

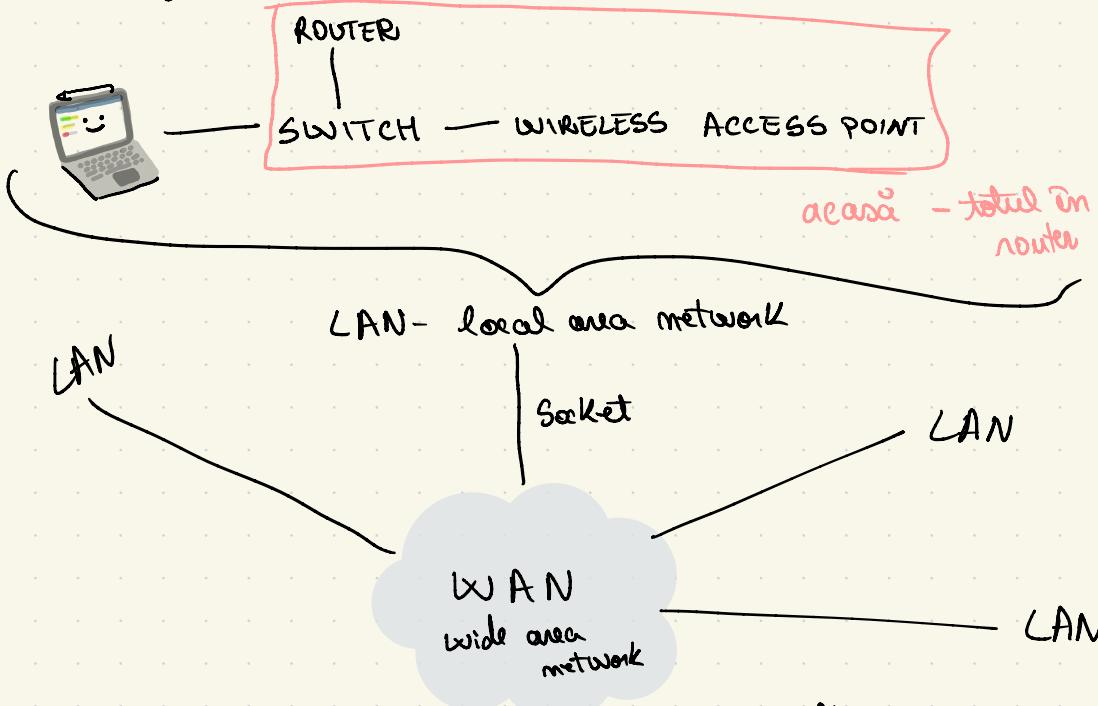
Regule de calculatoare

2023



Introducere în rețele

- CISCO - vendor
- cum funcționează o rețea, OSI Model, TCP/IP
- o rețea = un grup de echipamente / obiecte interconectate
- o rețea de calculatoare = un grup / sistem de calculatoare interconectate



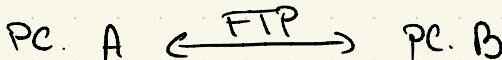
- switchuri în BACK - interconectarea mai multe PC-uri

ROUTER — INTERNET — SERWER

- adresa IP = id al unui calculator dintr-o rețea
 - ex: 192.168.0.10
- service provider = companie care îți oferă acces la internet
- MAN - metropolitan area network
- WAN - pe mai multe continente

- CISCO IOS - 50 router, CLI (command line input)

Cum comunică calculatoarele în internet

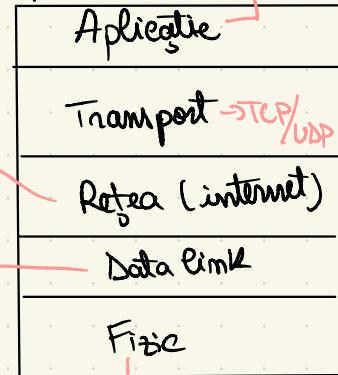


- Strîns de protocoale OSI (TCP/IP)

