

## 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, este 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

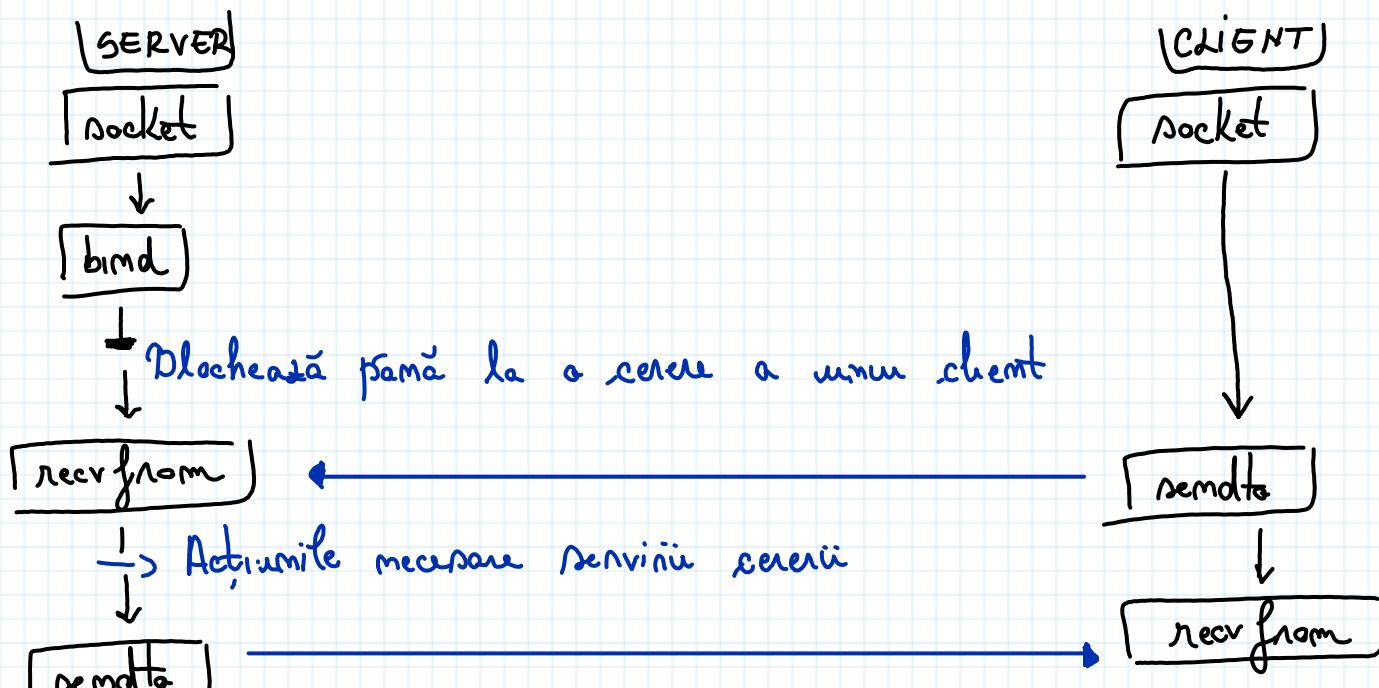
### Socket datagram

- 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-uri, fiecare dispozitiv sau proces are o adresă 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



### Apeluri sistem

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, clientAdres = s.recvfrom(buffer)

datele primite de la client (octete) de la tuplu (ip, port) octeti

### BUFFER

= zonă de memorie temporară utilizată pt memorarea și manipularea datelor

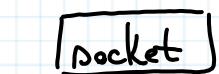
- discul, rețeaua, dispozitive hardware

poate avea mai multe adrese IP  
nu trebuie să fie ambele comunicări  
căruia IP să poată căuta

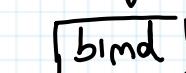
are un port efemer  
initială comunicarea  
tbd. să fie IP  
portul renunțului

## SOCKET STREAM

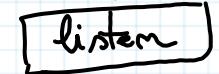
### SERVER



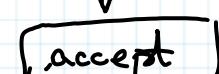
creare socket



port la care renunțul așteaptă să fie contactat

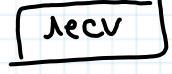


indica disponibilitatea de a primi cereri de conexiune pt  
socketul cu descriptorul dat și  
parametrul specific ca maxim de soliciting care pot fi  
blocate în timp ca se așteaptă ca renunțul să le accepte



Blochează până la o cerere a unui client

← STABILIRE CONEXIUNE (IP + PORT)



← DATE PT CERERE

↑ Actiunile necesare renunțului serverului

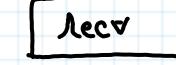
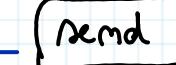


→ DATE DE RASPUNS

### CLIENT



Socket



### Apeluri sistem

IP = '0 0 0 0'

port = 8888

⇒ socket socket (socket AF\_INET, socket SOCK\_STREAM)

⇒ -1 → fail

⇒ bind ((IP, port))

⇒ listen (n)

⇒ connect ((IP, port))

clientSocket, clientAddr = ⇒ 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-urile

Când eșuează socketul?

- 1 Portul sau IP-ul e deja folosit
- 2 Lipsa permisiunilor
  - nu suntem portari sau IP
  - port < 1024 nu este folosit de procese de nulare
3. Rularea prea multor conexiuni simultan
  - depășirea limitelor de nerunză ale sistemului (limită de descriptori de fizici)
- 4 Setările incorecte ale firewall-ului
  - se blochează conexiunea
- 5 Probleme de netea
  - server indisponibil
- 6 Deactivarea protocolului necesar
  - ex: Încercarea utilizării unui socket IPv6 pe un port IPv4
7. Depășirea limitelor de nerunză ale sistemului
- 8 Erori de memorie sau depășirea bufferului

## MEDIU DE TRANSMISIE

### 1. Fibre 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 în nodule locale
  - mai multe moduri de propagare

### 3. Wireless Media

- unde radio
- infraroșii (lumini de vedere directă)

### 4. Satellite Communication

- sateliți geostationari
  - înălțime fixă deasupra Pământului
  - comunicare la nivel global (ex. tv)
- sateliți non-geostationari
  - 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

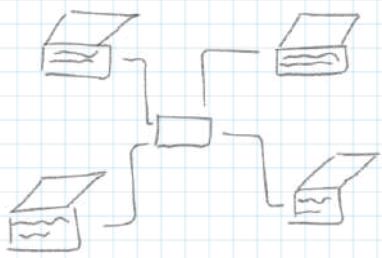
Cablu cross-over (se schimbă portocaliu cu verde) - dispozitive similare

- ,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



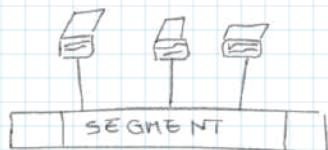
### 2 Extended star topology

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



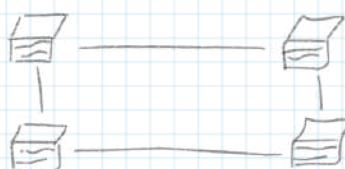
### \* 3 Bus Topology

- ↳ 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 destinație



## 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, restrânsă 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
- traficul 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
- viteza 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 transmisarea fiabilă a datelor în 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)

- transmitere 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 adrese 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 (router, switch-uri, server)
- permite administratorului să monitorizeze starea dispozitivelor și să facă modificări

## 11. SSH (Secure Shell)

- conexiunea securizată și criptată între 2 dispozitive

## Poșta electronică (e-mail)

- = serviciu de trimis / primire mesaje prin intermediul internetului
  - ↳ ex. Google, Yahoo, Microsoft etc
- utilizatorul primește și adresa de e-mail său ( poate trimite mail unic 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 autenticitatea proprietarului 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 scrie 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
- și **RFC** (Request For Comments). descriu de ce trebuie să facă comunitatea implementator, de client și server

## Tipuri de trafic

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

**Broadcast** - metodă de transmitere a informațiilor către toate dispozitivele dintr-un LAN

- comunicare 1:tot
- servicii ARP, RARP, DHCP (prin UDP)

↳ nu se poate prim TCP (ar fi mai mult pe același descriptor de socket)

**Multicast** - comunicare 1:n (nu primește totuști mesajul)

- ex. 2 peis dintr-un LAN intră pe un ntb → 2x trafic unicat între device și server
- ABCD
  - ↳ A = maxim 223
  - ↳ → ABCD validă

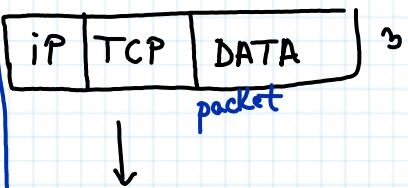
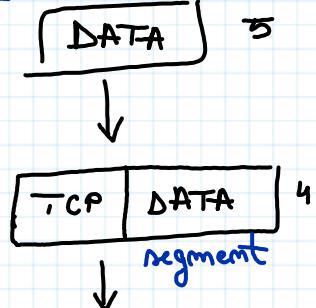
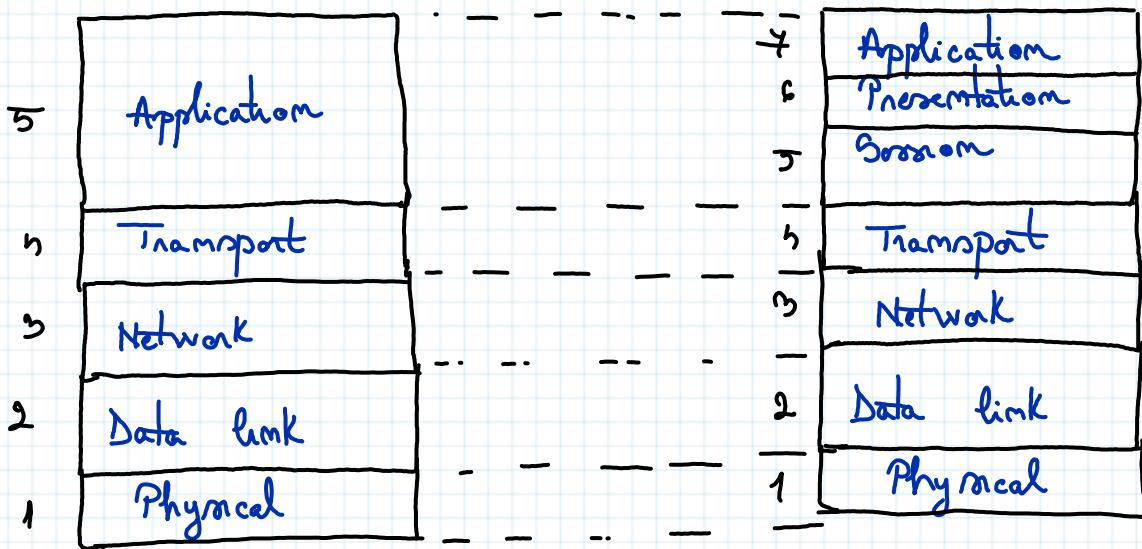
**Anycast** - comunicare 1:cel puțin 1 din mai multe

- 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

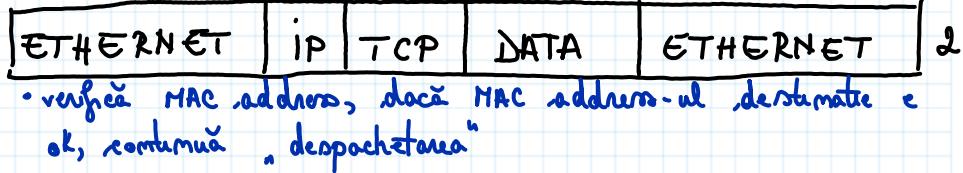
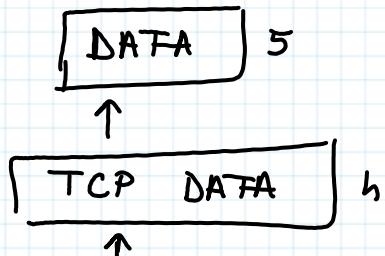
- = MAC de broadcast
- utilizat pentru comunicarea cu toate echipamentele dintr-un LAN

## TCP / IP model



- se adaugă protocolul
- pt fiecare protocol se trimiț informații specifice  $TCP = IP + port$

- verifică dacă ip dest e ok, sunt „despachetate” (procesarea)

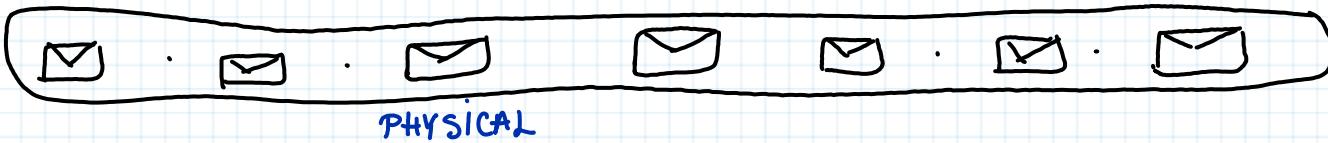


- verifică MAC address, dacă MAC address-ul destinat este ok, continuă „despachetarea”



destination and source MAC address

error check and make sure the data has been received correctly



## OSI model

OSI (Open System Interconnection) este un cadru conceptual utilizat pentru a descrie în întregime funcționalitatea rețelelor.

<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><b>Application (7)</b></p>
<p>Se ocupă cu formatarea datelor și le face compatibile cu dispozitivele în aplicările din rețea. Comprimare, criptare și alte operații de prelucrare a datelor.</p>	<p>Format Data, Encryption</p>	<p><b>Presentation (6)</b></p>
<p>Stabilire, menținere și închidere semnificative de comunicare între dispozitive</p>	<p>Start &amp; Stop Session</p>	<p><b>Session (5)</b></p>
<p>Asigură comunicarea fizică și controlul flexibil între dispozitive (protocole)</p>	<p>TCP, UDP, Port Numbers</p>	<p><b>Transport (4)</b></p>
<p>Routarea datelor între rețele dif. Utilizarea adrese IP și a unui pește de către destinație corectă.</p>	<p>IP Address, Routers</p>	<p><b>Network (3)</b></p>
<p>Transmiterea datelor între dispozitive conectate la același LAN. Se adăugă MAC addresses și se efectuează verificarea de erori și asigură transmiterea fizică.</p>	<p>Mac Addresses, Switches</p>	<p><b>Data link (2)</b></p>
<p>Componente fizice ale rețelei (cablu, conector, semințe electrice sau optice). Se ocupă cu transmiterea bruto a datelor.</p>	<p>Cable, Network Interface, Cards, Hubs</p>	<p><b>Physical (1)</b></p>



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

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

4	Application
5	Presentation
6	Session
7	Transport
8	Network
9	Data link
1	Physical

## Port numbers

adresa URL  $\rightarrow$  adresa IP  $\xrightarrow{\hspace{1cm}}$  server

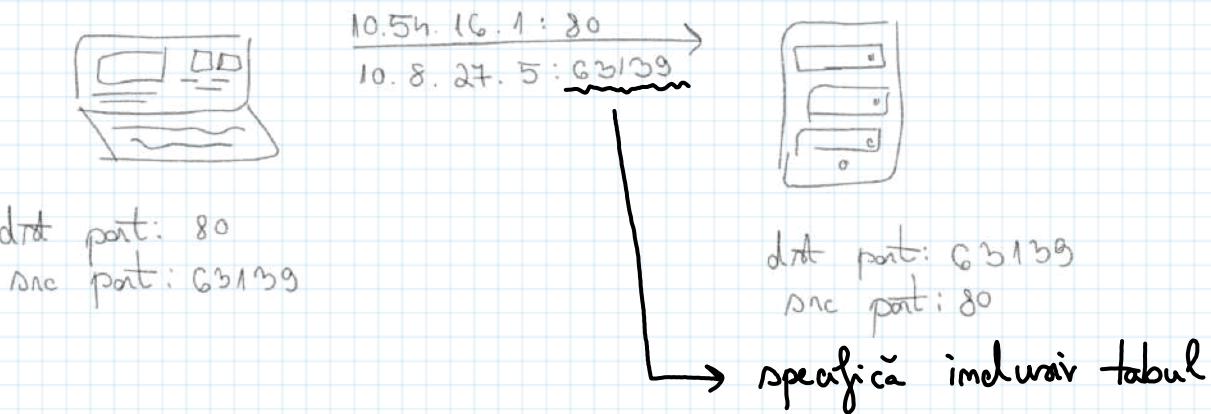
$\downarrow$   
se adaugă portul  
specific funcției  
de aplicatie

ex: 10.54.16.1:80

dst port: 80 (HTTP)

src port: 63139 (generated random)

