ anată toate informațiile, inclusiv  
s și FS (schior în  $(FD, RD)$ )

## Frist Hop Redundancy Protocol (FHRP)



se creează un grup pentru a evita eșuaile care să se petnească dacă un router nu va fi conectat la un singur router, care ar putea emula, dacă un dispozitiv ar cumora, datele unui singur router etc.

→ în cazul în care un router eșuează și cel rămas nu primește nicio cerere pentru adresa MAC (deci dispozitivele vor să se folosească de tabele de adrese), acesta va trimite un arp, ca dispozitivele să își actualizeze tab-

FHRP

HSRP  
(Hot Standby Router Protocol)

VRRP

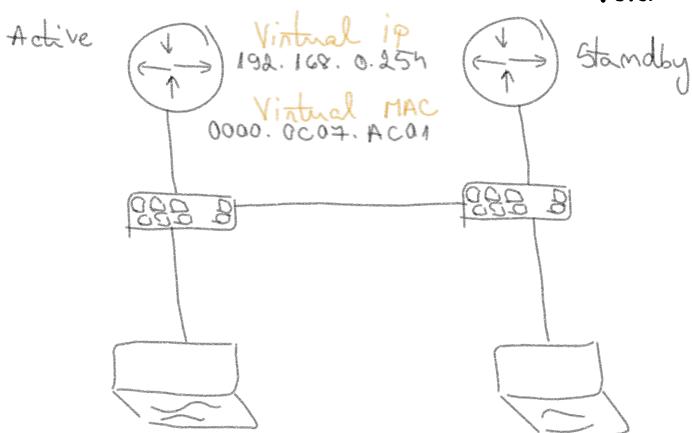
(Virtual Router Redundancy Protocol)

GLBP  
(Gateway Load Balancing Protocol)

### HSRP

- dispozitivele trimet și primesc multicast UDP hello packets la fiecare 3 sec

Venom 1 22h.00.2  
Venom 2 22h.00.102



Active Router Election

Highest HSRP Priority  
↓

Highest ip Address

Virtual IP - configurat să devină default gateway

Virtual MAC - se generează automat

Versiune 1

0000.0C07 ACXX

Group ID

Versiune 2

0000.0C9F TXXX

Group ID

- dacă active router e prezentă, standby devine automat active și îi sumează în pe neregulă

→ dacă suntem nevinde, devine standby (repete rețea care standby preia dăcă vrei să fie activ)

### VRRP

- în loc de active și standby  $\Rightarrow$  master și backup

- unul dintre routere preia Virtual IP  $\Rightarrow$  IP Address Owner

### Master Router Election

IP Address Owner

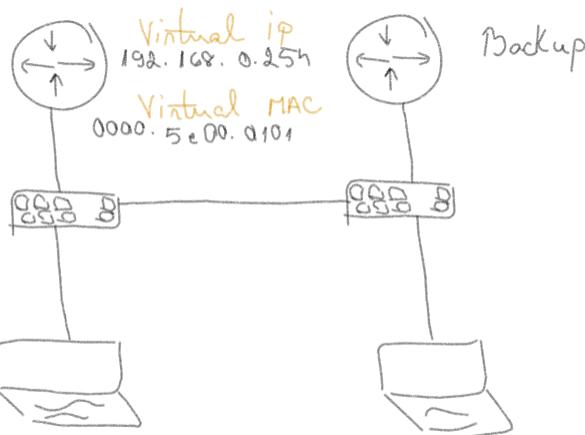


highest priority



highest IP-Address

Master



D/G: 192.168.0.254

D/G: 192.168.0.254.

Virtual MAC: 0000 5e00.01XX

VRRP

Virtual MAC

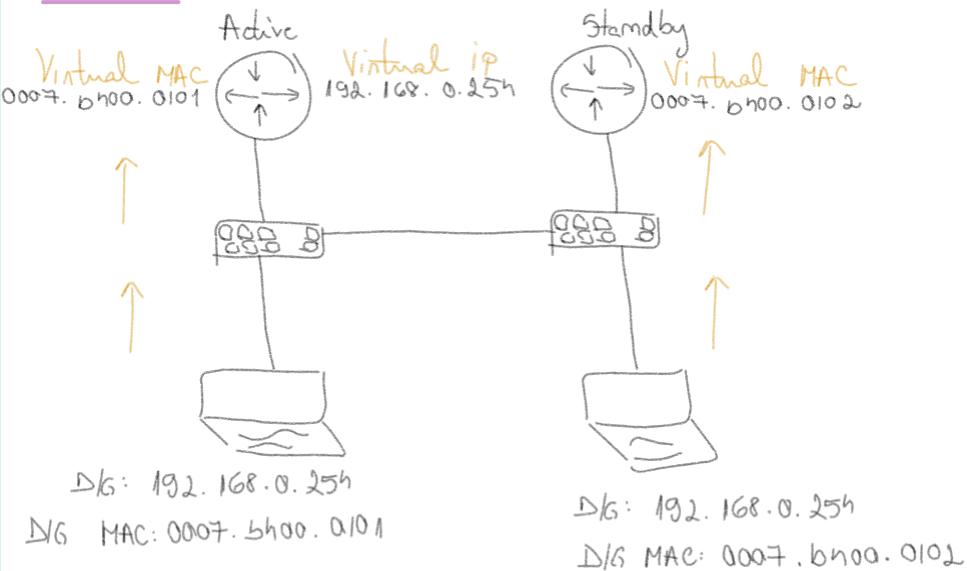
group id

- doar master trimite mesaje în rețea

- VRRP Master devices trimite adverseminte la adresa multicast 224.0.0.18 tot de 1 sec.

- dacă master ernează, se arreagați 3 rec. (3x adv timer) și urm. pac, apoi backup devine master
  - ↳ dacă împărește, devine iar master automat

### GLBP

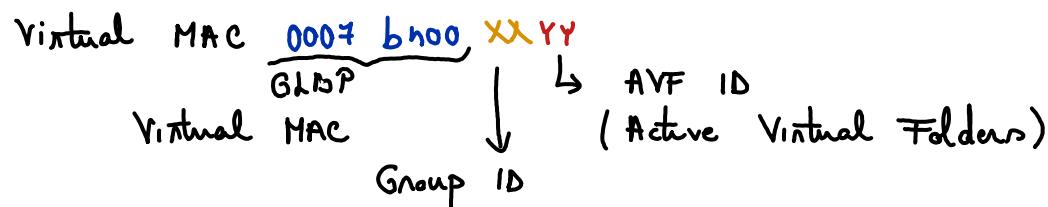


### Active Router Election

Highest Priority

↓  
Highest IP Address

- fiecare router trimite "hello" ramsa ca să comunice între ele (multicast UDP)
  - la fiecare 3 sec
  - Multicast: 224.0.0.102
  - UDP Port 3222



- chiar, dacă host-urile au aceeași default gateway IP address, routerele pot să răspundă cu adrese MAC diferite ⇒ se pot folosi ambele routere ca să ne flindizeze traficul, în loc să fie numul routere care este numul standby
- dacă ernează active, se arreagați 10 sec., apoi standby devine active și preia adresele MAC
  - ↳ dacă împărește, devine ca standby, dar își ia adresa MAC înapoi

Nº de adr. MAC  
virtuale core  
pot f. falante

	Router Roles	Multicast Addrs.	MAC addr. Format
--	--------------	------------------	------------------

<b>HSRP</b>	Cisco Proprietary	Active Standby	v1 224.0.0.2 v2. 224.1.1.102	0000 0C07 ACXX	One	Hello (L) 3 sec. Hold (L) 10 sec
<b>VRRP</b>	RFC 5498	Master Backup	224.0.0.18	0000 5e00.01XX	One	Advertisement (L) 1 sec. Master Down (L) 3 sec.
<b>GIDP</b>	Cisco Proprietary	Active Standby	224.0.0.102	0007 b400 XXYY	Four shared	Hello (L) 3 sec. Hold (L) 10 sec