- adresă MAC - identifică mod unic un dispozitiv în LAN

Ethernet
(folosește adresa MAC)

IP



- TCP - de încredere

UDP - rapidă

- HTTP - protocol pt către date web

- Standard - 100 Mbps (transfer date)

- interfață = portul logică, port = portul fizic
↳ configurăm

Aplicație

Transport \rightarrow TCP / UDP

Retea (internet)

Data link

Fizic

mediu
de transmitie

IPv4 (pe nivel 3)

→

→ type of service

TCP Header

Version	TOS	Total Length	
Identification		Flags	Fragment offset
Time to leave TT	Protocol	Header checksum	
IP source addr.			
IP dest. addr			
Options			
DATA			

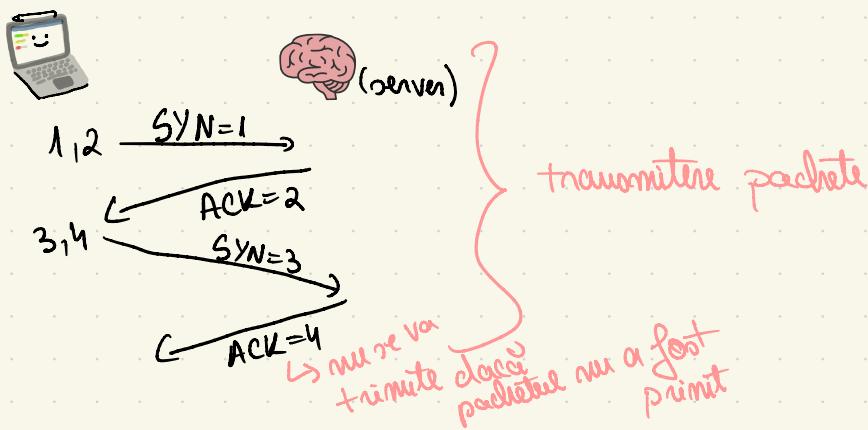
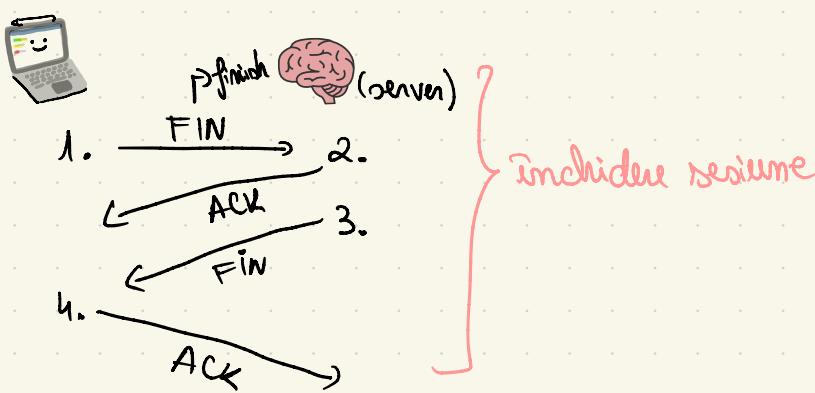
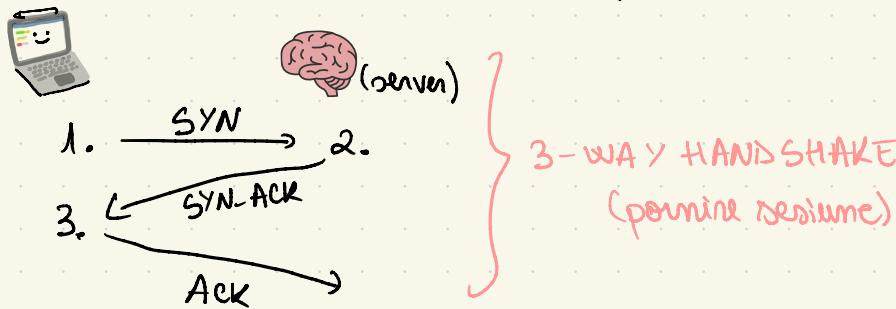
protectie
impostor
buclelor

sumă control (stabilirea integrității
paquetului)