Adresa IP date către computer  
Port date către aplicatie



## Porturi

0 - 1023  $\rightarrow$  well known

1024 - 49151  $\rightarrow$  registered

49152 - 65535  $\rightarrow$  dynamically assigned

## Adresa IP

- identificator unic assignat fiecarui dispozitiv conectat la o retea de calc

IP v4

ex: 192. 168. 32. 152

32 biti

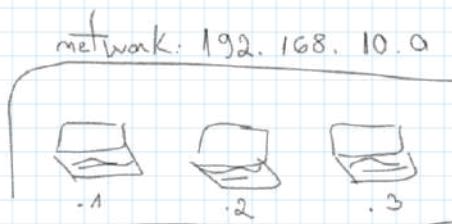
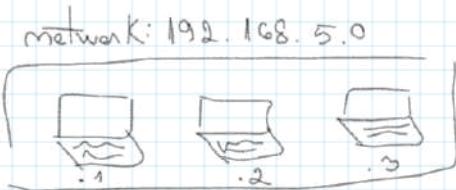
(retea)

(nr. casei)

- impartita in 2 prima parte NETWORK, a doua parte: HOST

Subnet mask 255.255.255.0

ex:



192.168.5.⑤ → host  
255.255.255.⑥

Class

Public

A 1000 - 126 255 255.255

subnet 255 0 0 0 → hosts: 16777216 hosts

B 128 0 0 0 - 191. 255. 255. 255

subnet 255 255 0 0 → hosts: 65536

C 192. 0 0 0 - 223 255 255. 255

subnet 255 255 255 0 → hosts: 256

PRIVATE

10 0 0 .0 - 10 255 255 255

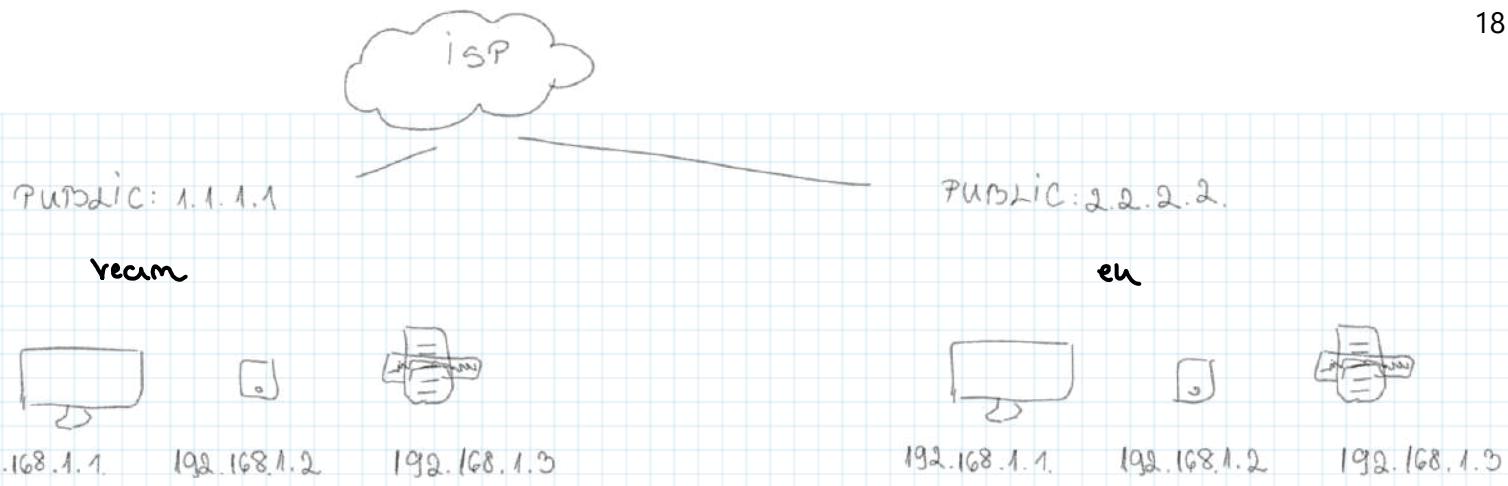
142. 16 0 0 - 142. 31. 255 255

192. 168 0 0 - 192. 168 225 225

D - multicast addresses

E - experimental use

Public - doar prim internet, trebuie sa fie unice



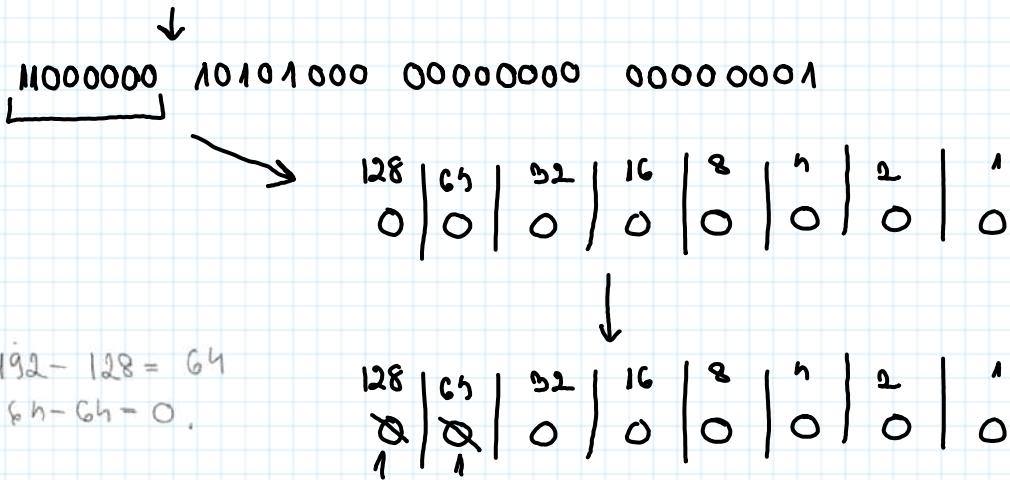
### Adrese IP private (false)

- trebuie să fie ușoare să se facă în propria rețea  
( $\hookrightarrow$  nu și în vecinul patern are același adresa IP)
- nu pot fi folosite pentru internet (să nu face un ghicirea de dupăcat)
- În general, la crearea contractului pentru internet primești o adresă IP
- avantaje permit economia de clase de addr. IP reale
- dezavantaje: trebuie SNAT ca să meangă internetul  
nu se pot rupe servicii pe care să fie accesibile din alte părți din internet fără DNAT
- nu sunt mutabile

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

## Binary

ex: 192 168 0 . 1



1-on  
0-off

Punem 1 în 0, adică  
încăt sănătatea să  
aducăm valoarea din  
căsuțele la care am  
pus 1, să ne dea  
mn. deosebit. Începem  
de la mijloc, cu cea  
mai mare valoare.

$$192 - 128 = 64$$

$$64 - 64 = 0$$

$$8 - 8 = 0$$

$$1 - 1 = 0$$

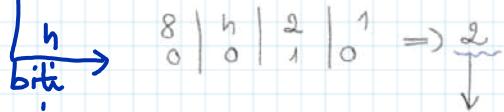
## IPv6 Addresses

IPv6 - 128 bits (4 octets)

IPv6 - 128 bits (8 hexadeci)

↳ separati prim:

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



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

- nu se mai folosesc subnet mask

host

2001:0db8:0000:0000:a111:b222:c333:abcd / 64

network

primii 64 biti = network

când vede că e incomplet, se completează automat cu 0

2001:0db8::a111:b222:0:abcd

- înlătură spatiul deasupra
- cu 0 - vici
- poate fi folosit o singură adresa pentru multe interface de la un singur host

pentru că nu se mai poate folosi prefixul cu 0, să nu se completeze automat cu restul

## Tipuri

### 1. Global Unicast

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

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

global prefix

subnet

host / interface ID

(64 biti)

- minim. 48 biti

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

### 2. Unique Local

- ca în una privată vă

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 și un 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 celor 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
- 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 rute)

### 2. MULTICAST

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

multicast prefix

- pentru aplicație / protocol  
- se trimite tuturor dispozitivelor,elor suscitate de cele cu aplicația / protocolul

### 3. BROADCAST

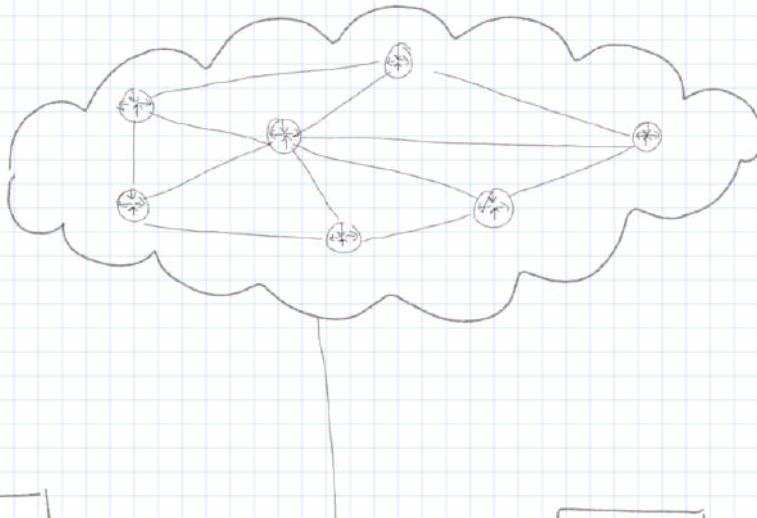
FF-FF-FF-FF-FF-FF

↳ se trimit tuturor dispozitivelor dintr-o rețea

## Moduli de rețea

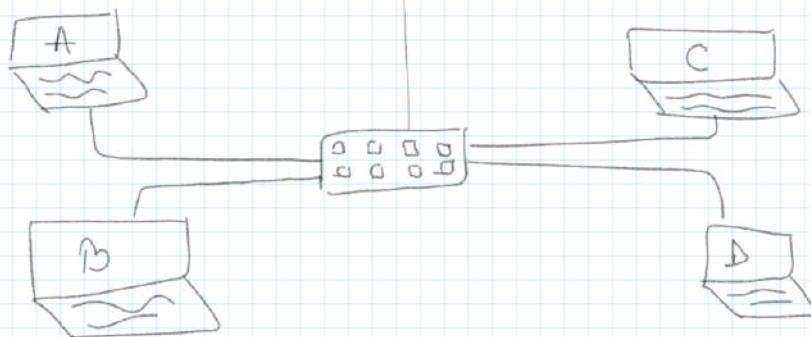
dimux / Apple 08 00 27: EC: 10 . G1  
 Microsoft: 08-00-27-EC-10-G1  
 Cisco 0800 27EC 10 G1

Router



layer 3  
 IP Addresses  
 → global communication

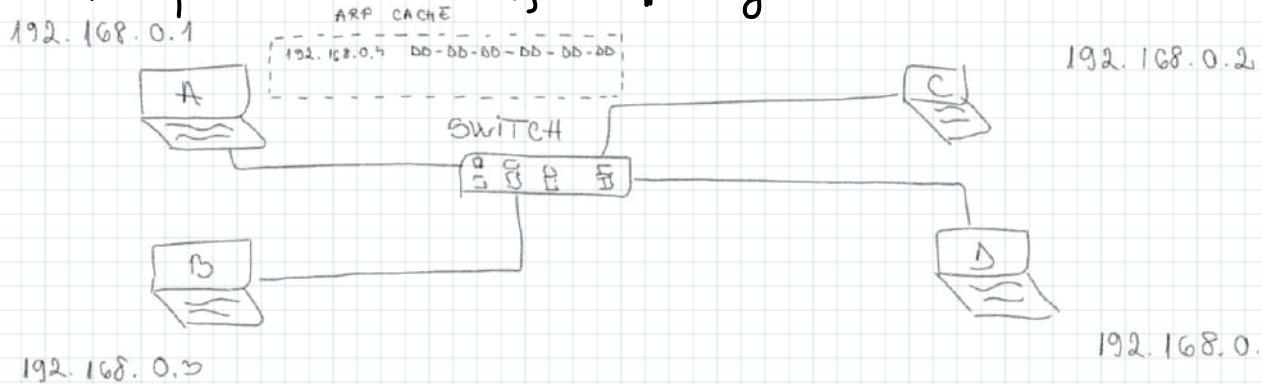
Switch



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

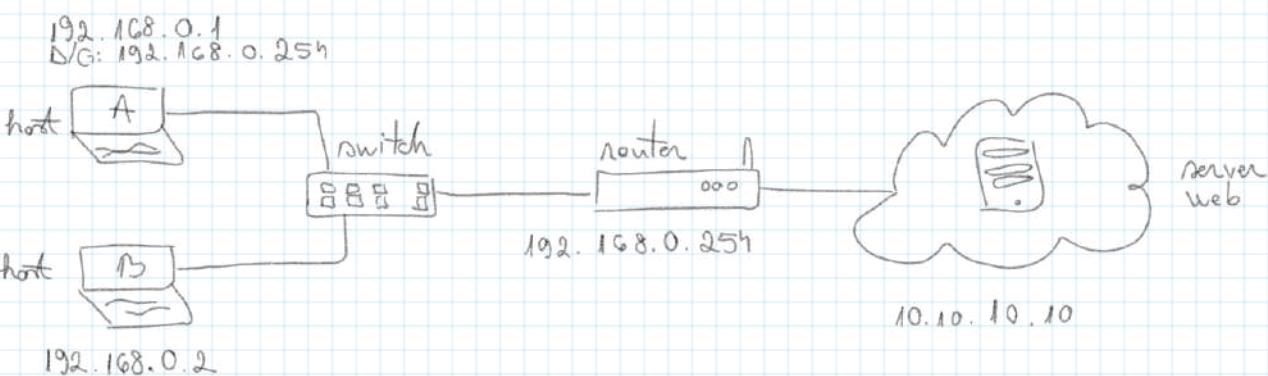
B și C nu reacționează

D își trimită înapoi lui A adresa MAC

Se va salva în ARP cache pentru viitoare utilizări

### Switch

- layer 2
- doar adr. MAC



A vrea să comunice cu serverul web

A verifică IP-ul serverului și vede că nu face parte din aceeași rețea

A trimisă cerere ARP în rețea: B refuză, routerul își răspunde cu adresa MAC (ca A să poată ieși din rețea)

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 = "una" 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 funcționare: dispozitivul trimite adresa MAC în solicită una IP  
• un server RARP răspunde cu adresa cerută
- ⇒ află adresa IP pe baza adresei MAC
  - ↳ doar pe rețea o cunoaște
- o fapt imlocuit de DHCP

## VLAN

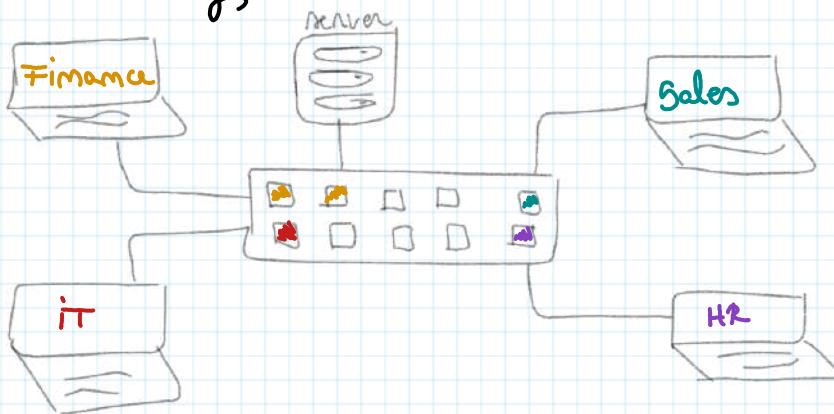
### Virtual Local Area Network

- separă virtual LAN-urile

### De ce să folosim VLAN?

#### Broadcast traffic

- se separă virtual rețeaua
- 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” aceluiși VLAN (finanțe și server pot communica, păcă să fac parte din același VLAN)

#### Mod de funcționare

- VLAN initial VLAN1

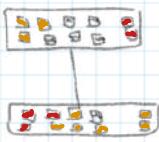
Toate interfețele pot communica între ele

- se pot adăuga maxim 4096 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 cu „stan” ce e un VLAN
- ⇒ comunicarea gestionează frame-uri

↳ gestionație de switch



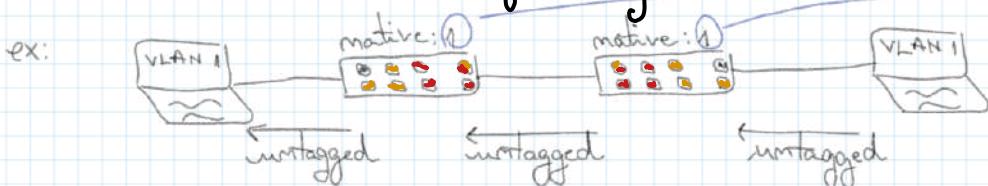
(preamble | sfld | dst. add | src. add | 802.1Q | type | data | fcs) → FRAME