## Network Address Translation (NAT)

### The problem

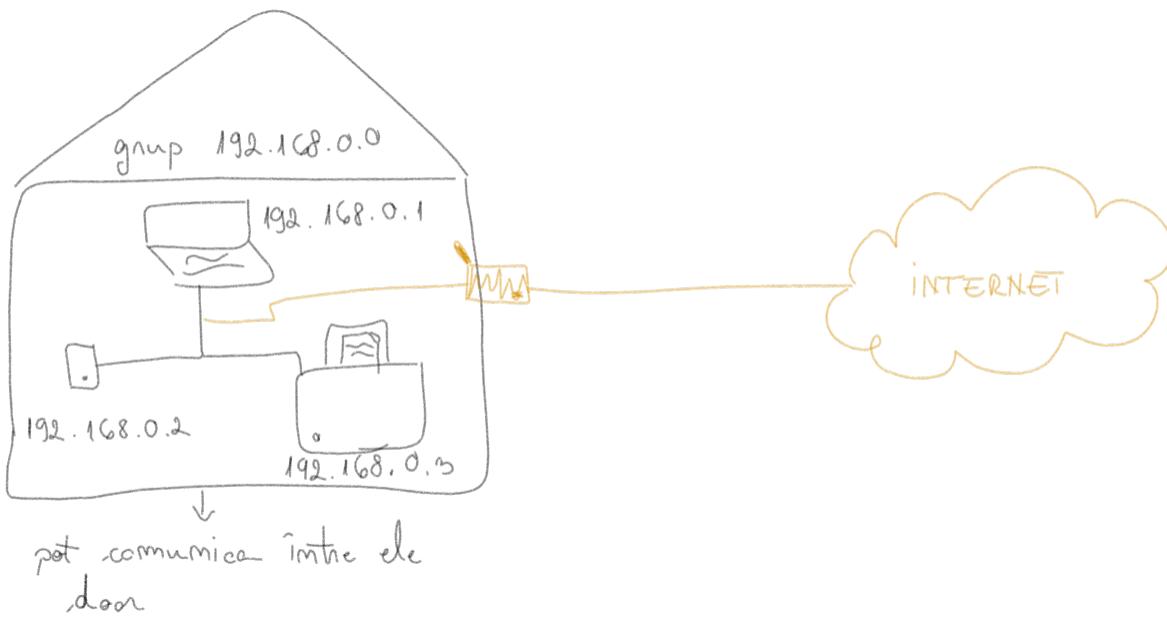
- internet a tot ,creare in un terminal ip - unile

### The solution - Private Addresses

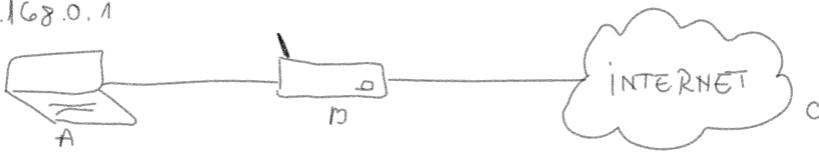
Se pot folosi doar în  
rețele interne, nu se poate  
în internet public

10.0.0.0 - 10.255.255.255.  
172.16.0.0 - 172.31.255.255  
192.168.0.0 - 192.168.255.255

NAT converteste adrese private în adrese publice



192.168.0.1



## Tipuri de NAT

### 1. Overload / PAT (Port Address Translation)

- cel mai popular

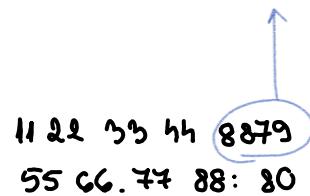
 $A \rightarrow B$ 

indica inclusiv aplicatia / tabelul de care aparține

Source 192.168.0.1 8897  $\xrightarrow{\text{noutenul le schimba}}$  11.22.33.44 8897  
 Destinatorm 55.06.47.88.80  $\xrightarrow{\text{schimba}}$  55.06.47.88.80

creaza tabelul, apoi trimite datele

de obicei se păstrează daca e scris, se foloseste un număr liber

 $C \rightarrow B$ 

Source: 55.06.47.88.80

Destinatorm 11.22.33.44 8897

Înainte	Înainte
192.168.0.1:8897	11.22.33.44 8897

55.06.47.88.80  
 192.168.0.1 8897

icau match

### 2. Dynamic

- funcționează corect, dar

$A \rightarrow B$  - noutenul alege prima adresa liberă găsită în 11.22.33.44 - 11.22.33.99

- se fac totuți pașii de mai sus (cu schimbare, tabel în tot)

$C \rightarrow A$  - se fac totuși totuși pașii

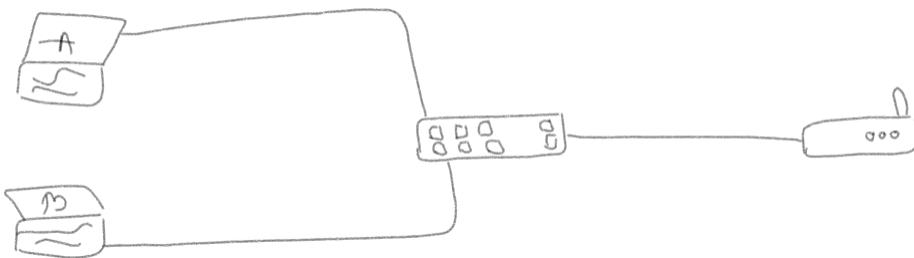
- după ce ajung datele înapoi la device, adrese ip minge înapoi în pașimă și va putea fi folosită iar

### 3 Static

- adresa privată și cca publică trebuie introduse manual
- în rest, funcționează la fel
- se folosesc mai mult porturi servere web (ex: http , unde portul < 80 )

## DHCP (Dynamic Host Configuration Protocol)

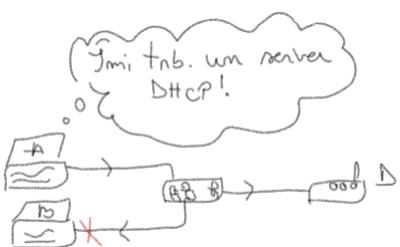
- assignează adrese IP unice device-ului
- client / server → UDP Port Client 68  
Server 67



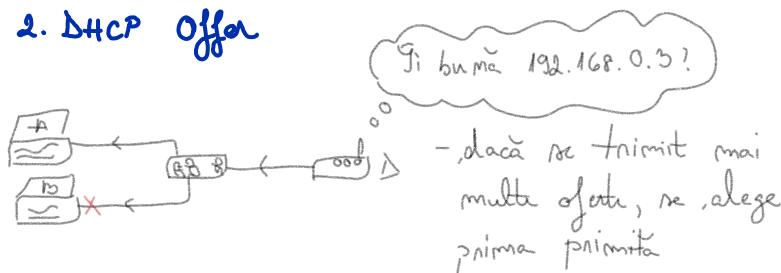
Adresa lui A în B trebuie să fie unică, ca să meargă datele unde trebuie

→ la ce urmează, se trimit mesaje broadcast, deci le primește tota lumea și cui nu îi se adresează, le ignorează

### 1. DHCP Discover



### 2. DHCP Offer



### 3. DHCP Request

A găsește că o are

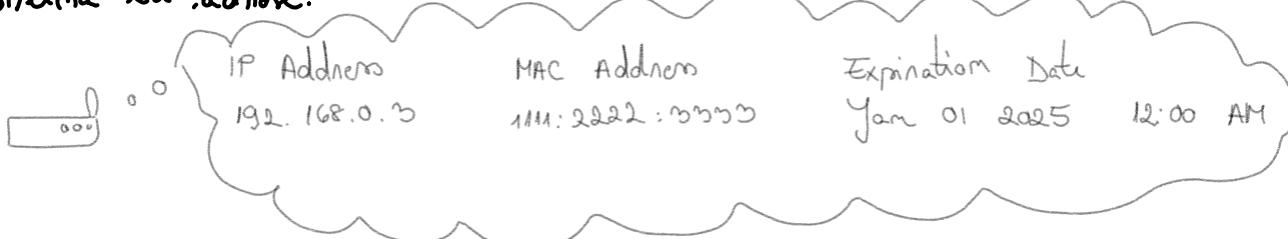
### 4. DHCP ACK

D: OK, fișoare, sănătate și sănătate

D → A: adresa IP, subnet mask, default gateway și serverul DNS

Serviciul DHCP trece apoi evidență

Dispozitivul trebuie să își rețină adresa, astfel expira și menține înapoi în prima cu adresa.