} de unde pleacă^z
} și unde ajunge paquetul

TCP / UDP

- stabilește conexiune cu serverul
- retransmite pachete (în caz ce nu a ajuns)
- ordonează pachetele cînd se face distinție



UDP

- NU stabilește conexiune
- NU retransmite datele
- NU ordonează pachetele

→ utilizat pt aplicații real-time (VoIP)

Socket → end-point communication

- o interfață de programare a aplicației utilizată în comunicarea între două dispozitive / procese

client

```

struct sockaddr_in server; stocă info despre server
int c = socket (AF_INET, SOCK_STREAM, 0); crează socket
if (c < 0) return 1;
↳ descriptor socket
memset (&server, 0, sizeof (server)); iniț. struct server
server.sin_port = htons (1234); setează portul
server.sin_family = AF_INET; set. familia de adrese (IPv4)
server.sin_addr.s_addr = inet_addr ("192.0.0.1"); convert
if (connect (c, (struct sockaddr *) &server, sizeof (server)) < 0)
return 1; ne conectată la server
:
send (c, &var, sizeof (var), 0);
recv (c, &rez, sizeof (rez), 0);

```

htons()
convertește în format

Server

```

struct sockaddr_in server, client;
int s = socket (AF_INET, SOCK_STREAM, 0);
if (s < 0) return 1;

```

```

memset (&server, 0, sizeof (server));
server . sin_port = htons (1234);
server . sin_family = AF_INET;
server . sin_addr . s_addr = INADDR_ANY;

```

```

if (bind (s, (struct sockaddr *) &server, sizeof (server)) < 0)
    return 1;                                leaga socketul la adresa si portul
                                                specificat

```

```

listen (s, 5);   astepta socket cu o coada de
                  acceptare de pana la 5
int l = sizeof (client);
memset (&client, 0, sizeof (client));

```

```
while (1) {
```

```
;
```

```
c = accept (s, (struct sockaddr *) &client, &l);
```

```
recv (c, & var, sizeof (var), MSG_WAITALL);
send (c, & rez, sizeof (rez), 0);
close (c);
```

```
}
```

Port numbers

- HTTP 80 → SMTP 25, FTP 21
- port per aplicatie (m. per casă)
- Destination port, Source port

IP: Port



10.54.16.1:80
10.8.24.5:63139



Server

DST Port: 80

SRC Port: 63139

DST Port: 63139

SRC Port: 80

- freeene protocoale sunt porturi din urmă
- 0-1023 porturi cunoscute
- 1024-49151 porturi înregistrate
- 49151-65535 porturi generate dinamic

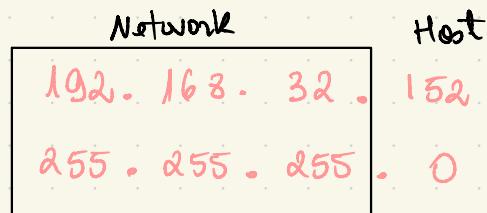
HTTPS → 443

Adrese IP (network layer)

↳ identificator unic assignat fiecărui computer conectat la o rețea

- IPv4, IPv6.
- ↓
32 bits

subnet →
mask



- Subnet mask = utilizat pt a separa o adresă IP în Network și Host

- adresele IP trebuie să fie unice doar în rețea la care (ex: vecinii pot avea aceeași adresă IP)

ISP-ul trebuie să aibă adresa IP publică unică

- clase IP

- clasa A: 1.0.0.0 - 126.0.0.0 (rețele mari)
- clasa B: 128.0.0.0 - 191.255.0.0 (rețele medii)
- clasa C: 192.0.0.0 - 223.255.255.0 (rețele mici)

192. 168. 0.1 → 1100 0000 . 1010 1000 . 0000 0000 . 0000 0001
8 bits - 1 octet
Binar

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

192.54.103.0

↳ network addr.