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

## Native VLANs

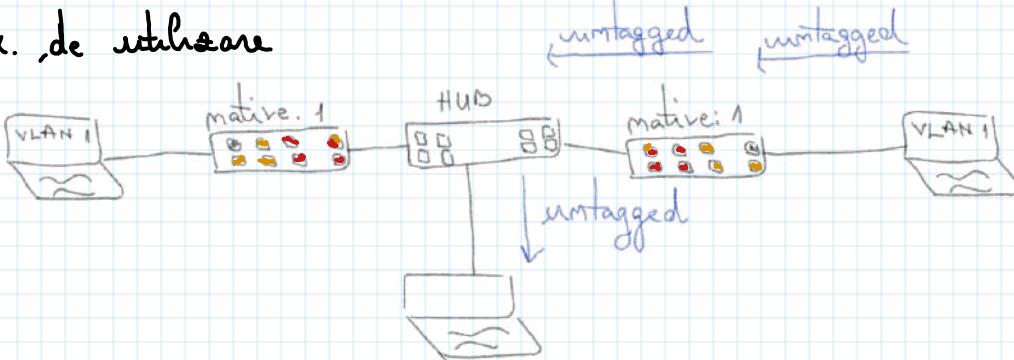
- trunk interface

- traversarea unui trunk fără tag



nu pot fi diferențiate  
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

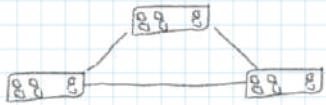
### Tipuri de STP

- 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

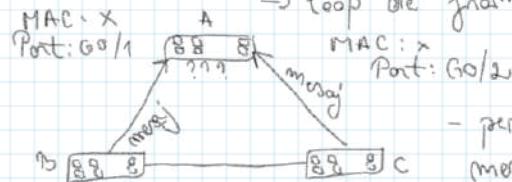
- prevene loop-urile, când sunt utilizate 2 sau mai multe switchuri

→ broadcast storm



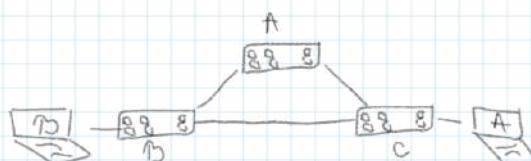
- de fiecare dată când se trim. un mesaj broadcast, se trimite către toate switch. ⇒ loop de frame-uri

→ unitable MAC Address Tables



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

→ duplicate frames



Host B → host A

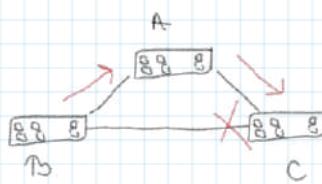
sw. B nu știe adresa lui h A

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

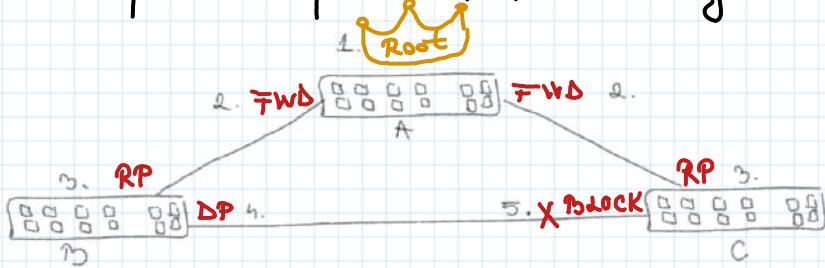
(⇒ trimite la totă lumea

- 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

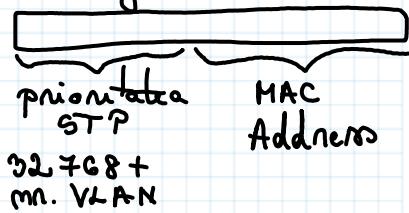
## Protocol

### 1. Root Bridge Election

- forward switch are user BPDUs

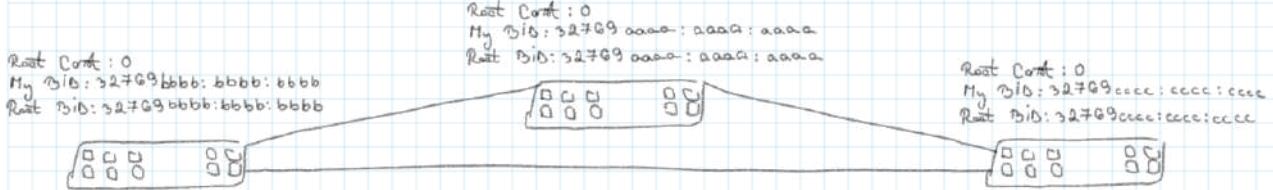
↳ root cost, route in local BID

↳ Bridge ID

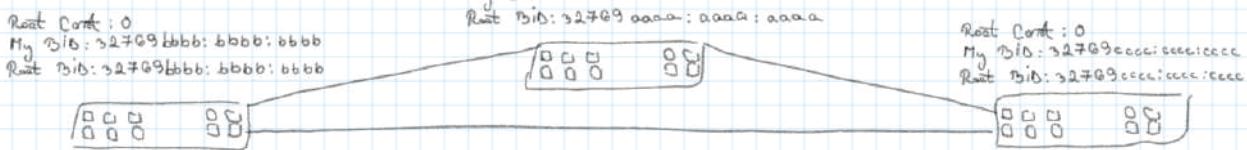


- devine root bridge switch-ul cu cel mai mic BID pe total

- la început, fiecare sw. re emite deosebită rădăcimă

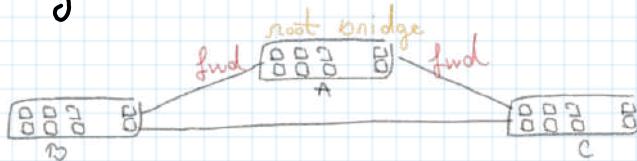


- apoi își trimit BPDU-urile între ele, iar cele care au BID mai mare re "conformă"



## 2. Root interface forwarding state

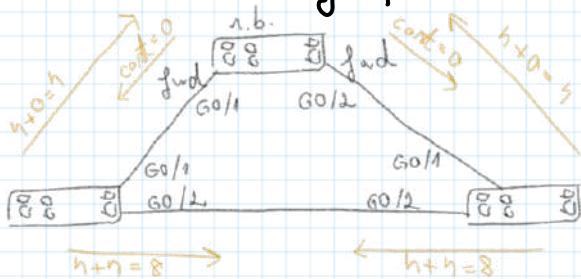
- toate porturile care se află în legătură directă cu rădăcina sunt în forwarding state



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

- se bazează pe cantitatea porturilor,

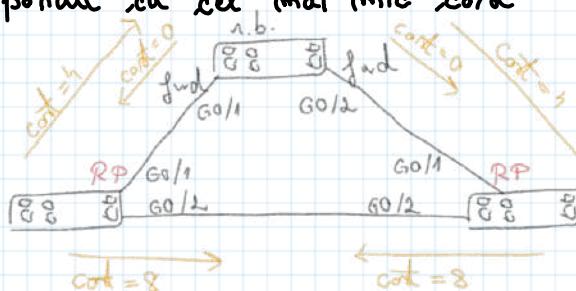
nr. porturilor care merg spre rădăcimă



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

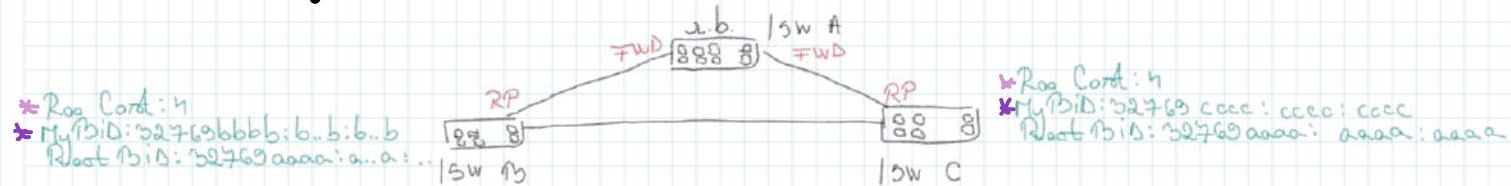
⇒



- dacă se fi făcut același sort 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

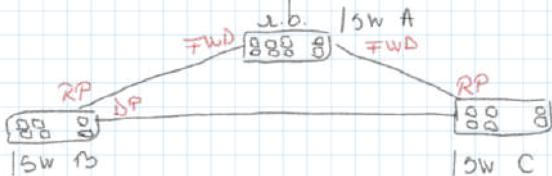
- se alege dintre cele care nu sunt repezis



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

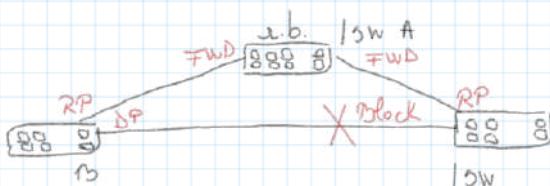
- se verifică cel mai mic cost \*
- se verifică cel mai mic BiD \*
- se verifică cel mai mic neighbor port priority
- se verifică cel mai mic neighbor port number

- în ex. de mai sus, SW B devine designated port



#### 5. Blocking

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



## Timers - Default

Hello 2 sec → intervalul de timp în care RB creează și trimite messages (arașă rătie totă „lumea” că funcțiile încă funcționează)

MaxAge  $10 \times \text{Hello}$  (20 sec.) → atât timp arată switchul că reacționează că ceea ce nu e ok

Forward delay 15 sec → timp între linklayer state și learning state

## STP states (ordinea efectuării)

Forwarding state → Blocking state → Listening  $\xrightarrow[rec]{15} \text{Learning}$   $\xrightarrow[rec]{15}$

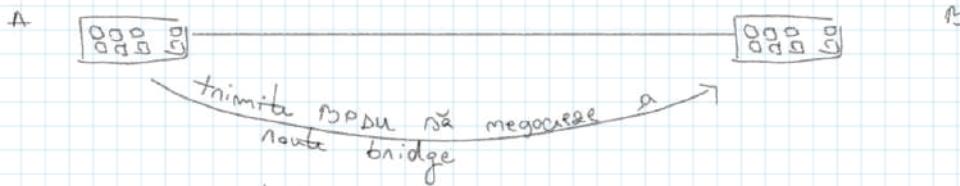
↳ durată multă → rezolvare rapid splicing tree protocol

↳ 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, scoti cablul și îl bagi la loc (processul durează multă, tu îl întreținuși și apoi tib. Nă încearcă 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 mediotice)

show spanning-tree summary (nu vedem dacă portfast/bpduguard e enabled / disabled)

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

spanning-tree bpduguard enable

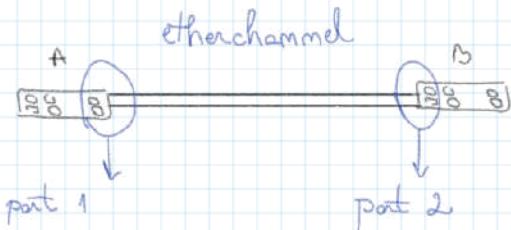
spanning-tree portfast bpduguard default

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 table

Se poate dubla capacitatea informațiilor.  
Dacă un cablu e rău / 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

- acces port / trunk port

același VLAN

deobicei

același allowed VLAN și native 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 una  
etherchannel

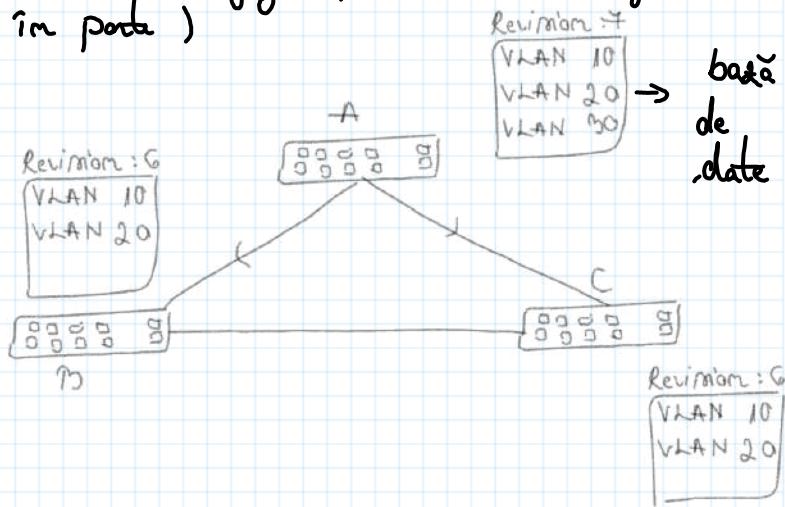
Channel Group = Port Channel = Etherchannel

## VLAN Trunking Protocol (VTP)

- le permite switch-urilor să își mențină și schimbe configurațiile VLAN (setul ar trebui introdus de către la fiecare în port)

### 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  
 Se fiecare 5 min se trimit de la fiecare sw. către toate restul summary advertizamente

Se verifică nr. revisionu. Dacă un nr. are newrev mai mare, în funcție de followers\* se trimit celalalte nr. subset advertisements (sabia acum se trimit informații despre VLAN - înainte să fi fost trafic de date între el).  
 Celalalte 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 baza de date a VTP

Clienț - nu poate crea VLAN-uri

- poate doar să trimită update-uri în adv. către baza 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 ignorează

- 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

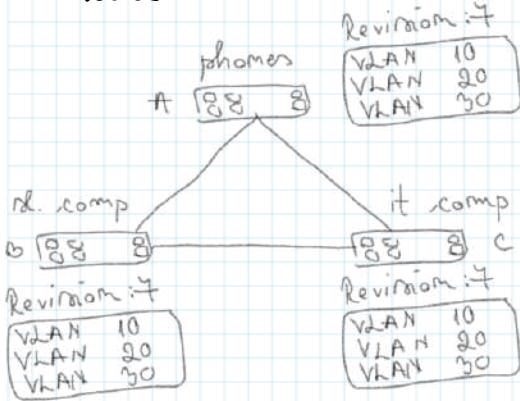
3. VTP password să fie la fel la toate  
→ este optimă

I) nu se trimite  
niciun mesaj pe  
porturi de acces

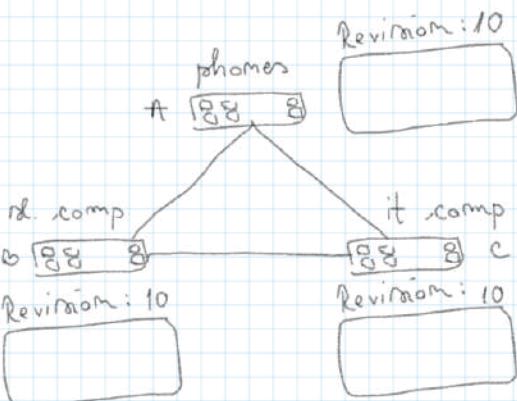
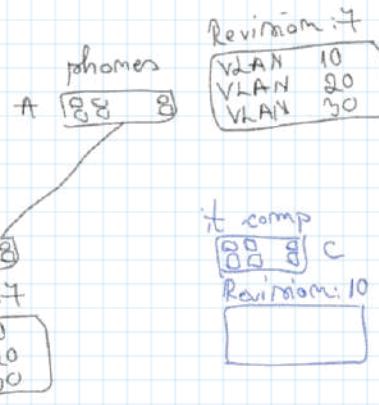
→ important pt. SW, ca să stie ce  
mesaje să ascute și pe care  
nu le ignore

## DEZAVANTAJ

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



deconectăm SW. C, facem experimente  
pe el, și ne ajunge la:



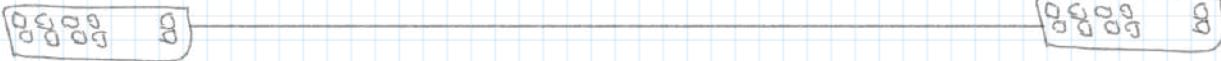
← astfel, când conectăm înapoi  
switch-ul, se vor actualiza.