⇒ evitarea nimănui adreseelor IP (dacă suntem / deconectăm un dispozitiv)

## Syslog

Cinco device le rețin în RAM

### Syslog Server

- toate dispozitivele din rețea îi trimit log information  
UDP Port 514

- Beneficiu
- verifică toate informațiile mai ușor, dintr-un singur loc
  - date retention (când se rezarcă un device, logurile se sterg)
  - se arhivează mai ușor

### LOG information

= înregistrările / jurnalul  
care capturează evenimentele activității sau mesajelor într-un sistem / aplicație

### Jurnal de rețea

- informații precum - comenzi, decodări, trafic de date, erori de comunicare etc

### Data retention

= politica și practicele în ceea ce privește păstrarea și stocarea datelor pentru o anumită perioadă de timp

### Severity

= cât de urgent e logul

### Timestamp / Sequence number

Aug 26 18:04:43.647: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up

număr  
mesajului

### Facility

### Mnemonic

- cod pentru identificarea  
SMS-ului

Description  
= mesajul log

## Facility

0	Kernel	Kernel logs
1	user	user-level logs
2	mail	mail system
3	daemon	system daemons
4	auth	security / authentication logs
5	syslog	logs generated internally by syslogd
6	lpr	line printer subsystem
7	news	network news subsystem
8	uucp	UUCP subsystem
9	cron	clock daemon
10	authpriv	security / authentication logs
11	ftp	FTP daemon
12	ntp	NTP subsystem
13	security	log audit
14	console	log alert
15	solaris-cron	clock daemon
16-23	local	local use

## Severity

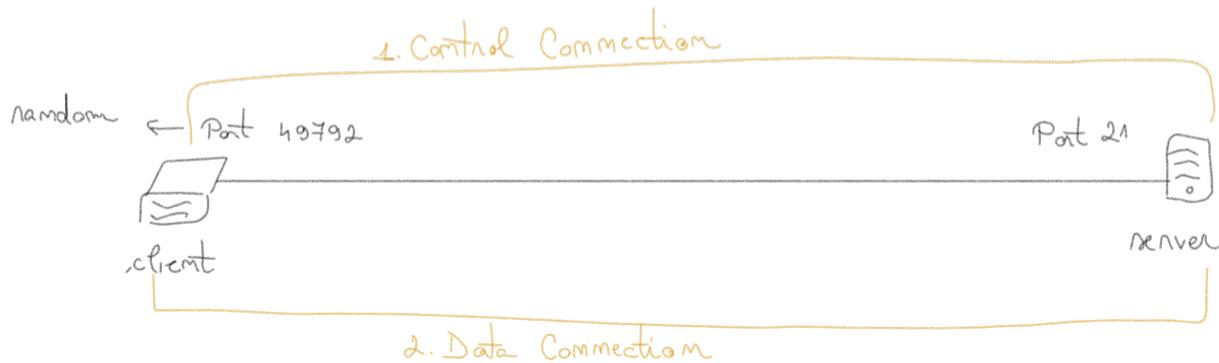
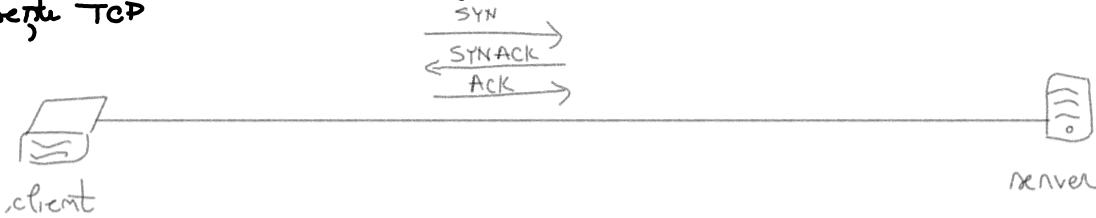
Fiecare grav 0 → 7 Nu este grav

Code	Severity	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant conditions
6	Informational	Informational messages
7	Debug	Debug-level messages

Când scriem log cu o anumită severitate, trebuie să le dă pe toate de la severitatea 0, sau în sensul că scriem "Informational", și să dă tot (6→0) pînă la Emergency

## FTP (File Transfer Protocol)

- = protocol folosit pentru transferul de fișiere între o rețea
- folosește TCP



### Data Connection

1. Active: serverul face primul pas, având ca port număr 20, către un port generat dinamic random
  - dacă există un firewall între client și server, cel mai probabil conexiunea mesajată către partea serverului nu să fie blocată
2. Passive: clientul face primul pas de pe portul număr generat random, către portul destinație 21
  - dacă există firewall, acesta nu blochează traficul, pentru că se alegea inițiat conexiunea, a făcut clientul
  - nu este necesar, deocamdată, toate datele sunt transmise clar

FTPS  
(FTP Secure / FTP SSL)

- extensie a FTP care supune utilizarea a TSL și SSL encryption
- protocol de criptare pentru menținerea datelor în siguranță și departe de hackeri  
 (îl primit ca și pe un tunnel, care nu lasă datele să se vadă)
- să nu confundăm cu SFTP (SSH File Transfer Protocol)  
 ↳ extensie a protocolului SSH

TFTP  
(Trivial File Transfer Protocol)

- = varianta mai "nedură" a FTP
- metodă simplă pentru un transfer de fișiere rapid și eficient
- folosește port UDP 69
- nu există autentificări, criptări, etc

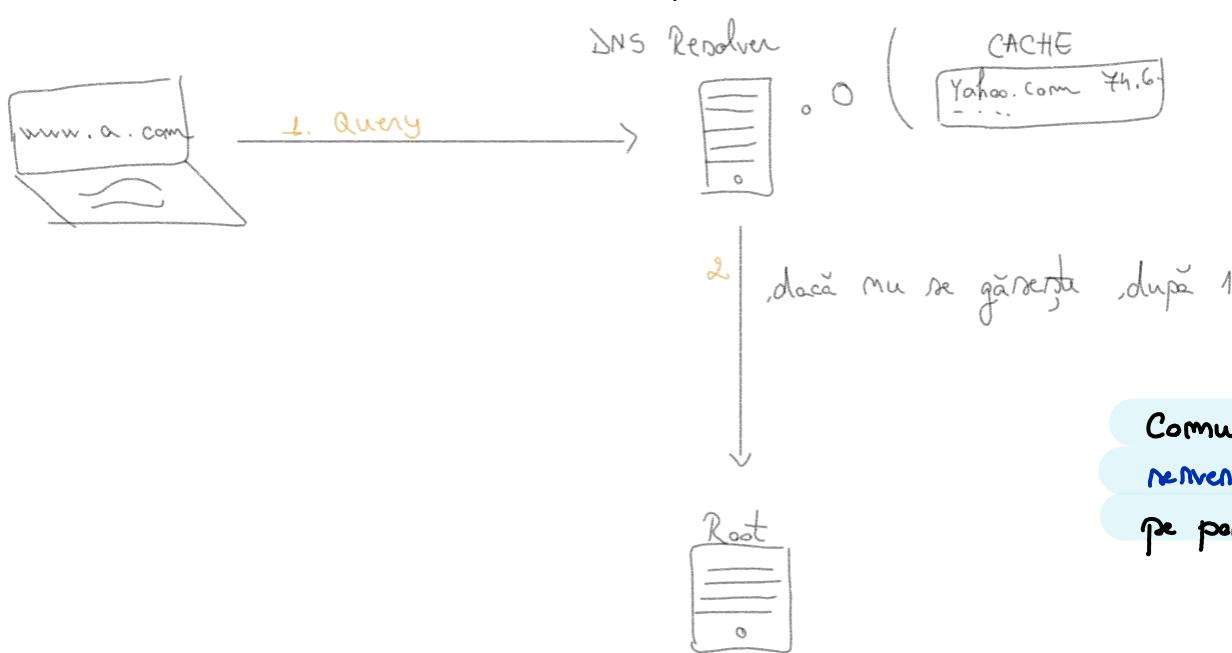
## DNS Domain Name System

- preia un link și îl transformă în adresă IP (serverele web funcționează cu adrese IP)