192.54.103.255

↳ broadcast
addr.

pot trimite
în tota
rețea

IPv6

- IPv4 - 4 bilioane

NETWORK

$2001:0db8:0000:0000:a111:bada:0000:abcd/64 \rightarrow$ 128 bit
global prefix
(ISP) subnet
ID HOST
interface id

$$\begin{matrix} 0 & | & 2 & | & 1 & | & 4 \\ 0 & | & 0 & | & 0 & | & 0 \end{matrix} \Rightarrow \begin{matrix} & & & & 2 \\ & & 8 & | & 0 & | & 1 & | & 1 \\ & & 0 & | & 0 & | & 1 & | & 0 \end{matrix}$$

$$\Rightarrow \begin{matrix} & & & & d \\ & & 8 & | & 1 & | & 2 & | & 1 \\ & & 1 & | & 1 & | & 0 & | & 1 \end{matrix}$$

echivalent cu $2001:db8:a111:bada:0:abcd$ secvență?

- fiecare dispozitiv are un IP public (nu mai e necesar un IP privat)

Global unicast / anycast - 2/3 --

Unique local - FC/FD

link local - FE

Multicast - FF

Adresa MAC (Data link layer)

↳ Media Access Control

identificator unic assignat unei interfețe de rețea (NIC)

NIC = dispozitiv hardware ce permite unui PC să

se conecteze la rețea și să comunice cu alte PCuri

ex: 08-00-27-EC-10-61 (adresa fizică)

- sunt unice și nu pot fi schimbate

08-00-27-EC-10-61
OUI-vendor unique value
(organizational
unique identifier)

3 tipuri:

- unicast (unică)
- multicast (pt explorare)
prefix: 01-00-5E
- broadcast (trimite spre toate disp.)

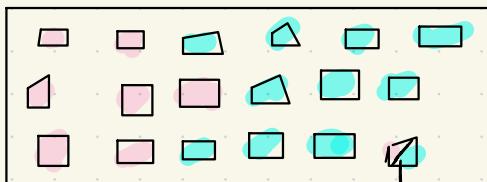
ARP → discover MAC addr., map them to an IP addr.

↳ addr resolution protocol

- ping = protocol de rețea utilizat pt a măsura timpul necesar ca un pachet să parcurgă drumul de la persoană până la destinație și înapoi.
- RARP - Reverse ARP (un disp cunoaște MAC addr și vrea să afle IP addr)
- DHCP - Dynamic Host Configuration Protocol
 - ↳ asignare dinamică a adreselor IP (permite disp. să obt. automat o adresă IP)
 - ↳ configurație gateway și servere DNS
(adresă gateway - router, etc)

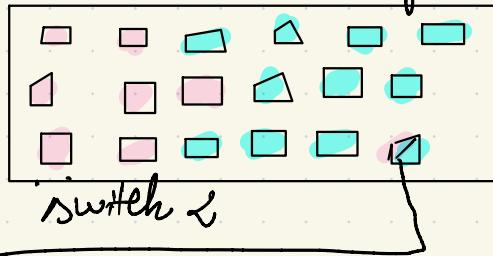
VLAN

- ↳ virtual local area network
- used for broadcast in department
- ↳ 04 VLANs



switch 1

- VLAN10
- VLAN20
- Trunk → send traffic with diff. VLAN.



switch 2

Computerul generează un frame (un set de date) ce conține și VLANul. Când există 2 switchuri, 802.1q tag (TPID, TCI)

TPID → 1 byte (ajută la identificarea frameului cu 802.1q)

TCI →
tag protocol id
Tag control info

TCI → prioritize
→ DEI (drop eligible indicator)
→ VLAN id

Native VLAN (default VLAN) → pt frameuri nntagged.
configurat pe trunk interface

Teorie project CISCO

→ intro IP & Netmask

IPV4: 32 bits long = 4 bytes ex: 1.2.3.4

Network mask (Netmask): 4 bytes ex: 255.255.255.0

255.255.255.0

y

2^8 IP address
network

→ using IP and NM

2^{32} total IPs in the internet

networks → into subnetworks

.0 → network addr. . 255 → Broadcast addr.