## Exemplu comenzi

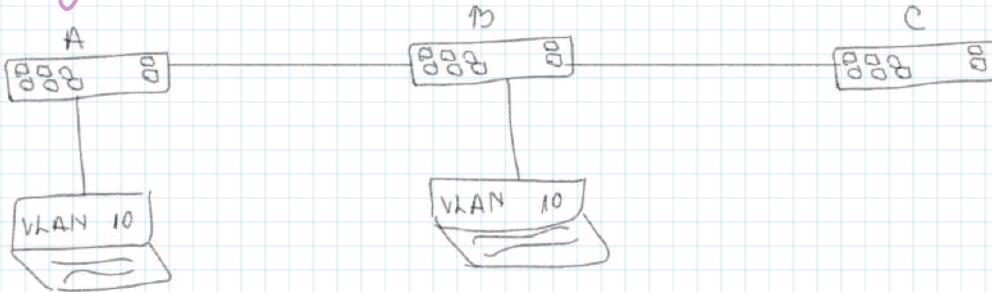
```
vtp mode server
vtp domain dama
vtp password dama
vtp pruning
interface fastEthernet 0/1
switchport trunk encapsulation dot1q
switchport mode trunk
```

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 nu „zică” pt ce VLAN -uri său porturi

⇒ înloc ca A și B să îi trimitem informații avusea lui C (care le-ar ignora), C „spune” să nu rețin că nu acceptă VLAN 10 ⇒ A și B nu îl mai trimit lui 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 tabelor 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 starea rețelei  
 $\Rightarrow$  noptenile pot lua decizii 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 unul mai specific  $\Rightarrow$  reduce complexitatea în dimensiunile tabelelor de rutare

## Rute implicite

- adresa IP a gateway-ului implicit

Tabelă de rutare = lista de rute disponibile peintru un router

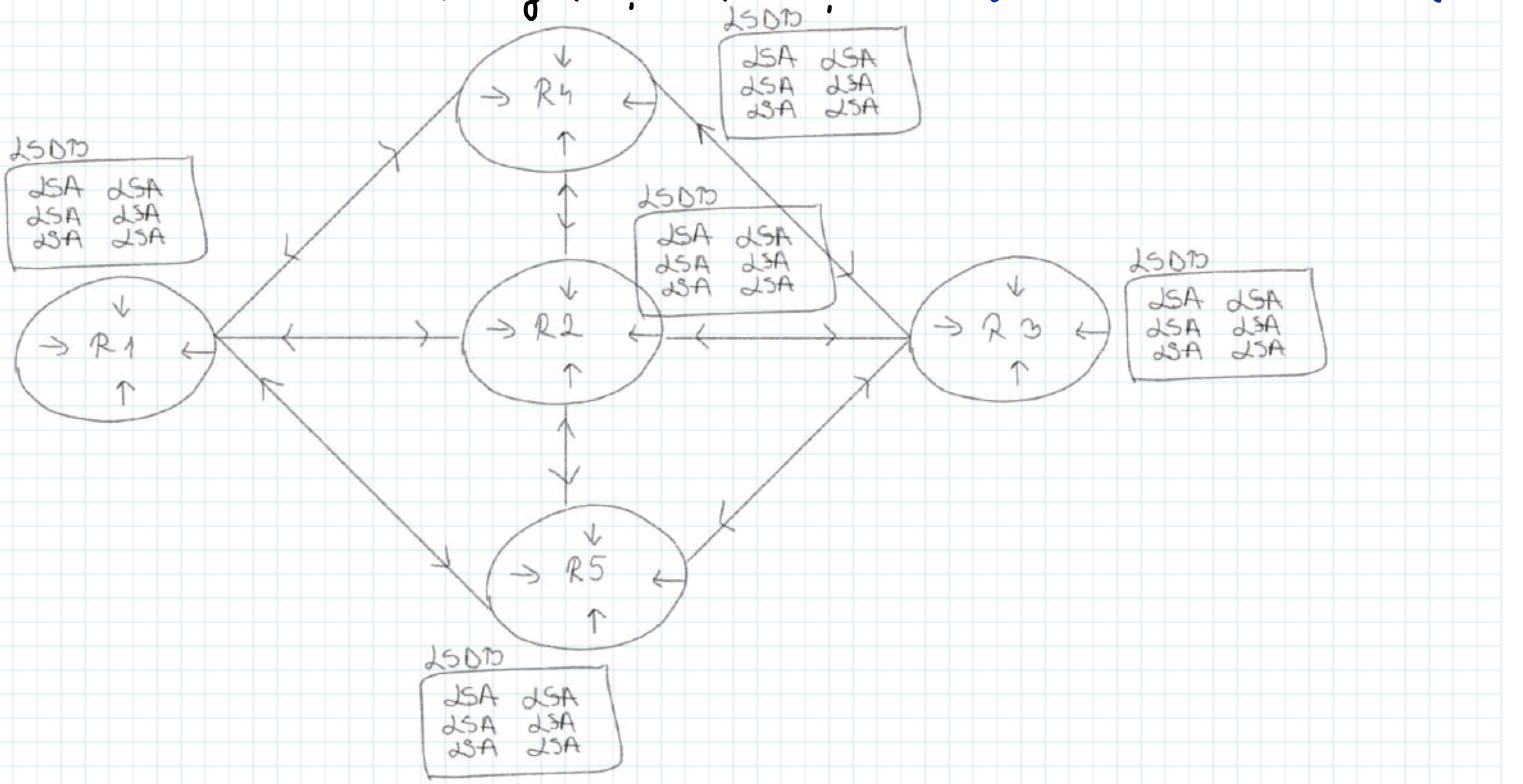
- informații despre rețele disponibile, adrese IP și metrice (=costurile) fiecărui nod

$\hookrightarrow$  minimă = conține numai 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

routing protocol → learning routers  
OSPF → learning about every router and subnet dim rețea  
↳ fiecare router sănătate informație despre rețea  
prin trimiterea link state advertisements (LSA)  
↳ toate informațiile rețelei în link state database (LSDB)



### Pasi

#### 1. Become neighbours

- 2 rute dim sănătate 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 router adaugă în routing table cele mai bune variante din funcție de informațiile dim LSDB

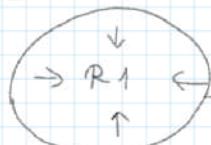
## Router ID (RID)

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

highest 'up' status loopback interface IP addn.

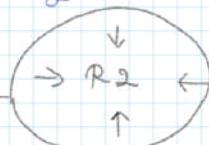
highest 'up' status mon-loopback interface IP addn.

Hello  
My RID: 1.1.1.1  
Neighbours:



Down

Requirements match?  
intervalul pînă în care să  
că ceva să fie ok, by default  
e 10 sec. pînă la Hello  
connecting links on the same subnet  
Area ID  
Subnet  
Hello and Dead interval  
Authentication (nă fie la fil)  
Sub-area flag  
Unique router ID

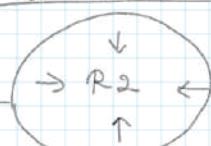


Down



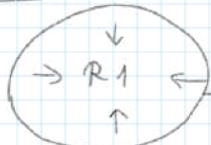
Down

Hello  
My RID: 2.2.2.2  
Neighbours: 1.1.1.1



Ymit

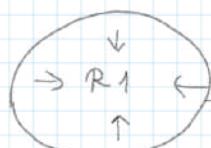
Hello  
My RID: 1.1.1.1  
Neighbours: 2.2.2.2



2-way

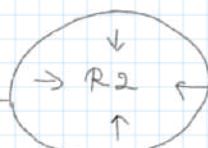


Ymit

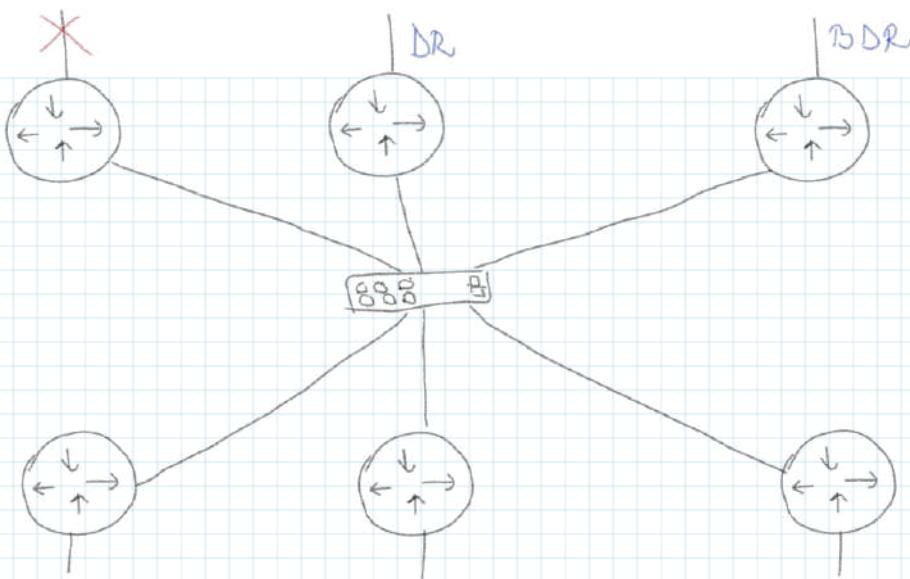


2-way

, acum sunt gata pentru schimb  
de informații



2-way



Designated router (DR)

- router care devine responsabil pt. gestionarea actualizării OSPF

Backup Designated Router (BDR)

- preia funcția de DR dacă aceasta este eșuată

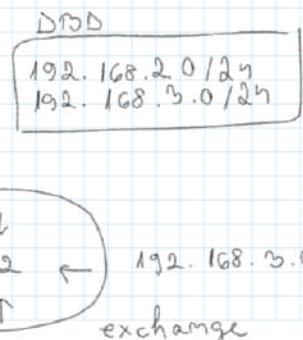
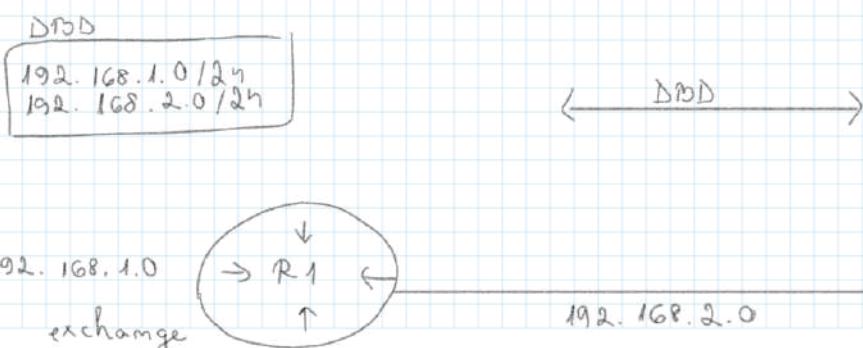
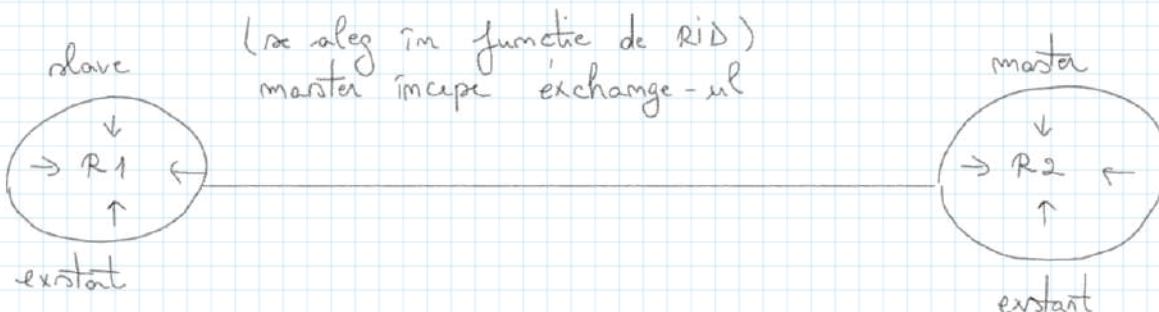
Pregătimem că un router eșuează. O să ne trimitem modificări ca mesajele la toti vecinii și apoi de la fiecare la fiecare.  $\rightarrow$  avem nevoie de DR și BDR.

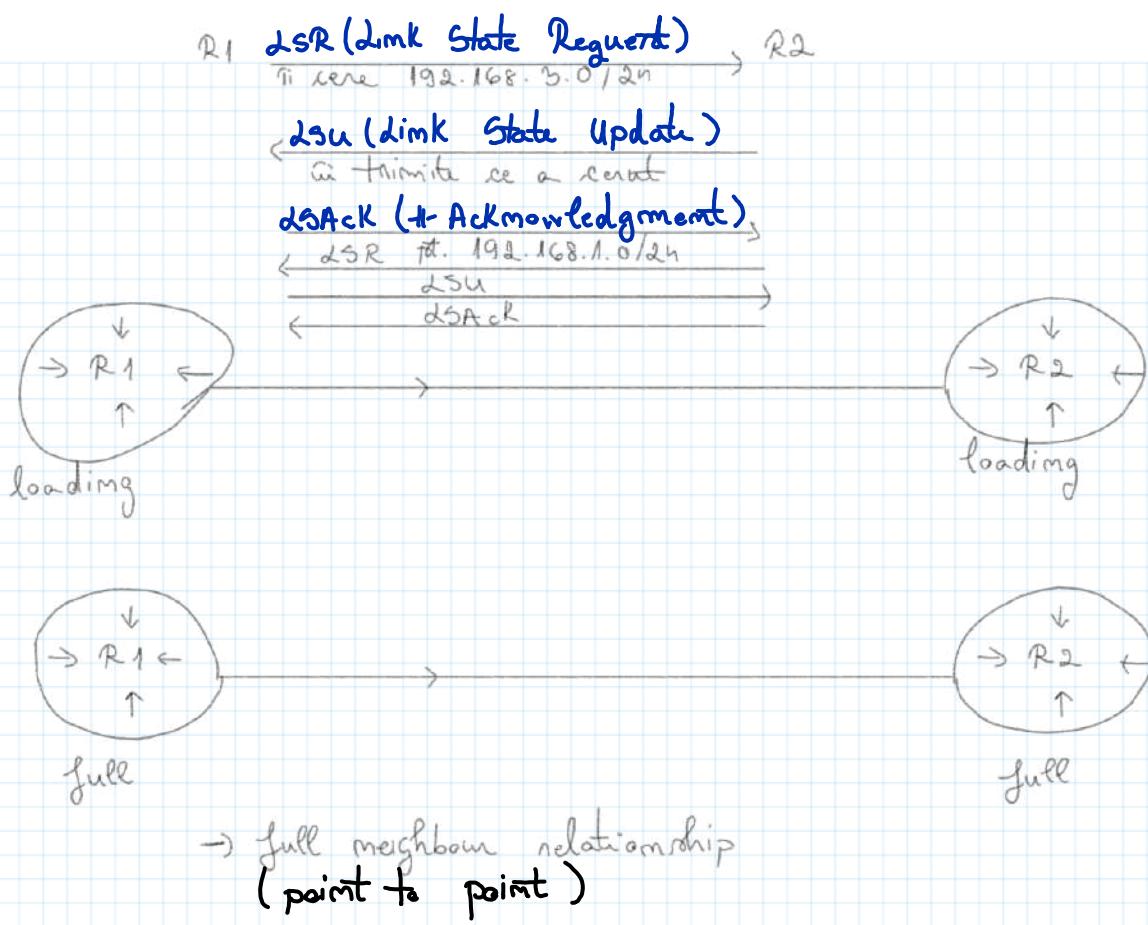
Această, dacă un router a eșuat, el trimite modificare la toti vecinii, dar toate routenele în afara de DR și BDR o ignoră. Astfel, se ocupă DR apoi să modifice routenele (care așteaptă ce să fie DR și BDR)

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 seama doar updateurile venite de la DR / BDR)