- se verifică local cache pe computer și browser
- se verifică o local configuration file

,dacă nu există date, se trimită un query care cere o adresă IP pentru www.a.com



### Root Name Server

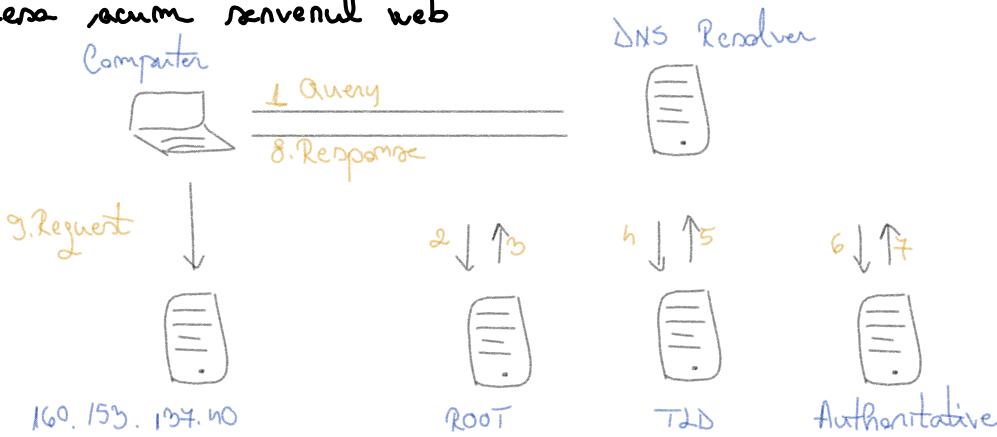
- primul din ierarhia DNS
  - primul pas pt. transformarea linkului în adresă IP
- ↳ există foarte multe, dar fiecare folosește 1 din 13 adrese IP
- Rol: să găsească deținătorul, după top level domain server (com, org etc)

### TLD Name Server

- conține informații pentru domenii with a specific extension (com, net, org etc)
- tot înțelesă IP de care avem nevoie
- ține locația lui authoritative master server

## Authoritative Name Server

- ultimul pas în obținerea răspunsului cerut
- conține informații DNS pentru domeniile de care se ocupă
- trimite adresa IP lui DNS Resolver, care o trimite computerului, care poate accesa același serverul web



## Lista de adrese IP pentru Root

198.41.0.4  
 199.9.14.201  
 192.53.4.12  
 199.49.1.13  
 192.203.230.10  
 192.55.241  
 192.112.36.4  
 198.94.190.53  
 192.36.148.14  
 192.58.128.30  
 193.0.14.129  
 199.48.83.42  
 202.12.24.33

### Type A

= înregistrare a unei IPuri (pt un domeniu)

- pt IPv4, AAAA

## Proxy Server

- server intermediar între un client și alte servere web
- performanță, securitate, confidențialitate ( poate bloca accesul la diferite site-uri web, poate limita accesul la diferențe surse de date)
- poate ascunde adresa IP a unui client  $\Rightarrow$  navigare anonimă pe internet

### Tipuri

Forward Proxy - client -> servere web

- performanță și securitate conexiunii la internet

Reverse Proxy - client -> unul / mai multe servere web

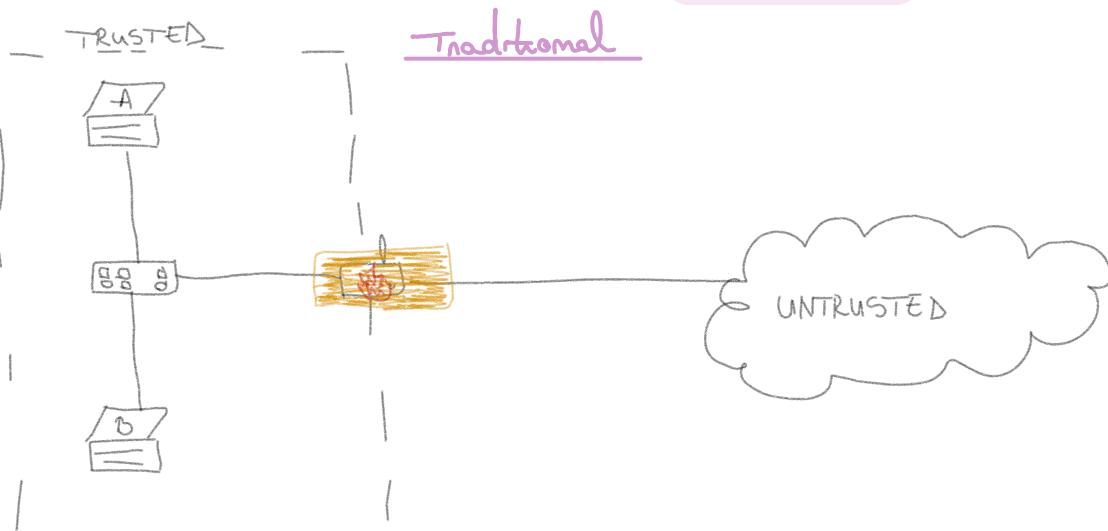
- performanță, securitate și confidențialitatea conexiunii la internet

Open Proxy - server intermediar utilizat de oricine pt a accesa internetul

$\downarrow$

în general pt navigare anonimă, ascunderea adresei IP

## Firewall



- cu scopul de a proteja rețelele trusted de cele untrusted
- by default, ele blochează tot traficul, dar vom să blocăm doar ce nu e bun

### Firewall rules

SOURCE	DESTINATION	PORT	ACTION
Host A	any	HTTP	allow

Acum, A va putea să trimită mesaje cui vrea. B, de exemplu, va fi în continuare blocat de firewall.

**Stateful firewalls** - monitorizează conexiunile active  
 ⇒ dacă lui A i-a fost permis să trimită date, e acceptat și traficul invers

### NGFWs (Next-Generation Firewalls)

Application level inspection (identifică și blochează)  
 IPS (Intrusion Prevention System) - patterns / signatures  
 - anomalies

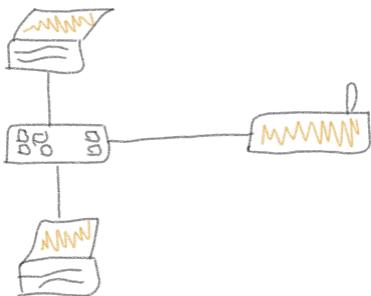
### Threat Intelligence (updates)

#### Features

URL filtering  
 email scanning  
 DLP (Data Loss Prevention)  
 etc.

} UTM (Unified Threat Management)

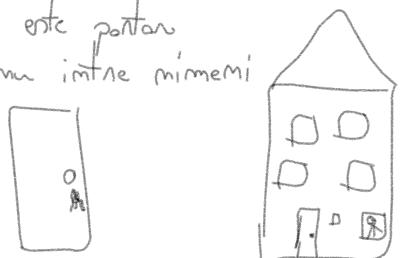
## Software based firewalls



- dublă protecție, dacă am emis o față vîme, dim ext
  - protecție, dacă vîme dim interior
- ex Windows Firewall

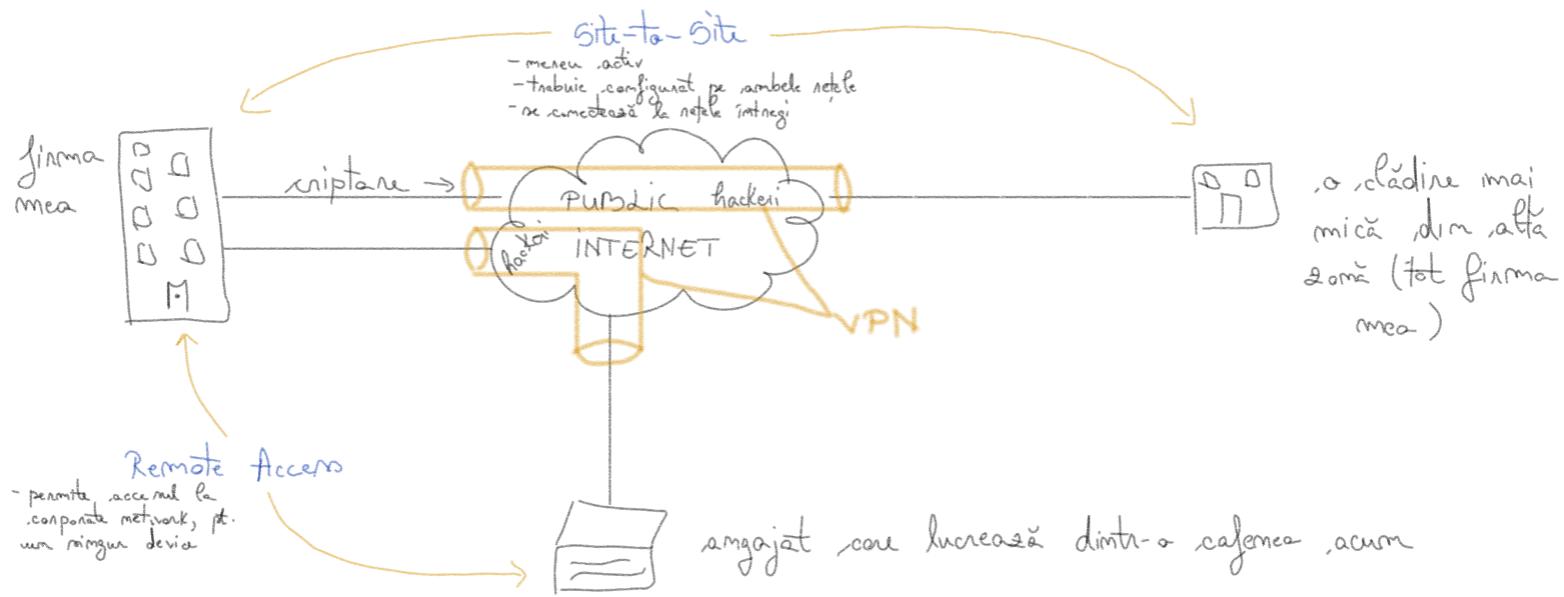
Ca și la cămin:

Teoretic nu pot intra persoane năzădate, pt. că și în trebuie, caseta și este poartă  
Dar mai tot ne închidem cu cheia usă de la cămină să nu nu intră nimic  
, care e în cămin deja ( sau pt. mai doar me poartă)



## VPN (Virtual Private Network)

- se ocupă de livrarea în siguranță a datelor în rețea publică



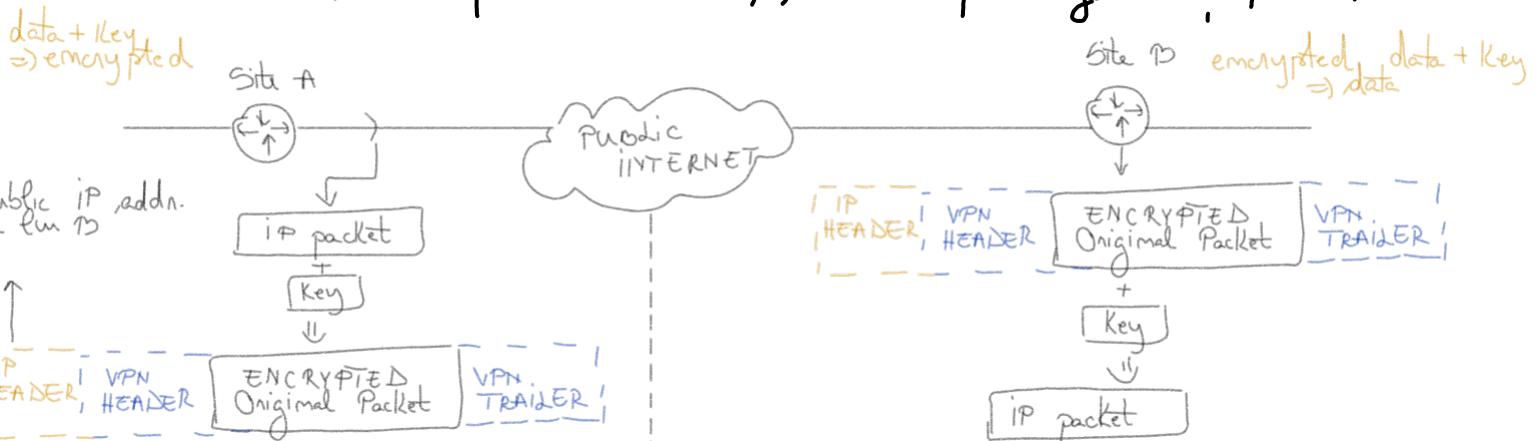
### Site-to-Site VPN



- se configurașă deobicei pe un router / firewall pe ambele rate-uni

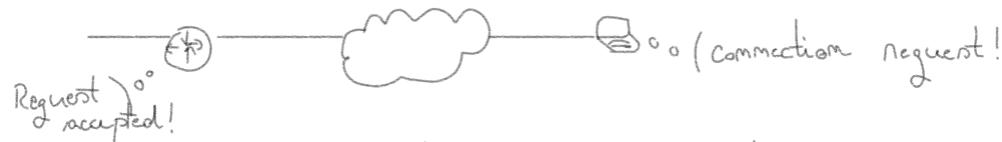
IPSEC VPN = framework / set of rules pt. crearea VPN-ului în rețea (= pt securizarea comunicării între o ip network)

- permite folosirea mai multor protocoale pt. găcire VPN feature
- deobicei pt rate-to-rate, dar se poate folosi și pt remote access



## Remote Access VPN

- permite conectarea unui singur device la o corporate network
- trebuie să existe către host ca să te conectezi la rețea



ex. de VPN Client Application: Cisco Anyconnect, OpenVPN

- se folosește în general **TLS (Transport Layer Security)**

↳ succesor a lui SSL (Secure Sockets Layer)

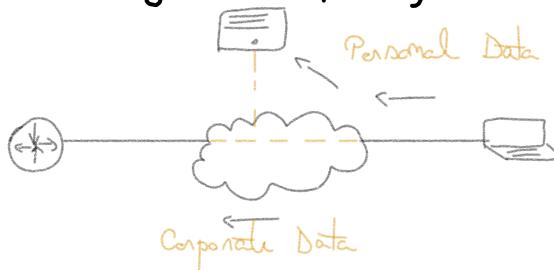
- se folosește și pt traficul web, la conectarea la rețeaua http

- folosește în general portul 443, care e permis în general (e bine, pt. ca unele wifi-uni publice blochează porturile IPSEC)