### Adăugarea celor mai bune rută 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  
 Kilobito/s

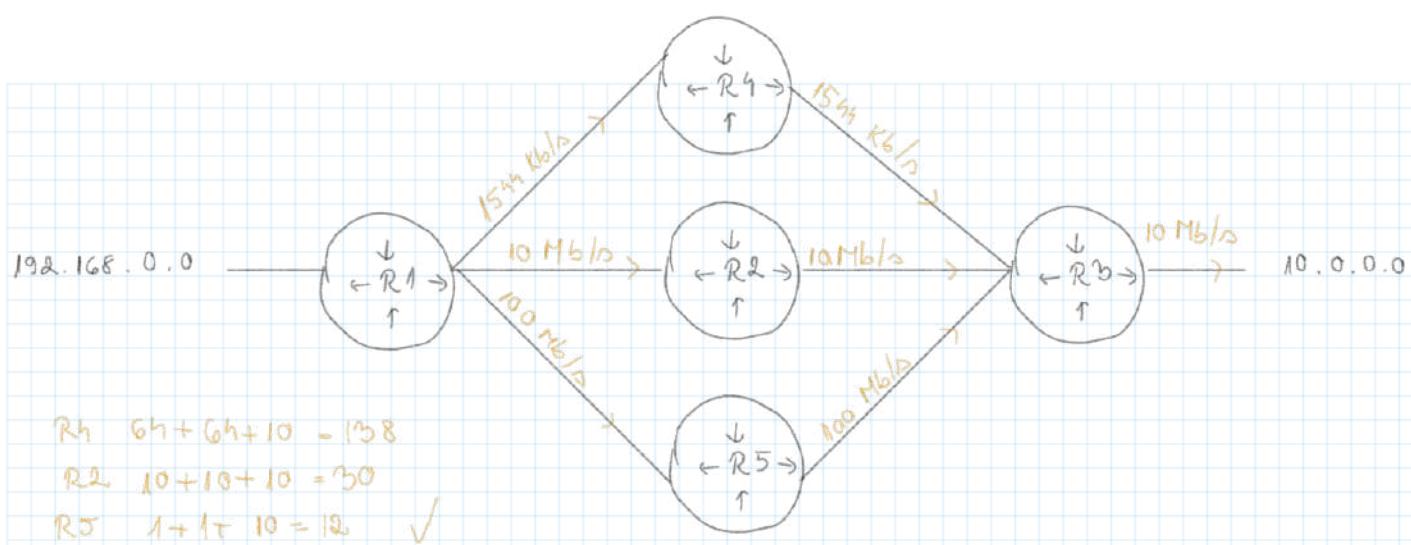
### Bandwidth

- = capacitatea de transmitere a datelor pînă într-un medium de comunicație
- se măsoară în bit/s sau multipluri de bit/s / secundă

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

### 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



$$R_1: 6h + 6h + 10 = 138$$

$$R_2: 10 + 10 + 10 = 30$$

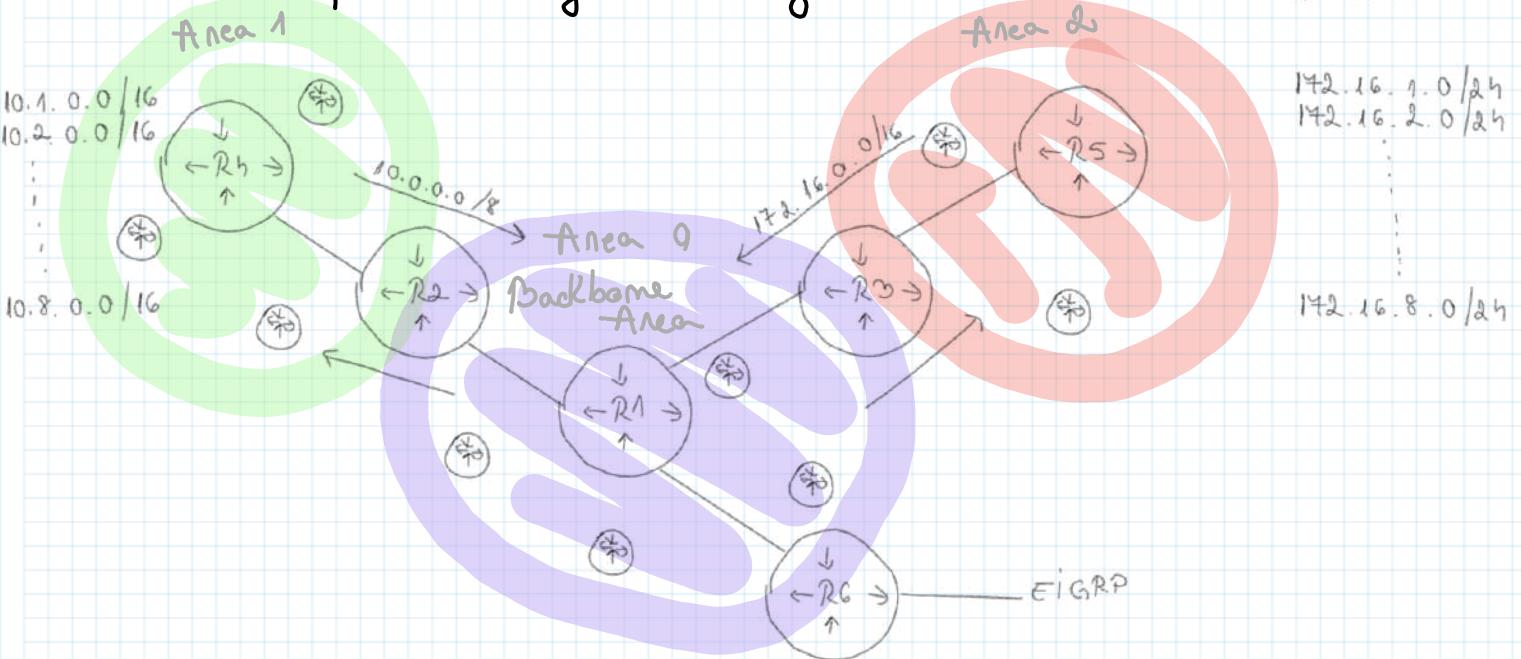
$$R_5: 1 + 1 + 10 = 12 \quad \checkmark$$

## OSPF Multi Areas

- reduce dimensiunea LSDB-ului
- "condensează" în routing tables
- update mesajelor într-o singură zonă

Area = grup de noutere

recomandat să fie max.  
50 de noutere



### 1. Area 0 (Backbone Area)

- toate celelalte nouturi să se conecteze la ea

### 2. Subnetting

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

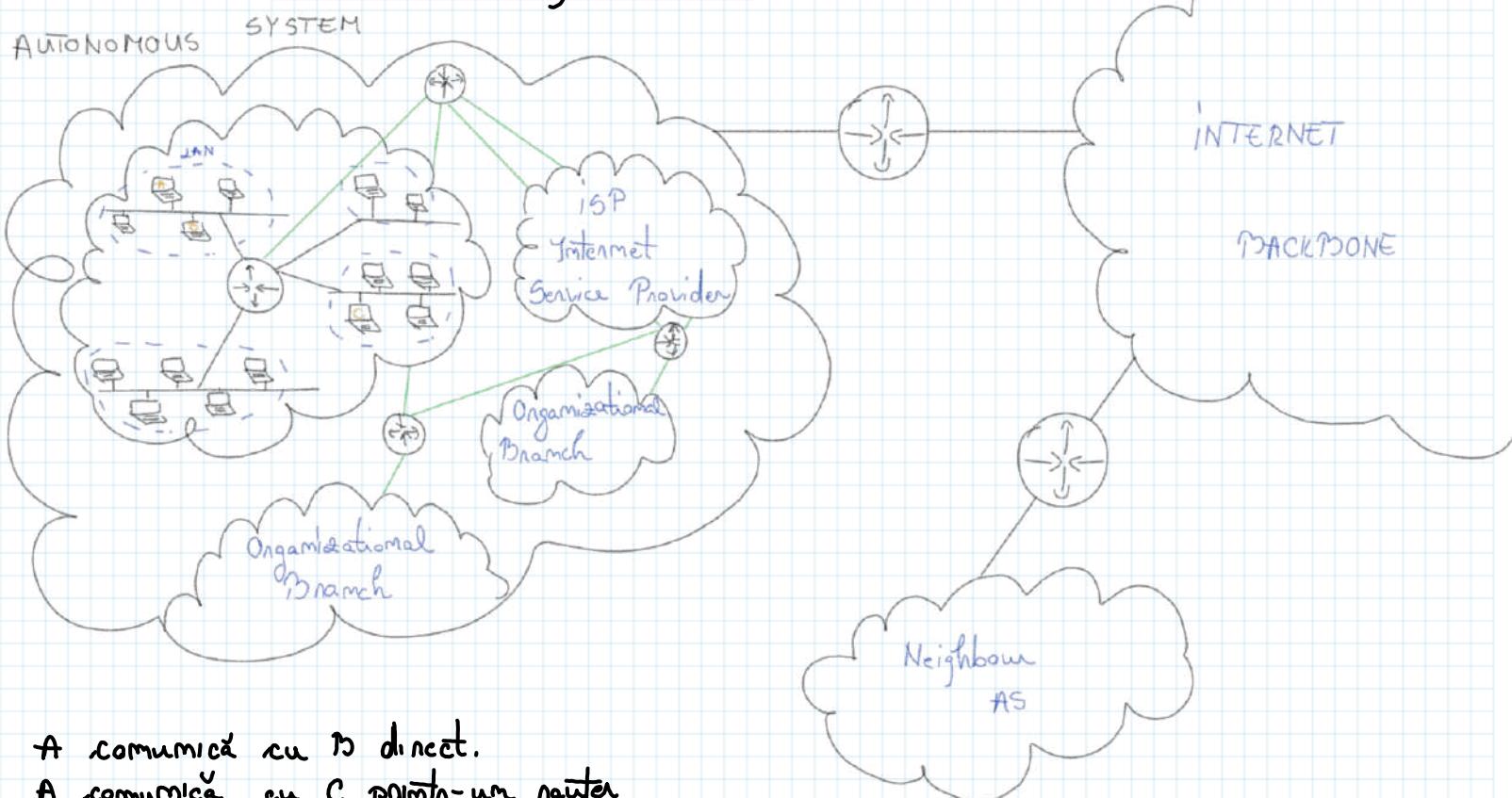
Backbone router R1, R2, R3

Area Border Router (ABR) R2, R3 → interfață

Internal router R4, R5 → într-untrarea cu alte zone

Autonomous System Boundary Router (ASBR) R6

## BGP (Border Gateway Protocol)



A comunică cu B direct.

A comunică 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)

comunicații TCP între vecini

informații despre rețea

actualizări periodice

Path Vector Protocol  $\Rightarrow$  informații se transmit împreună cu ruta în sine  $\Rightarrow$  evitarea buclelor în rutare BGP

Autonomous System

- colectie 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 provocate prin UDP (RIP advertisements / updates)

hop count: hop = traseu pînă la un router

rute mai bune = cele mai puține hopsuri

primesc actualizări prin broadcast și își actualizează tabelele de rutare  
valoare maximă: 15 (RIP v1)

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

split horizon: nu trimit informații despre rutele învăluite pînă la interfața primă  
aceeași interfață

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

actualizări + alte caracteristici  $\rightarrow$  convergență rapidă

- RIP v2 - subnetting
  - autentificare
  - adrese IP v6

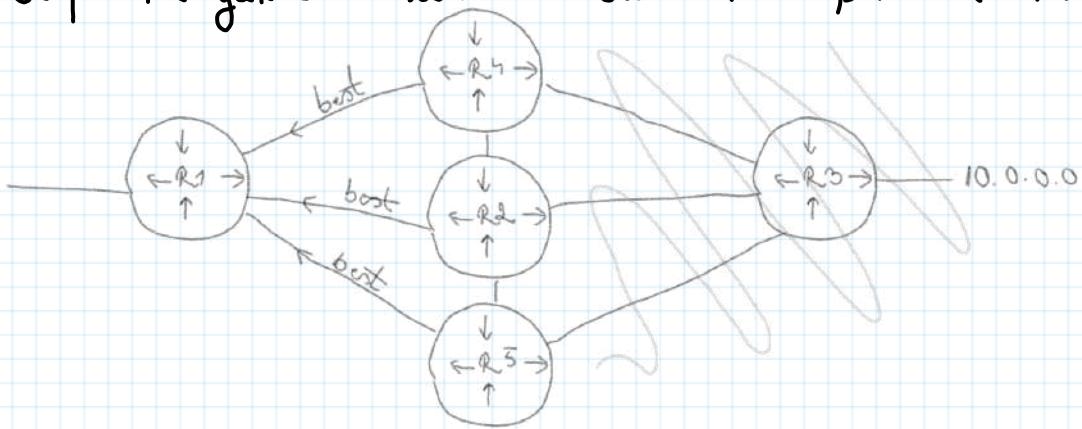
Placa de rețea - device  
fisice prin care se conectează calculatorul cu extinderile. Pe o placă  
fisică de rețea se pot pune mai multe adrese IP  $\rightarrow$  interfețe

## EIGRP

### (Enhanced Interior Gateway Routing Protocol)

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

Săptăm să găsească cea mai bună rută până la unicore router, din rețea



R1 săptăm să găsească rută până la router 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 până la destinație. Apoi, R1 tot amintește vecinii, din stânga în tot ară.

### 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 până crede că e „mort” celălalt, dacă nu trimit hello înapoi

Multicast - 227.0.0.10

Requirements match?

Hello și Hold intervalele nu trebuie neapărat să fie la fel!!!

Bz. fie același  
Autonomous System → AS number  
Number (ca să fie în configurația EIGRP pe rețea) → nă nu este router  
Subnet → nu este nevoie să fie pe același router  
K-values → nu este nevoie să fie pe același router (nu se potrivește pe rutele routelor)  
Authentication → dacă se folosesc, nu se potrivește



Dacă nu au îndeplinit condițiile ⇒ nu sunt vecini

## 2. Exchange routing information

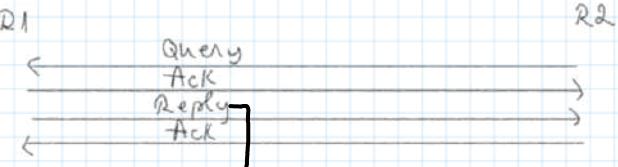
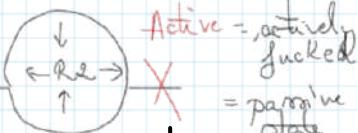
- nu foloseste UDP / TCP

- foloseste RTP (Reliable Transport Protocol)

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



După ce mai trimit update-uri doar dacă există modificări în rețea și în mod, nu se continuă cu hello.



dacă găsește  $\neq$  ack, dacă nu, trebuie eliminată din routing table

Dacă pe întâmplă arăta, se va petrece un Route Read Computation: noul rutează să caute o nodă liberă până la obiectiv  
Dacă nu găsește, îi întreabă pe vecini, dacă au ei o nodă

## 3 Choosing the best routes

### Metric Calculation Formula (Default)

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

value given to each outgoing link (microsec)

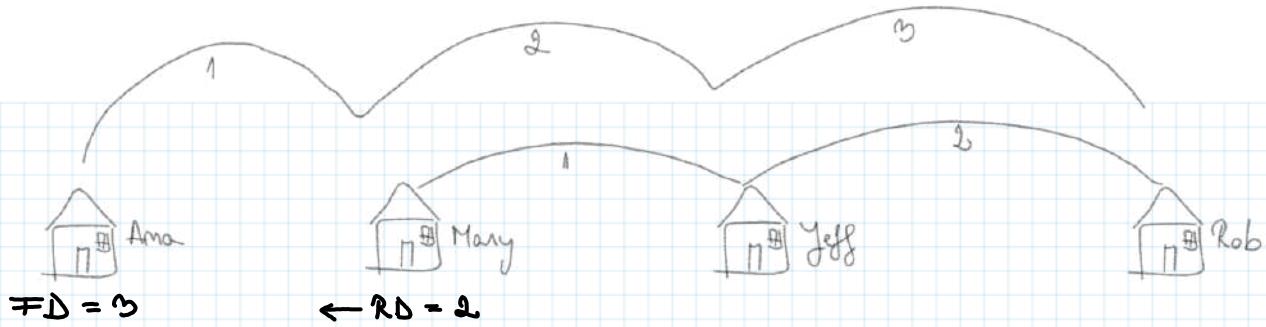


$$((10^7 / 100,000) + 10 + 10 + 10) * 256 = 33,280$$

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

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

distanta raportată + distanța de la vecinul care me-a spus de nodă



Ama: Hei Mary, tu cum să facă locoaste Rob?

Mary: Da, la 2 case de mine!

Ama: Super, slăcă ești vecina mea, îm se amă că la 3 case de mine!

### • Successor vs Feasible Successor

↓  
noda cu cel mai  
bun metrică până  
la deschidere  
(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

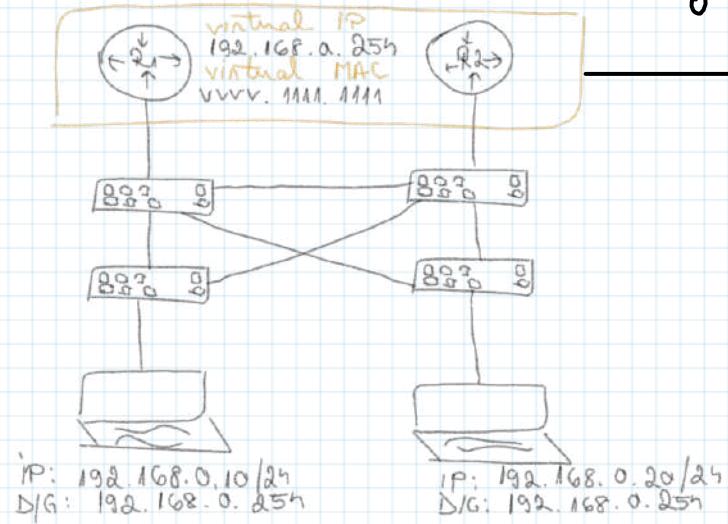
bandwidth x  
show ip route eigrp 1 b 10.0.0.0  
begin

show ip eigrp topology

→ schimbă valoarea în x

→ schimbă toate informațiile, inclusiv  
s și FS (chiar și  $(FD, RD)$ )

## Frist Hop Redundancy Protocol (FHRP)



se creează un grups pentru a evita erorile care să se petnească dacă un router nu poate să se conecteze la un singur router, care ar putea erau, dacă un dispozitiv ar emisări, datele unui singur router

→ în cazul în care un router creează și cel rămas nu primește nicio reacție pentru adresa MAC' (deci dispozitivele vor să se folosească de tabela 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

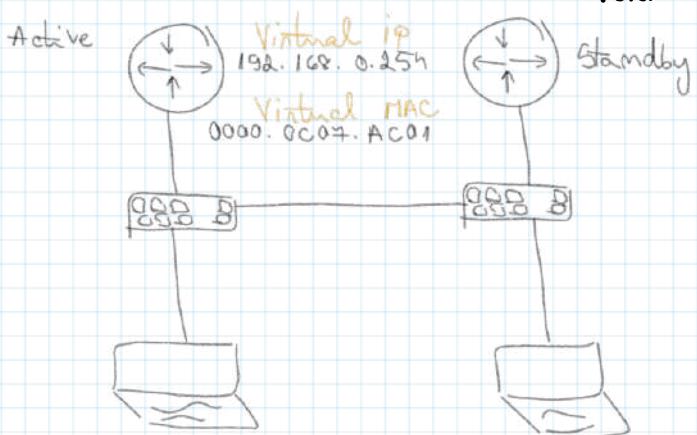
(Gateway Load Balancing Protocol)

### HSRP

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

Venmion 1 22h.0 0 2

Venmion 2 22h.0 0 102



Active Router Election

Highest HSRP Priority  
↓

Highest ip Address

Virtual IP - configurat să devină default gateway  
 Virtual MAC - se generează automat

Version 1  
 0000.0C07 ACXX  
 ↓  
 Group ID

Version 2  
 0000.0C9F TXXX  
 ↓  
 Group ID

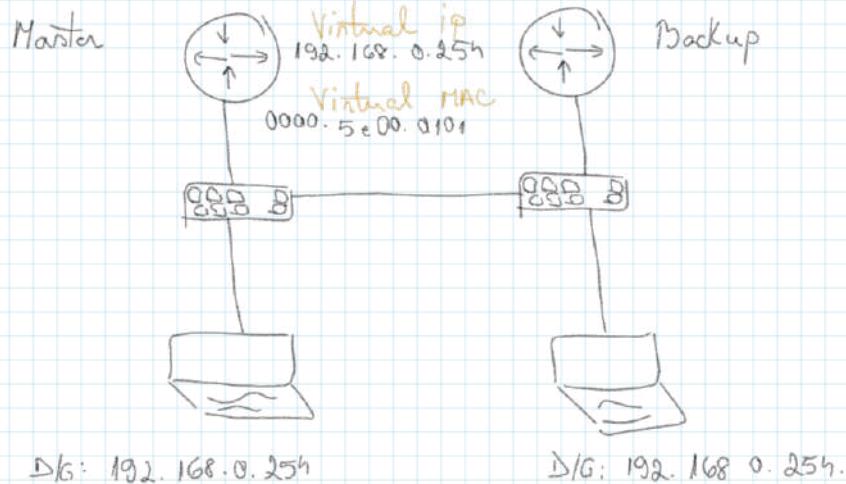
- dacă active router e înecă, standby devine automat active și îi sumează în pe rețea
  - ↳ dacă suntează revine, devine standby (repetă rețea care standby preiau dacă vrei să fie activ)

### VRRP

- în loc de active și standby  $\Rightarrow$  master și backup
- unul dintre routere primește Virtual IP  $\Rightarrow$  IP Address Owner

### Master Router Election

IP Address Owner  
 ↓  
 Highest priority  
 ↓  
 Highest IP Address

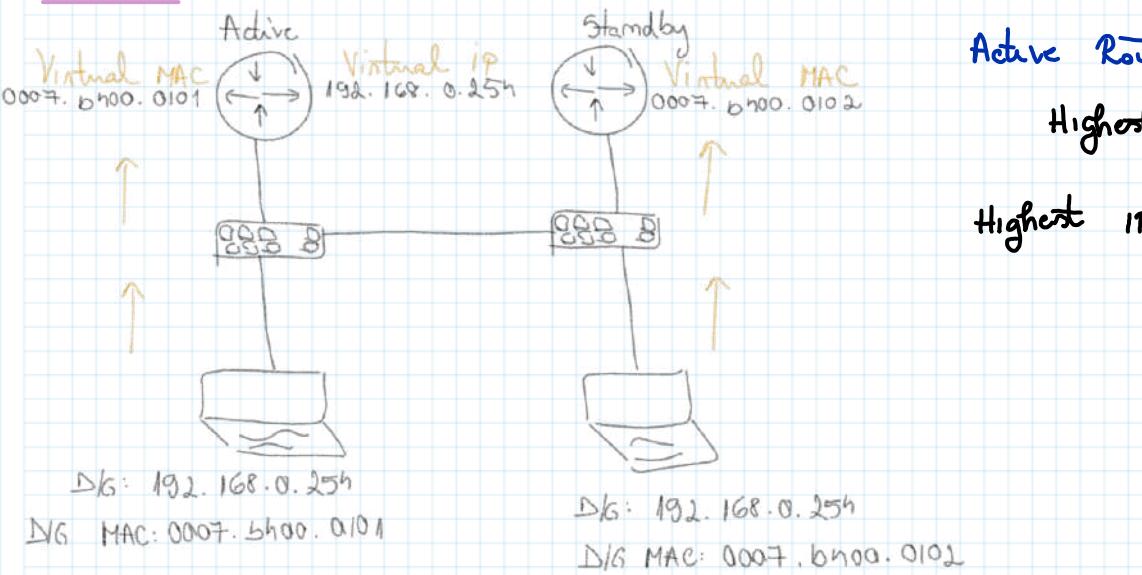


Virtual MAC: 0000 5e00. 01XX  
 ↓ group id  
 VRRP  
 Virtual MAC

- doar master trimit mesaje în rețea
- VRRP Master devices trimit advertisement la adresa multicast 224.0.0.18 tot la 1 sec.

- dacă master erneată, se arreagați 3 rec. (3x adv timer) și urmă păc, apoi backup devine master
  - ↳ dacă îmi revin, devine iar master automat

### GLBP

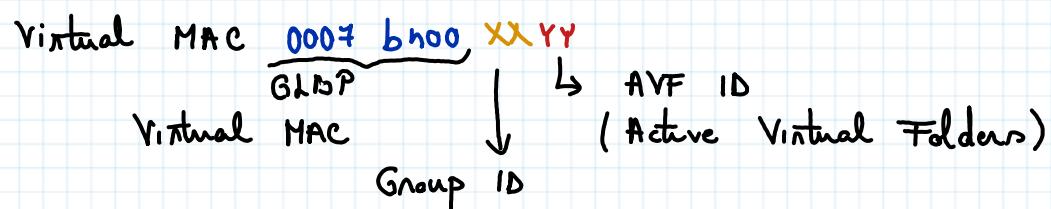


### Active Router Election

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, înlocuind fie unul active în altul standby
- dacă ernează active, se arreagați 10 sec., apoi standby devine active și prezintă adresele MAC
  - ↳ dacă revin, devine iar stand by, dar își ia adresa MAC împotrivă

Router Roles Multicast Addrs. MAC addr. Format

HSRP	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.
VRRP	RFC 5498	Master Backup	224.0.0.18	0000 5E00.01XX	One Advertisement (L) 1 sec. Master Down (L) 3 sec.
GDPBP	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

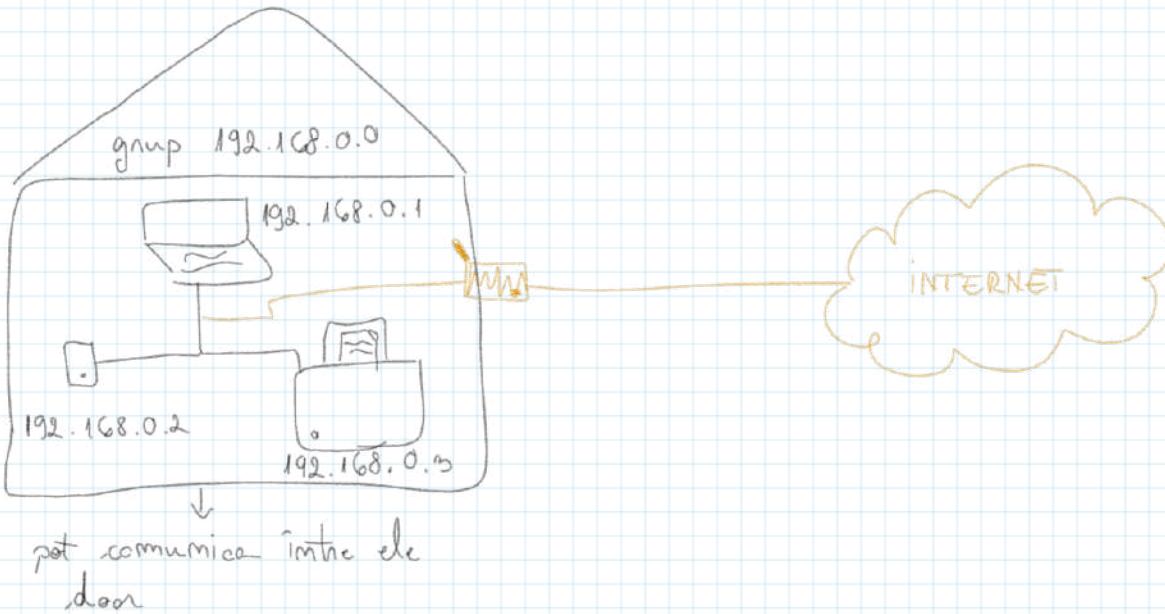
- internetul a tot crenat în niciun terminal ip-urile

The solution - Private Addresses 10.0.0.0 - 10.255.255.255.

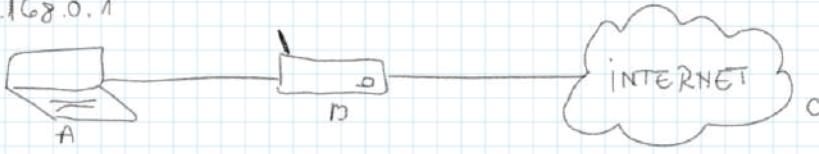
Se pot folosi doar în  
rețele interne, nu se poate  
în internetul public

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

 $A \rightarrow B$ 

0

indica inclusiv aplicatia / tabul de care aparține

Source 192.168.0.1 8897 noutenul le  
 Destinație 55.06.47.88.80 schimbă

crează tabelul, apoi trimite datele

de obicei se păstrează dacă e necesar, și folosint  
următorul liber

11.22.33.44 8899  
55.06.47.88.80



Întrare	iese
192.168.0.1:8897	11.22.33.44 8897

Întrare	iese
192.168.0.1:8897	11.22.33.44 8897

 $C \rightarrow B$ 

0

Source: 55.06.47.88.80

Destinație 11.22.33.44 8897



55.06.47.88.80

192.168.0.1 8897

### 2. Dynamic

- funcționează asemănător, dar

$A \rightarrow B$  - noutenul alege prima adresă liberă găsită în  
piscină

11.22.33.44 - 11.22.33.99

- se fac totușu pași de mai sus (cu schimbare, tabel în tot)

$C \rightarrow A$  - se fac totușu pași de mai sus

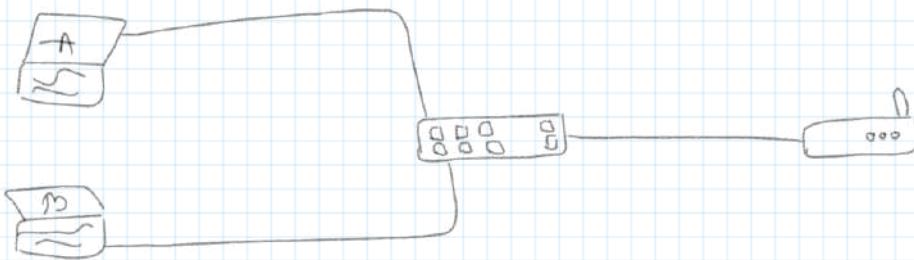
- după ce ajung datele înapoi la device, adrese ip merge înapoi în  
piscină și va putea fi folosită iar

### 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

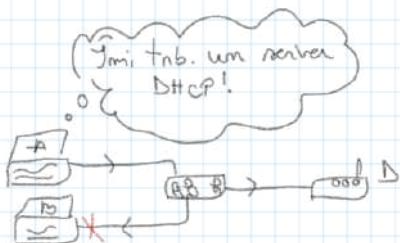


Adresele lui A și B trebuie să fie unice, ca să meargă datele sunte trăbuite

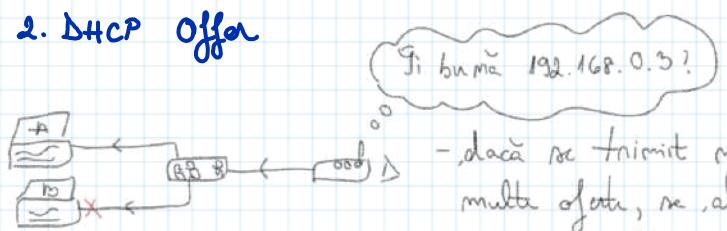
A și B vor fi DHCP clients  
DHCP Server va fi un router sau un server

→ la ce urmărește, se trimit mesaje broadcast, deci le primește tota lumea și cui nu i se adresează, le ignora

### 1. DHCP Discover



### 2. DHCP Offer



### 3. DHCP Request

A zice că o vrea

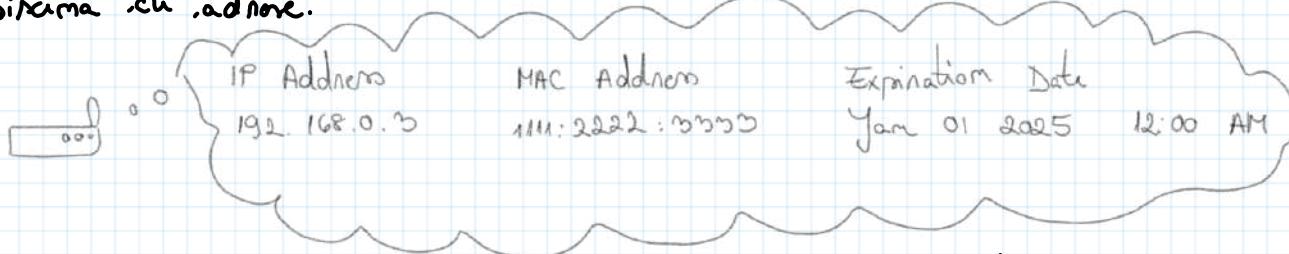
### 4. DHCP ACK

D: OK, fi-o săn, săn și vreau înapoi

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

Serviciul DHCP time apoi evidență

Dispozitivul trebuie să își recunoască adresa, astfel expira în mișcare înapoi în prima cu adresa.