**Full tunnel** - dacă te-ai conectat la VPN, tot traficul te va transmite la corporate network (în dacă stai pe fb de ex)

**Split tunnel** - doar traficul destinat ei va fi transmis către corporate network

- bandwidth saving + user privacy



## ACL (Access Control List)

- = rule-based lists folosite de ruteaza si switch-uri pt identificarea traficului  
 - in functie de ip addrs.  
 (source addrs, destination addrs)  
 si port numbers
- in general folosite pentru deny/allow traffic
- se mai folosesc pentru NAT si quality of service

10	deny	tcp	192.168.10.0	0.0.0.255	host	192.168.20.50	eq	ftp
20	deny	tcp	192.168.10.0	0.0.0.255	host	192.168.20.50	eq	telnet
30	permit	ip	host	192.168.10.0	host	192.168.20.50	eq	

### Ondimea regulii

- margele din 10 in 10 ca sa pot sa reviu sa mai adaugi intre
- ondimea e foarte importanta, pentru ca margele de sus im da sa avem match pentru regula. daca gaseste, aplica regula si se opreste din cauza
- daca nu se face match  $\Rightarrow$  denied (ultima regula e implicit Deny)

### Tipuri

#### Standard ACL

- numbers: 1-99
- expanded mrs 1300-1999 ( $\Rightarrow$  mai multe ACL / device)
- folosint doar source addrs. sa identifice traficul

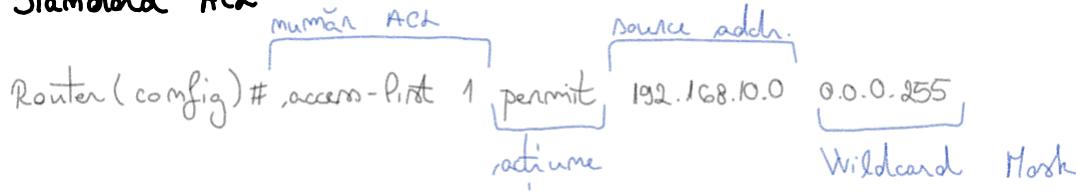
#### Extended ACL

- numbers 100-199
- expanded mrs. 2000 - 2699
- identifica traficul prin source addrs, destination addrs, protocol si port number

#### Named ACL

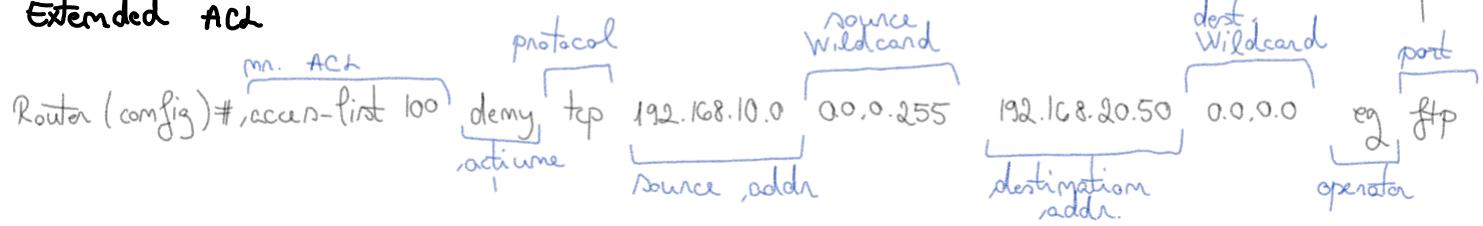
- permite ca standard si extended ACLs sa primeasca nume ca sa fie mai usor de gestionat (ex: cum ar fi mai multe ACLs pe un device, sa stiu care face ce se ocupă)

## Standard ACL



se pot folosi m. sau cuv.  
chiar;  
de ex: ftp imprezinta 21

## Extended ACL



## Named ACL tipul: standard/extended

Router (config) # ip access-list extended drama

Router (config-ext-macl) # deny ip 192.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0 eq 21

Router (config-ext-macl) # permit ip 192.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0

!!! ft importantă, analiză

## Wildcard Mask

- ca și o subnet mask inversată

0 = bits must match

1 - bits do not matter

### Adresa IP

192 168 10 0                    11000000 . 10101000 00001010 00000000

### Wildcard Mask

0 0 0 255                    00000000 . 00000000 00000000 11111111

→ match-wiești toate adresele ip între 192.168.10.0 și 192.168.10.255

## Port Operator

gt = Greater Than

lt = less Than

neq = Not Equal

eq = Equal

range = Range Specified

Extended IP access list 101

```
10 deny tcp 192.168.10.0 0.0.0.255 host 192.168.20.50 eg ftp
20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.20.50 eg telnet
30 permit ip host 192.168.10.0 host 192.168.20.50
```

Dacă Wildcard Mask are 32 de biți / e 0.0.0.0 , putem să îl scriem ca Keyword-ul "host"

Standard IP access list 114

```
10 permit host 192.168.10.10
20 permit host 192.168.10.15
30 permit host 192.168.10.20
```

Extended IP access list 114

```
10 permit tcp any host 192.168.20.50 eg www
20 permit tcp any host 192.168.20.50 eg ftp
```

Potem folosi Keyword-ul "any" , pentru a fi acceptată orice adresă IP

## API

(Application Programming Interface)

= "interfață" către o aplicație

### HTTP Methods

**POST**

**GET**

**PUT**

**PATCH**

**DELETE**

### CRUD

**CREATE**

**READ**

**UPDATE (replace)**

**UPDATE (modify)**

**DELETE**

exemplu:

aplicație mobilă  
care văză vremea



aplicație pt. vreme, cu fl. multe  
pt. buturi resurse

1: `https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}`

unic, al tău, ca aplicația să fie  
evidențiată, ceea ce face să nu î  
spămăldă

2: "main":  
"temp": 9.53,  
"feels-like": 7.62,  
"temp\_min": 7.78,  
"temp\_max": 10.56,  
"pressure": 1016,  
"humidity": 61

y

De încrezut [developers.google.com/youtube/v3](https://developers.google.com/youtube/v3)

↳ e interesant, poti de ex sa vezi cate like-uri, videodisponibile, subscrise etc. sau cum cam

## IP Addresses și NetMask

### IP Address (IP)

- 32 de bîta  $\Rightarrow$  4 octeta

00000001.00000010.00000011.00000100 Baciu mai ușor  $\rightarrow$  1.2.3.4  
 1 2 3 4

### Network Mask (NetMask, Mask)

- 32 de bîta  $\Rightarrow$  4 octeta

11111111.11111111.11111111.00000000  $\rightarrow$  255.255.255.0  
 24 de 1                  8 de 0  
 ↓                        ↓  
 124                      2<sup>8</sup> adrese IP în rețea

11111111.11111111.11111111.11000000  $\rightarrow$  255.255.255.192

26 de 1  $\Rightarrow$  /26

6 de 0  $\Rightarrow$  2<sup>6</sup> de adrese IP în rețea

11111111.11111111.11110000.00000000  $\rightarrow$  255.255.240.0

20 de 1  $\Rightarrow$  /20

12 de 0  $\Rightarrow$  2<sup>12</sup>, adrese IP în rețea

- amelioră diferențele dintre adresele IP ale device-urilor din aceeași rețea

Tot internetul: [0 0 0 0  $\rightarrow$  255 255 255 255]  $\rightarrow$  2<sup>32</sup> adrese IP în Internet  
 NM = 0 0 0 0

NetMask-urile împart internetul în subrețele (intervale)

NM = 11 10 000  $\rightarrow$  2<sup>x</sup> IP

[Start IP . . . End IP], range = 2<sup>x</sup>  
 ↓                        ↓  
 Network Address (NA) Broadcast Address (BA)

### Adresa de rețea

- adresa IP care identifică în mod unic o rețea
- permite adresa IP
- 192.168.1.10 cu masca 255.255.255.0, adresa de rețea e 192.168.1.0
- întotdeauna pară (broadcast = impară)
- nu se poate folosi

$$\begin{array}{r} 11000000 = 2^7 + 2^6 = \\ 128 + 64 = 192 \end{array}$$

$$\begin{array}{r} 11111111.11111111.11110000.00000000 = 2^7 + 2^6 + 2^5 + 2^4 = \\ 128 + 64 + 32 + 16 = \\ 192 + 48 = 240 \end{array}$$

Nu putem spune că o adresă IP e adresa de rețea fără să stăm netmask-ul!