⇒ evitarea nimănui adreseelor IP (dacă anunță / deconectezi un dispozitiv)

## Syslog

### Syslog Server

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

- Beneficii
- verifică toate informațiile mai ușor, dintr-un singur loc
  - date retention (când se rezarcăază un device, logurile se resetează)
  - se arhivează mai ușor

Cincoi devicii le rezolv în RAM



### LOG information

- înregistrările / jurnalul
- care capturează evenimentele activității sau mesajelor între sistem / aplicație

### Jurnal de rețea

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

### Data retention

- politica și practicile în ceea ce privește păstrarea și stocarea datelor pentru o anumită perioadă de timp

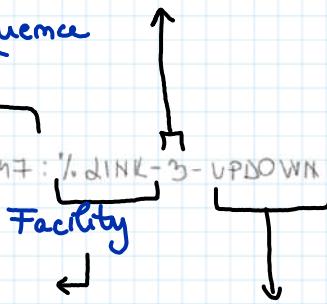
### Severity

= cât de urgent e log RMS

### Timestamp / Sequence number

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

număr  
mesajului



- cod pentru identificarea RMS-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, îți le dă pe toate de la severitatea saia în sus. dacă scri "Informational", îți dă tot ( $6 \rightarrow 0$ ) pînă la Emergency

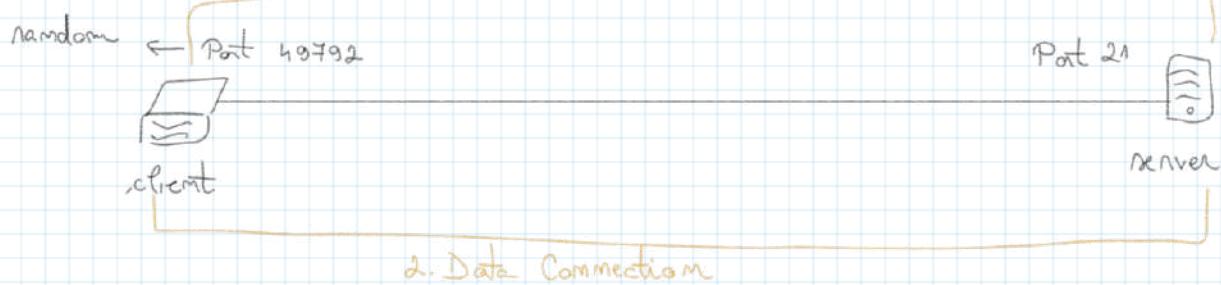
FTP

## (File Transfer Protocol)

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



## 1. Control Connection



## 2. Data Connection

## 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 din partea serverului, nu fie blocată
2. Pasiv: clientul face primul pas de pe portul număr generat random, către portul destinație 21
  - dacă există firewall, acesta nu blochează traficul, pentru că cel care a inițiat conexiunea, a fost clientul
  - nu este necesar, deocamdată, să toate datele sunt transmise clar

### FTPS (FTP Secure / FTP SSL)

- extensie a FTP care supune utilizarea a TSL și SSL encryption
- protocol de criptare permitând menținerea datelor în siguranță și departe de hackeri  
(îl privim ca și pe un tunel, care nu lăsa datele să se vadă)
- nu trebuie confundat 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

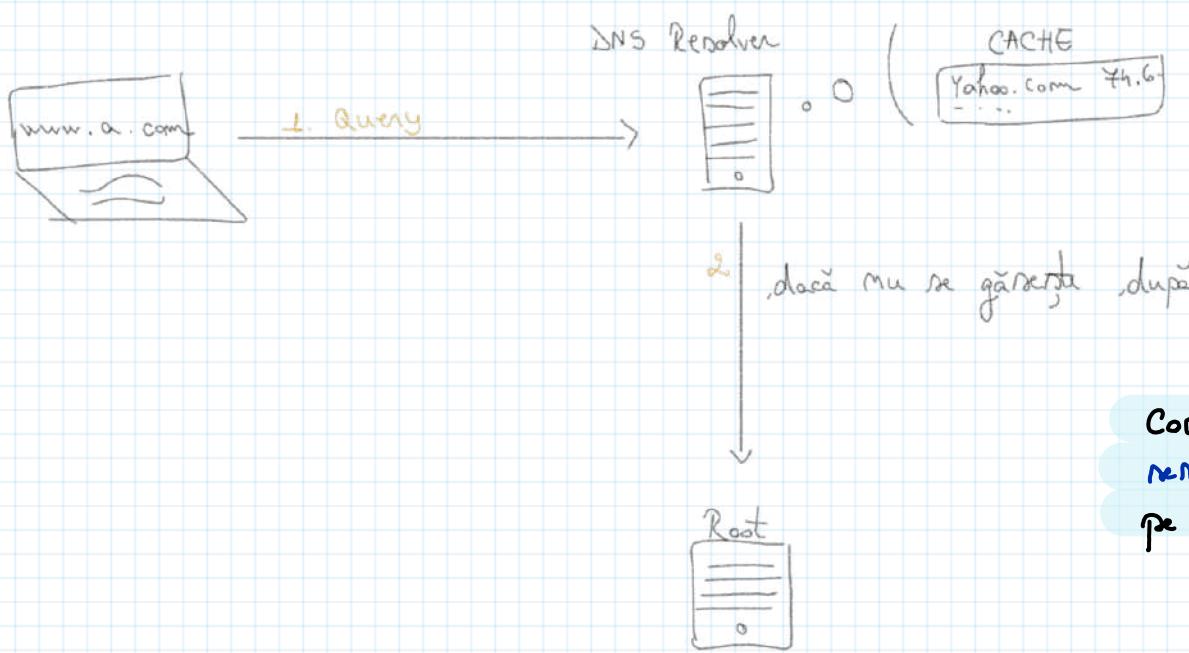
### Domain Name System

- 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



, dacă nu se găsește, după 1

Comunicația între  
servere DNS se face  
pe portul 53 prim **UDP**

### Root Name Server

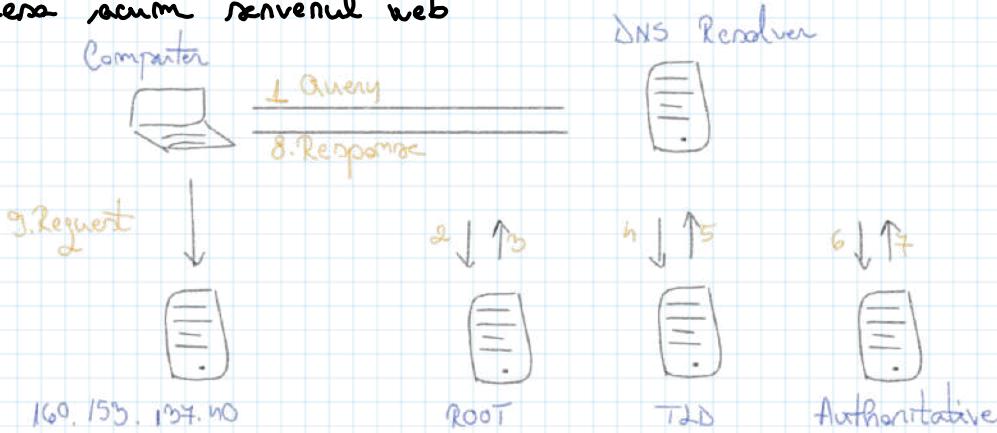
- primul din hierarhia DNS
  - primul pas pt. transformarea linkului în adresă IP
- există foarte multe, dar fiecare folosește 1 din 13 adrese IP
- Rol: să găsească datele, de pe top level domain servers (com, org etc)

### TLD Name Server

- conține informații pentru domenii with a specific extension (com, net, org etc)
- tot nu știe adresa IP de care avem nevoie
- știe 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 acum serverul web



## Lista de adrese IP pentru Root

198.41.0.4  
 199.9.14.201  
 192.53.12.12  
 199.4.91.13  
 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  
 199.4.83.42  
 202.12.24.33

### Type A

- = înregistrare a unei IPuri (pentru un domeniu)
- pt IPv4, AAAA

## Proxy Server

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

### Tipuri

Forward Proxy - client - servere web

- performanță și securitate comenziului la internet

Reverse Proxy - client - unul / mai multe servere web

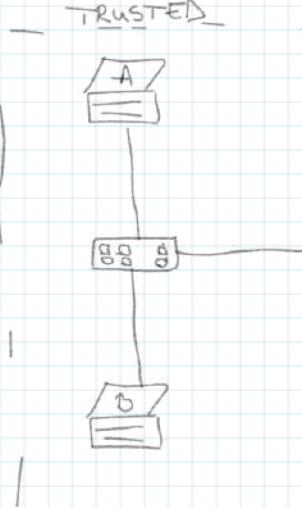
- performanță, securitate și confidențialitatea comenziului la internet

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

$\downarrow$

în general pt navigare anonimă, ascunderea adresei IP

## Firewall



## Traditional

- 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

Așa cum 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

## NGFW's

### (Next-Generation Firewalls)

Application level inspection (identifică și blochează)

IPS (Intrusion Prevention System) - patterns / signatures  
- anomalies

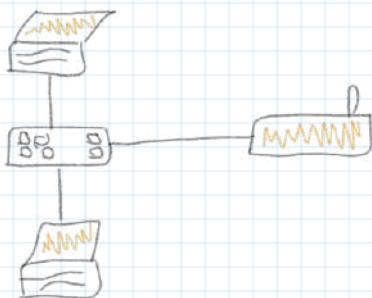
### Threat Intelligence (updates)

#### Features

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

} UTM (Unified Threat Management)

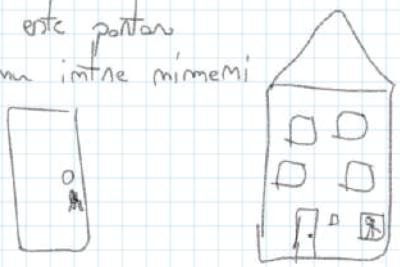
## Software based firewalls



- „dubă protecție”, dacă am emis o față vîme, din ext
  - protecție, dacă vîme din interior
- ex Windows Firewall

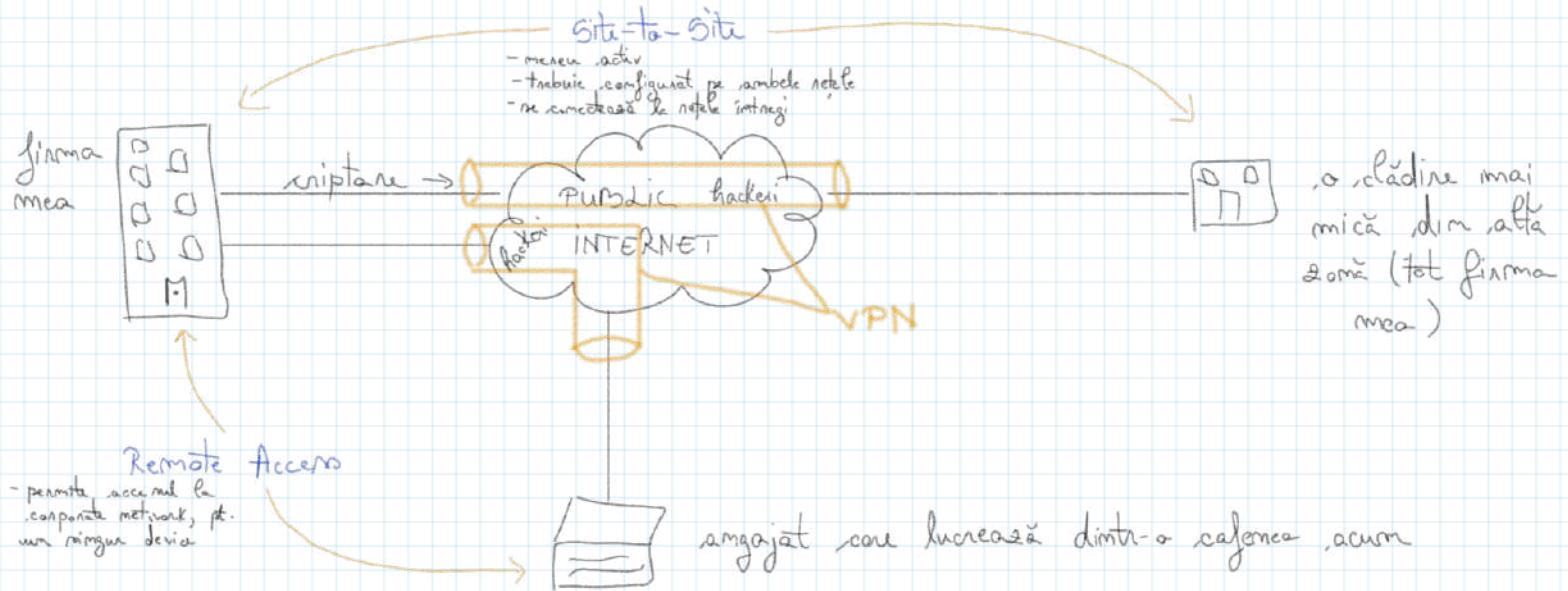
Că rai la cămin:

Trebuie să pot intra în pens. închisime, pt. că și tu trebuie, cătelea nu este porturi  
Dacă moi tot me închidem cu cheia usor de la cămină la ră sună înțele nimerești,  
care e în cămin deja (nu pt. că mai doar me portanul)

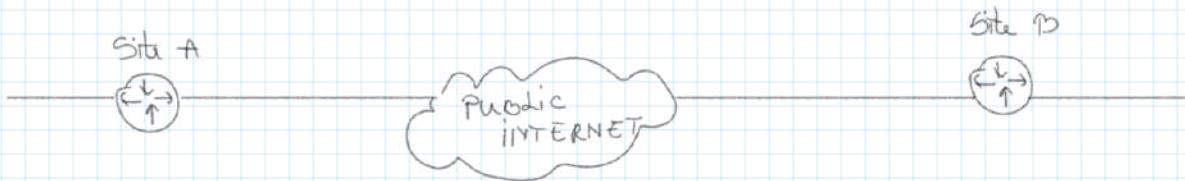


## VPN (Virtual Private Network)

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



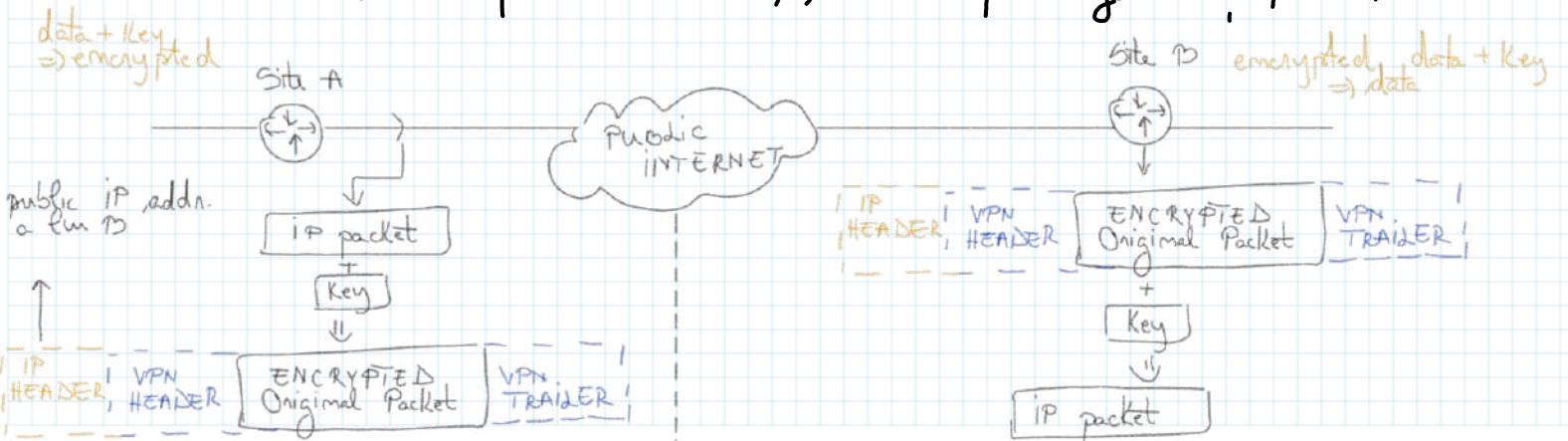
### Site-to-Site VPN



- se configuriște deobicei pe un router / firewall pe ambele rate-uri

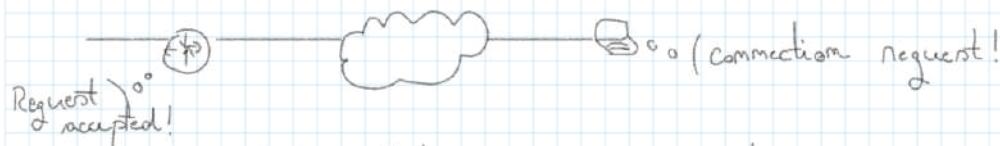
IPSEC VPN = framework / set of rules pt. creaarea VPN-ului în rețea (= pt. securizarea comunicării între o ip network)