## Network Address (NA)

$$NA = IP \text{ AND } NM$$

$$\begin{array}{r} \text{AND cu 1 în 0} \\ abcdefgh \text{ AND} \\ \hline 11111111 \\ abcdefgh \end{array}$$

## Broadcast Address (BA)

$$BA = IP \text{ OR } (\text{NOT } NM)$$

$$\begin{array}{r} \text{abcdefg AND} \\ 00000000 \\ \hline 00000000 \end{array}$$

Ex 1:

$$IP = 10.11.12.13$$

$$NM = 255.255.255.0 \quad /24$$

$$32-24=8 \Rightarrow 2^8=256 \text{ de IP-uri}$$

$$[NA \dots BA] \quad măre = 256$$

NA

$$\begin{array}{r} 10.11.12.13 \text{ AND} \\ 255.255.255.0 \\ \hline 10.11.12.0 \end{array}$$

BA

$$\begin{array}{r} \text{NOT } NM = 0.0.0.255 \\ 10.11.12.13 \text{ OR} \\ 0.0.0.255 \\ \hline 10.11.12.255 \end{array}$$

$$\Rightarrow IP = 10.11.12.13$$

$$NM = 255.255.255.0$$

$$[10.11.12.0 \rightarrow 10.11.12.255], \quad măre = 256$$

Ex 2:

$$IP = 10.11.12.13$$

$$NM = 255.255.255.1111000 \quad /29 \Rightarrow 2^5 \text{ IPs}$$

$$NM = 255.255.255.1111000$$

$$NA = 10.11.12.00001101 \text{ AND}$$

$$255.255.255.1111000$$

$$\hline 10.11.12.00001100$$

$$NA = 10.11.12.8$$

$$\text{NOT } NM = 0.0.0.00000111$$

$$BA = 10.11.12.00001101 \text{ OR}$$

$$0.0.0.00000111$$

$$\hline 10.11.12.00001111$$

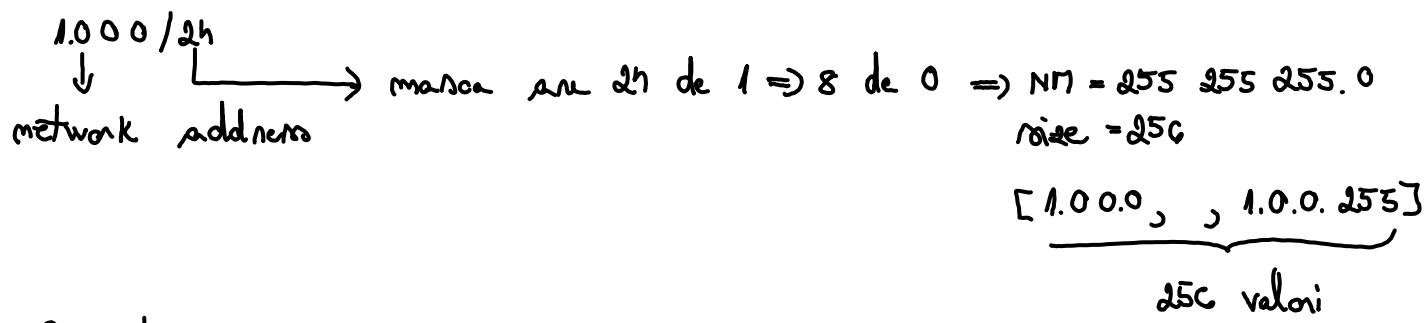


$$BA = 10.11.12.15$$

$$\Rightarrow IP = 10.11.12.13 \quad \text{și} \quad NM = 255.255.255.1111000 \quad /29$$

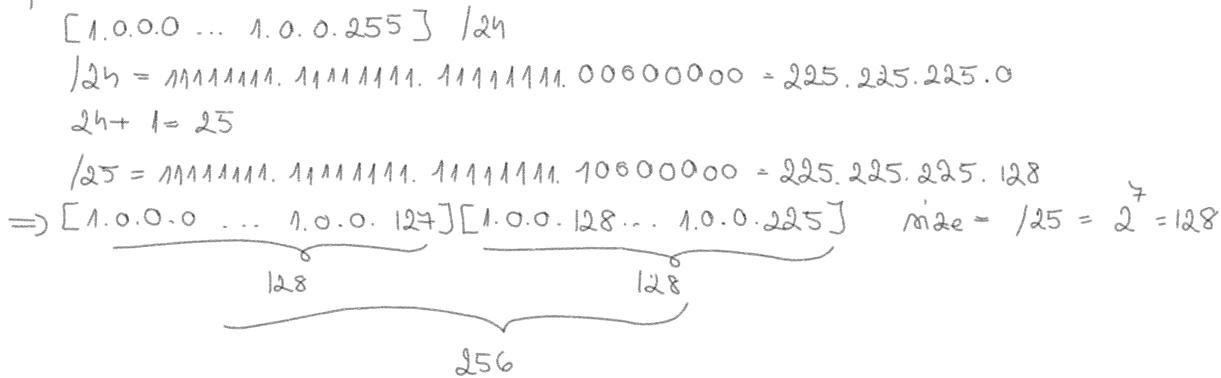
$$[10.11.12.8 \rightarrow 10.11.12.15], \quad măre = 8$$

## Network Splitting



Impărtirea:  $[NA \dots BA] \Rightarrow [NA_1 BA_1][NA_2 \dots BA_2]$   
 $NM+1$

Exemplu:



Cost general NA și BA:

m devices  $\rightarrow$  m IP addrs.

NA și BA nu se pot folosi pe dispositiv  $\Rightarrow$  m+2

mac -  $2^x \Rightarrow m+2 \leq 2^x$

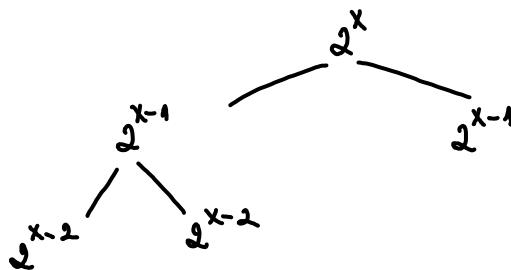
NM =  $/(32-x)$ , unde 11.10 00 avem x zero-uri în 32-x de 1

Cost general splitting:

1 10 0 x de 0

1 110 . 0 x-1 de 0

1 110 0 x-2 de 0




---

[NA	.	.	BA]	NM
[NA <sub>1</sub> . BA <sub>1</sub> ][NA <sub>2</sub> . BA <sub>2</sub> ]	.	.	BA <sub>2</sub> ]	NM+1
[NA <sub>1</sub> . BA <sub>1</sub> ][NA <sub>2</sub> . BA <sub>2</sub> ][NA <sub>3</sub> . BA <sub>3</sub> ][NA <sub>4</sub> .. BA <sub>4</sub> ]	.	.	BA <sub>4</sub> ]	NM+2

Exemplu:

NA (Network IP Address) = 82.228.39.0

NM (Mask) = 225.225.225.0 (/24)

Subnetworks: N1: 40 IPs

N2: 40 IPs

N3: 16 IPs

N4: 20 IPs

N5: 4 IPs

N6: 3 IPs (între 3 routere)

N7: 2 IPs (-II-2)