- permite folosirea mai multor protocoale pt. fiecare 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 secrete către host ca să te conectezi la rețea



ex. de VPN Client Application: Cisco Anyconnect, OpenVPN

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

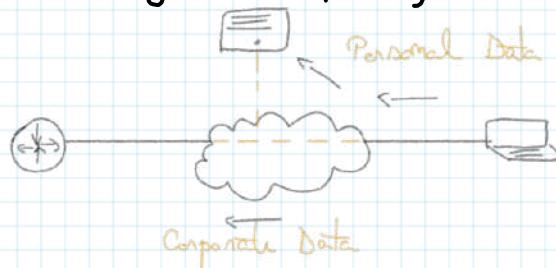
↳ succesor a lui SSL (Secure Sockets Layer)

- se folosește și pt traficul web, la conectarea la rețeaua http
- folosește în general portul 443, care e permis în general (e bine, pt. că unele wifi-uri 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 lista filtrelor de routare în switch-uri pt identificarea trafiului  
 → în funcție de ip addrs.  
 (source addrs, destination addrs)  
 și port numbers
- în general filtre pentru deny / allow traffic
- se mai folosesc pentru NAT și 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	or	

### Ondimea regulilor

- menge dim 10 în 10 ca să poată revă în mai adângi între
- ondimea e foarte importantă, pentru că menge de reguli împreună nu sănătă un match pentru regulă. dacă găsește, aplică regulă și se oprește din căutare
- dacă nu se face niciun match  $\Rightarrow$  denied (ultima regulă e implicit Deny)

### Tipuri

#### Standard ACL

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

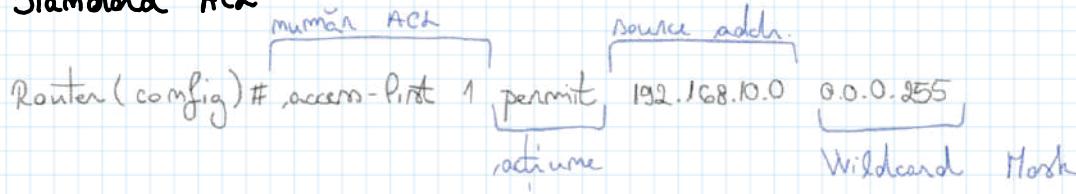
#### Extended ACL

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

#### Named ACL

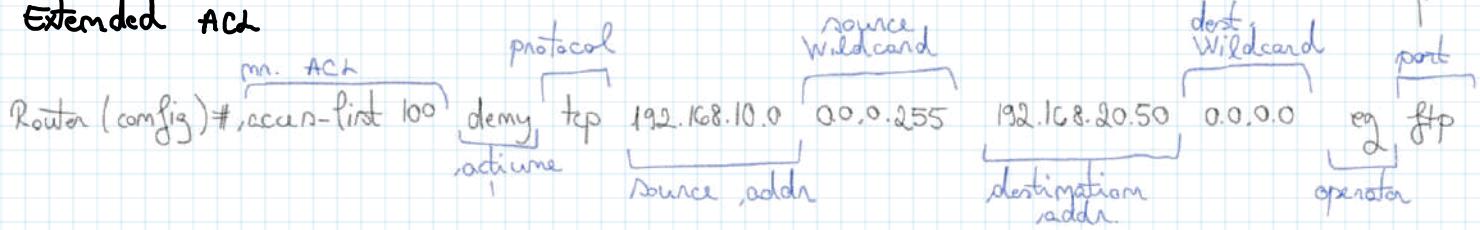
- permite ca standard și extended ACLs să primească nume ca să fie mai ușor de gestionat (ex: când sunt mai multe ACLs pe un device, să stiu care fiecare de ce se ocupă)

## Standard ACL



re pot folosi m. sau cuv.  
chiar;  
de ex: ftp inseamnă 21

## Extended ACD



Named ACD tipul: standard/extended

```
Router(config)# ip access-list extended name  
drama
```

```
Router (config-ext-mod) # deny top 192.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0 eq ft  
Router (config-ext-mod) # permit ip 199.168.10.0 0.0.0.255 192.168.20.50 0.0.0.0
```

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

!!! este importantă, analizarea

## Wildcard Mark

- can be a subnet mask inverted

$0 = \text{bits must match}$

$1 = \text{bits do not matter}$

Adresa IP

192 168 10 0      11000000 . 10101000 00001010 00000000

## Wildcard Mask

0 0 0 255 00000000 , 00000000 00000000 11111111

→ match-werst teile radnerte ip imme 192.168.10.0 /24 192.168.10.255

## Port Operator

- $\neq$  = Greater Than
  - $\leq$  = Less Than
  - $\neq$  = Not Equal
  - $\in$  = 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 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
```

Dacă Wildcard Mask are 32 de bîte / e 0.0.0.0, putem său Keyword-ul "host"

Standard IP access list 10

```
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 102

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

Potem folosi Keyword-ul "any", pentru a fi acceptata orice adresa IP

# API (Application Programming Interface)

= "interfață" către o aplicație

## HTTP Methods

POST

GET

PUT

PATCH

DELETE

## CRUD

CREATE

READ

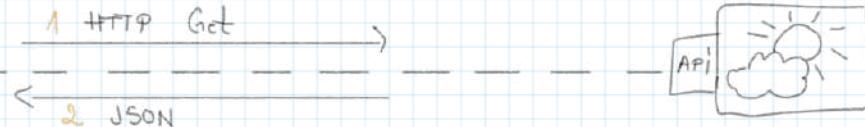
UPDATE (replace)

UPDATE (modify)

DELETE

exemplu:

aplicație mobilă  
care să vădă vremea



aplicație pt. vreme, cu pt. multe  
pt. buton de meniu



L: <https://api.openweathermap.org/data/2.5/weather?q=city name&appid=API key>

unic, al tău, ca aplicația nă timă  
evidența ceeașa că nu îți  
spălați

2: "main":  
 "temp": 9.53,  
 "feels-like": 7.62,  
 "temp\_min": 4.78,  
 "temp\_max": 10.56,  
 "pressure": 1016,  
 "humidity": 61

4

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

↳ e interesant, poti de ex să vezi căte like-uri, videoclipuri, subscriveni etc. une vrem, canal

## IP Addresses și NetMask

### IP Address (IP, h)

- 32 de bîta  $\Rightarrow$  4 octeta

00000001.00000010.00000011.00000100 Adresă mai ușoară  $\rightarrow$  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

### Adresa de rețea

= NM AND ADRESĂ IP

### 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

$$\begin{array}{r} 11111111.11111111.11111111.11000000 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 1.2.3.4 \end{array} \Rightarrow 2^7 + 2^6 + 2^5 + 2^4 = 128 + 64 + 32 + 16 = 240$$

$$2^7 + 2^6 + 2^5 + 2^4 =$$

$$128 + 64 + 32 + 16 =$$

$$192 + 48 = 240$$

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)

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

## Network Address (NA)

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

## Broadcast Address (BA)

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

Ex 1:

$$IP = 10.11.12.13$$

$$NM = 255.255.255.0 \quad /24$$

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

$$[NA \dots BA] \quad \text{mărime} = 256$$

NA

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

BA

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

$$\Rightarrow IP = 10.11.12.13$$

$$NM = 255.255.255.0$$

$$[10.11.12.0 \rightarrow 10.11.12.255], \quad \text{mărime} = 256$$

Ex 2:

$$IP = 10.11.12.15$$

$$NM = 255.255.255.248 \quad /29 \quad \Rightarrow 2^5 \text{ IP-uri}$$

$$NM = 255.255.255.1111000$$

$$\begin{array}{r} NA = 10.11.12.00001101 \text{ AND} \\ 255.255.255.1111000 \\ \hline 10.11.12.00001000 \end{array}$$

$$NA = 10.11.12.8$$

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

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

$$0.0.0.00000111$$

$$\hline 10.11.12.00001111$$

$$BA = 10.11.12.15$$

$$\Rightarrow IP = 10.11.12.15 \quad \text{și} \quad NM = 255.255.255.248 \quad /29$$

$$[10.11.12.8 \rightarrow 10.11.12.15], \quad \text{mărime} = 8$$

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

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

$$/32 \quad 255.255.255.255$$

- marcă de netea care demonstrează că nu există la un anumit IP, nu la cel din care face parte

0
1
10
11
100
101
110
111
000
001
1010
1011
1100
1101
1110
1111

## Network Splitting

1.0.0.0 /24  
 ↓  
 network address → mask are 24 de 1 ⇒ 8 de 0 ⇒ NM = 255 255 255.0  
 size = 256

[1.0.0.0, , 1.0.0.255]  
 \_\_\_\_\_  
 256 valori

Impărtirea: [NA .. BA] ⇒ [NA1 BA1][NA2.. BA2]  
 NM + 1

Exemplu:

[1.0.0.0 ... 1.0.0.255] /24  
 $/24 = 11111111.11111111.11111111.00000000 = 225.225.225.0$   
 $24 + 1 = 25$   
 $/25 = 11111111.11111111.11111111.10000000 = 225.225.225.128$   
 $\Rightarrow [1.0.0.0 ... 1.0.0.127] \underbrace{[1.0.0.128 ... 1.0.0.255]}_{128} \text{ size} = /25 = 2^7 = 128$   
 \_\_\_\_\_  
 128 \_\_\_\_\_ 128  
 \_\_\_\_\_  
 256

Cost general NA și BA:

m devices → m IP addrs.

NA și BA nu se pot folosi pe dispozitive ⇒ m+2

Mac -  $2^x$  ⇒  $m+2 \leq 2^x$

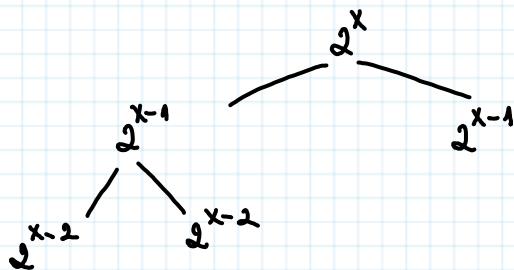
NM =  $/(2^x - 1)$ , unde 11.10 00 avem x zero-uri în  $2^x - 1$  de 1

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
[NA1 BA1][NA2 . . . BA2]	NM + 1	
[NA1 BA1][NA2 BA2][NA3 BA3][NA4 .. BA5]	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)

→ m devices (IP) + 1 router + 1 NA + 1 MA → m+3

acă nu trebuie +3,  
că suntem doar  
routere să conectăm  
rețelele între ele

⇒ + NA + MA = m+2

$$h0+3 = h5 < \textcircled{6h} = 2^6 \rightarrow 6 \text{ de } 0 \rightarrow 32 - 6 = /26$$

$$hh+3 = h7 < \textcircled{6h} \Rightarrow /26$$

$$1G+3 = 19 < \textcircled{32} \Rightarrow 2^5 \Rightarrow /24$$

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

$$h+3 = 7 < \textcircled{8} \Rightarrow /29$$

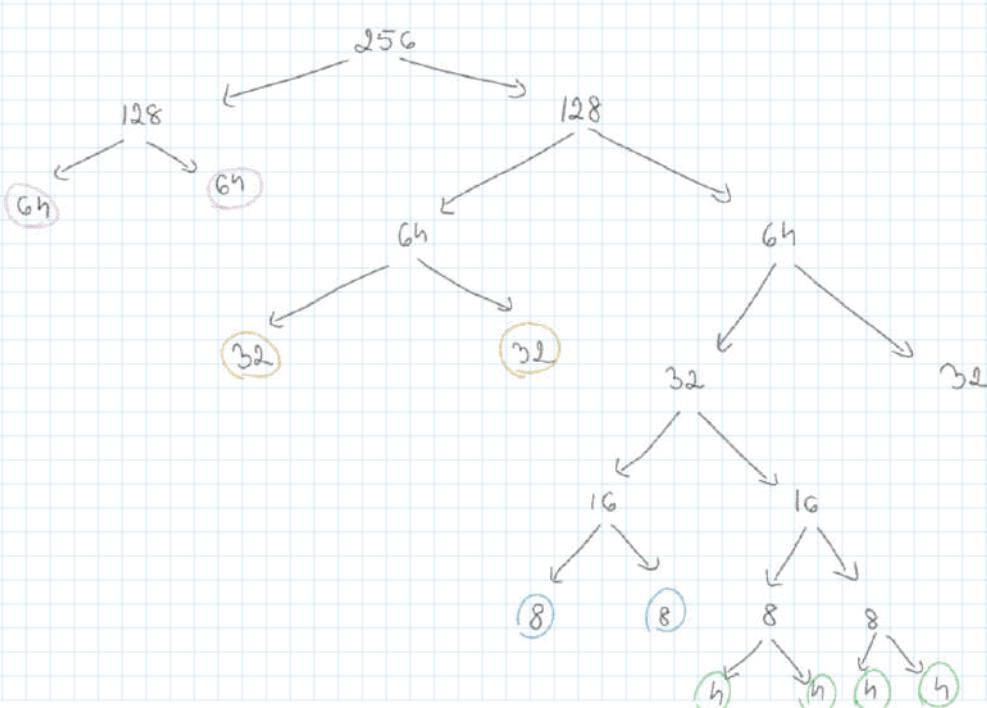
$$5+2 = 7 < \textcircled{8} \Rightarrow /29$$

$$2+2 = 4 < \textcircled{h} \Rightarrow /30$$

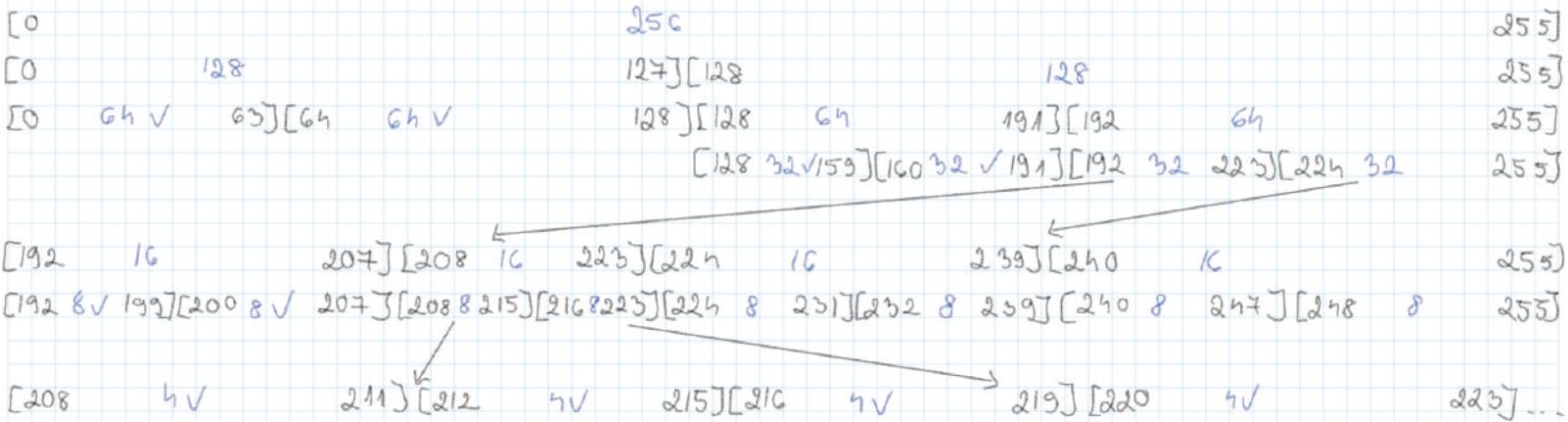
$$2+2 = 4 < \textcircled{h} \Rightarrow /30$$

$$2+2 = 4 < \textcircled{h} \Rightarrow /30$$

Verificare:  $2 \times 6h + 2 \times 32 + 2 \times 8 + 4 \times h = 22h < 256 \quad \checkmark$



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: 82.228.39.216/30
- N10: 82.228.39.220/30

- NM: 255.255.255.192
- 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

available: 82.228.39.224/29 NM: 255.255.255.248

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

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

## Router

- dispozitiv care conectă 2 rețele diferite
- redirecționează pachete de date folosind adresele IP atribuite fiecărui dispozitiv într-un LAN
- preia pachete de date de la device-urile conectate la LAN și le redirecționează către și de pe internet către segmentele LAN, prin NAT

## Switch

- dispozitiv care poate conecta mai multe dispozitive într-un LAN
- mai inteligente decât hub-urile (decât noilele mi)
  - ↳ nu pot limita traficul de date către și de la fiecare port ( $\Rightarrow$  fiecare dispozitiv are o capacitate suficientă de bandă)
  - ↳ 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 routernu care conectează un LAN la internet sau la altă rețea
- în general, e aceeași cu adresa IP a routernu
  - ↳ el directează datele între rețele