N8: 2 IPs (-II-)

N9: 2 IPs (-II-)

N10: 2 IPs (între un router și un wireless router)

$$40+3 = 43 < \underline{64} = 2^6 \Rightarrow 6 \text{ de } 0 \Rightarrow 32 - 6 = /26$$

$$40+3 = 43 < \underline{64} = 2^6 \Rightarrow /26$$

$$16+3 = 19 < \underline{32} = 2^5 \Rightarrow /27$$

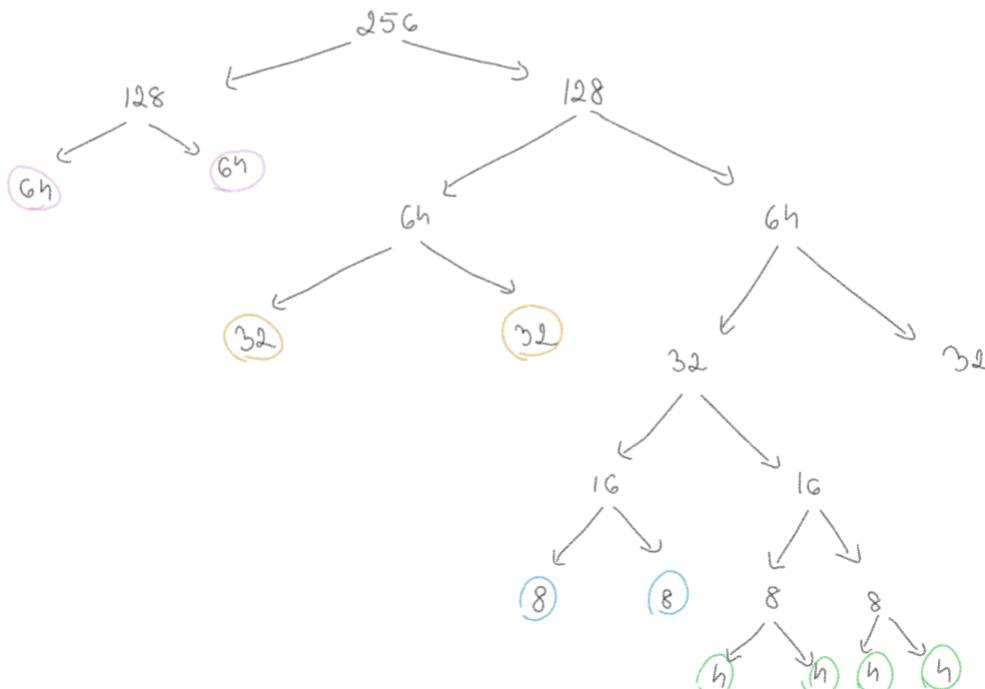
$$20+3 = 23 < \underline{32} \Rightarrow /27$$

$$4+3 = 7 < \underline{8} \Rightarrow /29$$

$$5+2 = 7 < \underline{8} \Rightarrow /29$$

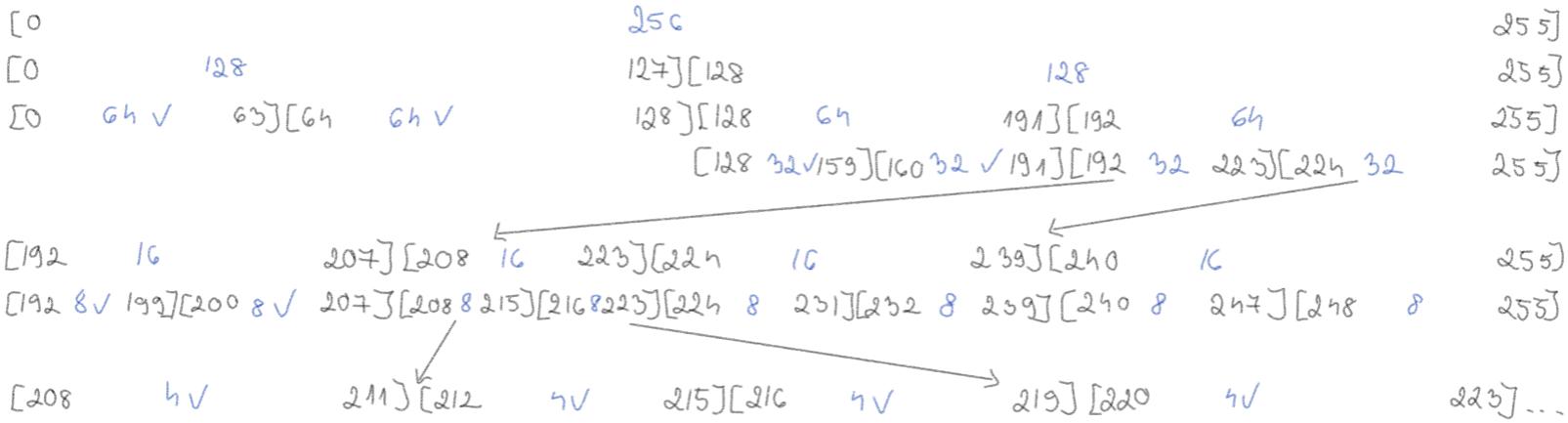
$$2+2 = 4 \leq \underline{4} \Rightarrow /30$$

Verificare:  $2 \times 64 + 2 \times 32 + 2 \times 8 + 4 \times 4 = 224 < 256 \quad \checkmark$



Rezultat → m devices (IP) + 1 router + 1 NA + 1 PTA ⇒ m+3  
 Astă multă trebuie +3, că suntem doar 3 routere să conectăm rețelele între ele  
 $\Rightarrow + NA + PTA = m+2$

82.228.39.0 /24



- N1: 82.228.39.0/26
- N2: 82.228.39.64/26
- N3: 82.228.39.128/27
- N4: 82.228.39.160/27
- N5: 82.228.39.192/29
- N6: 82.228.39.200/29
- N7: 82.228.39.208/30
- N8: 82.228.39.212/30
- N9: 82.228.39.214/30
- N10: 82.228.39.220/30

- NM: 255.255.255.192
- NM: 255.255.255.192
- NM: 255.255.255.224
- NM: 255.255.255.224
- NM: 255.255.255.248
- NM: 255.255.255.248
- NM: 255.255.255.252
- NM: 255.255.255.252
- NM: 255.255.255.252
- NM: 255.255.255.252

Dispozitivele vor primii IP-uri  
începând cu prima valoare disponibilă (necpermă că NA și SA să fie  
pe aceeași adresă)

- ex noutenul dim N1 82.228.39.1  
urm. disp. dim N1: 82.228.39.2  
etc  
noutenul dim N7. 82.228.39.161  
etc.

available: 82.228.39.224/29 NM: 255.255.255.248

## Router

- dispozitiv care conectează 2 rețele diferențiate
- redirecționează pachete de date folosind adresa IP atribuită fizicului dispozitiv într-un LAN
- primește pachete de date de la device-urile conectate la LAN și le redirecționează către și de pe internet către segmentele LAN, prim NAT

## Switch

- dispozitiv care poate conecta mai multe dispozitive într-un LAN
- mai inteligente decât hub-urile (decât noilele mi)
  - ↳ rețeaua deosebinte de ele, nu pot limita traficul de date către și de la fiecare port ( $\rightarrow$  fiecare dispozitiv are o capacitate suficientă de bandwidth)
  - ↳ latime de bandă

## Hub

- dispozitiv care poate conecta mai multe dispozitive într-un LAN
- cele mai simple dispozitive de rețea
- nu au capacitatea de a limita traficul de la și către fiecare port ( $\rightarrow$  toate dispozitivele conectate la hub împart aceeași latime de bandă)

## Default Gateway

- adresa IP a routerului care conectează un LAN la internet sau la altă rețea
- în general, e aceeași cu adresa IP a routerului
  - ↳ el directează datele între rețele