

## CURS 5 + 6 + 7

### Noțiunea de protocol

În cadrul stivei TCP/IP, la nivel Aplicație, este localizat orice proces, server. La acest nivel, aplicațiile comunică prin protocoale: *HTTP* (browser – clientul, server web), *FTP* (File Transfer Protocol), *SMTP* (Simple Mail Transfer Protocol), *POP* (Post Office Protocol), *IMAP*, *DNS*, *NTP*.

**Protocol** = set de reguli pe care trebuie să le respecte doi parteneri care comunică.

Pentru toate protocoalele din stiva TCP/IP există documente standard numite **RFC** (Request For Comments) care descriu foarte precis toate aspectele de comunicare de care trebuie să țină cont un implementator de client și unul de server, ca să le facă interoperabile, să funcționeze împreună.

### Protocoale text vs. Protocoale binare

```
unsigned short int i = 65123;    // același lucru cu uint16_t
```

Cum scriem acest număr într-un fișier?

Varianta 1:

```
f = open("date.txt", ...)
write(f, &i, sizeof(i));        // FE63 – în memorie 63h FEh – se scrie numărul
                                // așa cum e el în memorie
```

Varianta 2:

```
f = fopen("date.txt", ...)
fprintf(f, "%uh", i);           // se scriu 5 octeti: "65123", adică cifrele
                                // numărului
```

**Cum e cel mai bine să reprezentăm numărul în fișier?**

Dacă vrem să scriem un număr într-un fișier text, e nevoie de un delimitator, în cazul în care vrem să scriem după el un alt număr. În protocol, trebuie definit acest delimitator.

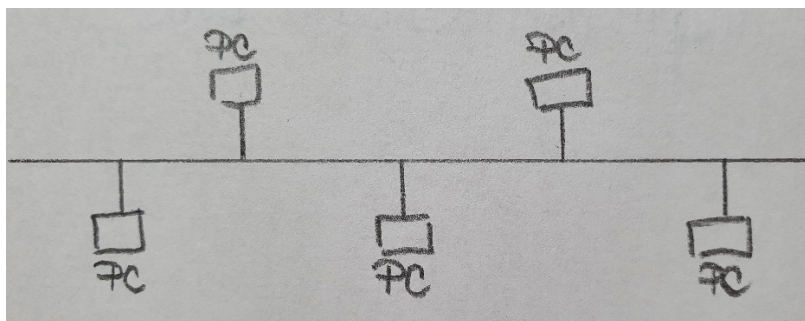
În format binar, trebuie să fim atenți la dimensiunea de reprezentare.

!!! În internet, se folosesc foarte mult protocoalele text, pentru că e mai ușor de făcut debugging și e mai ușoară de interpretat informația care se trimite pe pipe.

## Topologii de rețele

### **Rețele cu topologie liniară (topologie BUS)**

- toate calculatoarele erau conectate la același fir de comunicații (există o magistrală unică care conecta toate calculatoarele)



- operau la viteze de până la 10 Mbps, partajați între toate calculatoarele din cadrul rețelei (ex. dacă două calculatoare vor să comunice între ele, iar restul nu făceau trafic în internet, ele puteau comunica cu 10 Mbps; dacă voiau să comunice, în același timp, 4 calculatoare, 2 câte 2, viteza maximă cu care puteau să comunice 2 calculatoare se înjumătățea)
- această limitare a vitezei este datorată coliziunilor – fenomen strâns legat de rețelele cu difuzare

**Rețele cu difuzare** – dacă un calculator vrea să comunice cu altul, informația care se trimite nu se propagă numai pe segmentul de rețea dintre cele două calculatoare, ci se propagă pe toată lungimea firului → informația ajunge pe toate calculatoarele din rețea → este la latitudinea fiecărui calculator dacă vrea să primească informația → probleme de securitate.

!!! Singura metodă de contracarare a difuzării este **criptarea**

- dacă se rupe cablul de rețea, niciun calculator nu mai poate comunica, chiar dacă unele se află pe același segment întreg

Obs:

1 octet = 1 byte = 8 biți

1 Kilobyte = 1024 =  $2^{10}$  octeți

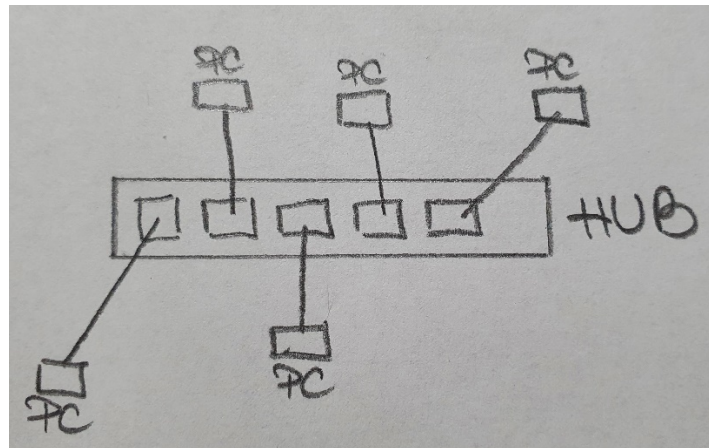
1 Megabyte = 1024 Kilobytes

1 Gigabyte = 1024 Megabytes =  $2^{30}$  octeți

→ unități care măsoară cantitatea de date

1 Kilobit = 1000 biti

### Rețele cu topologie stea

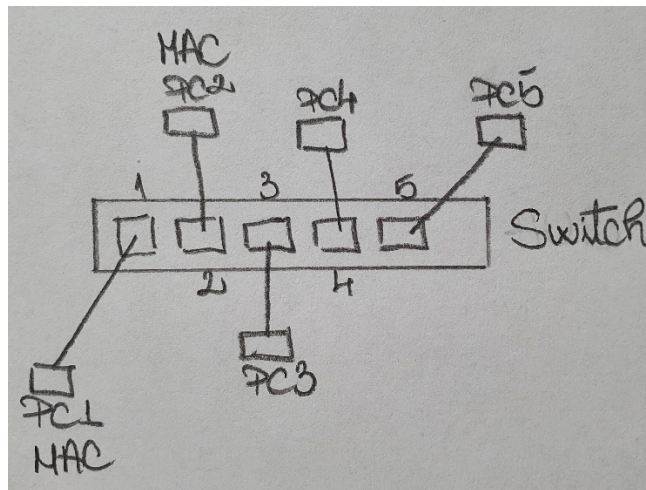


- primele rețele operau cu 10 Mbps
- în centrul rețelei se afla un HUB, care lua semnalul trimis de un calculator și trimitea tuturor calculatoarelor din rețea, nu numai calculatorului căruia îi era destinată informația
- dacă pica linia de comunicație între hub și un calculator, celelalte calculatoare din rețea continuau să funcționeze
- se păstrează problemele de coliziune și de înjumătățire a vitezei
- un HUB poate fi văzut ca un amplificator de semnal

## HUB vs. Switch

Spre deosebire de un HUB, un switch e un echipament “*smart*”, el are procesor și memorie, poate să ruleze și să implementeze logici funcționale. Un atu mare este că switch-ul *își numerotează porturile* și învață, în timp, ce calculator se află în spatele fiecărui port. Astfel, se evită coliziunile și viteza nu scade.

Switch-ul operează la nivelul Data-Link. Fiecare calculator este identificat printr-o adresă MAC (adresă Ethernet, adresă Hardware).

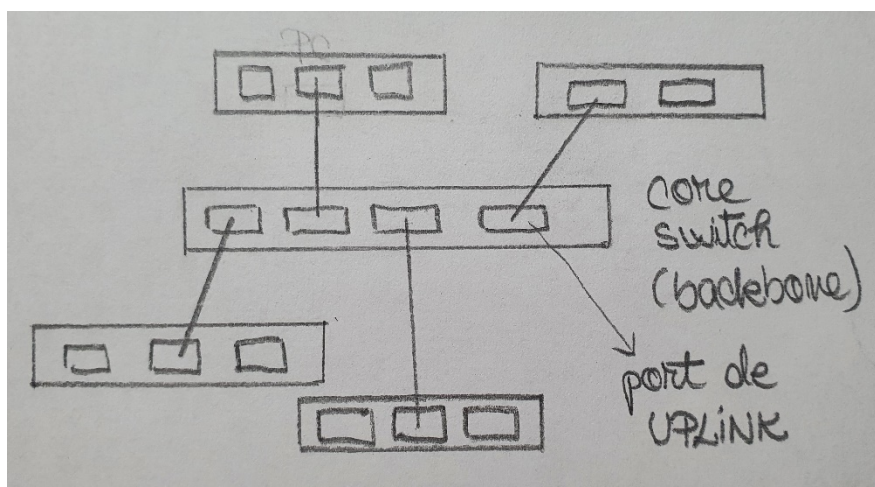


La nivel Data-Link, structura de date care se trimite, conține informația utilă + header-ele de la nivelul DL, poartă numele de *frame (cadru)*. La nivel mai sus în stiva TCP/IP, acest frame se numește *pachet*.

Pe măsură ce calculatorul PC1 vrea să trimită un frame la PC2, switch-ul e capabil să se uite la fiecare frame care circulă pe porturile sale și să vadă de unde vine. Apoi, switch-ul ține minte că în spatele portului 2 se găsește calculatorul cu MAC-ul lui PC2.

Inițial, switch-ul nu știe unde să trimită frame-ul, așa că îl trimite pe toate porturile sale (comportament de HUB). În momentul în care PC2 îi răspunde lui PC1, switch-ul vede că pe portul 1 îi vine frame-uri care are în antet adresa MAC al lui PC2. Astfel, a doua oară când se trimit informații de la PC1 la PC2, se va trimite doar lui, nu tuturor calculatoarelor.

## Rețele cu topologie stea extinsă



- în porturile din switch-ul central pot fi conectate și alte device-uri, nu numai alte switch-uri
- toată rețeaua (și cea cu topologie stea, și cea cu topologie BUS) reprezintă o rețea locală – LAN (Local Area Network)
- în spatele unui port din switch-ul central se găsesc echipamente cu mai multe MAC-uri

## Switch cu management

- are un terminal, o consolă
- i se poate da o adresă IP
- se pot configura porturile, ca ele să meargă la anumite viteze, sau să fie conectate calculatoare cu o anumită adresă MAC
- au o interfață web, pentru administrare

**LAN** (Local Area Network) – de obicei acoperă o organizație, o locație, de dimensiune “rezonabilă”

**MAN** (Metropolitan Area Network) – construită conectând mai multe rețele locale prin intermediul unor echipamente care conectează undeva mai sus de nivelul Data-Link

**WAN** (Wide Area Network) – toată rețeaua unui provider la nivel național

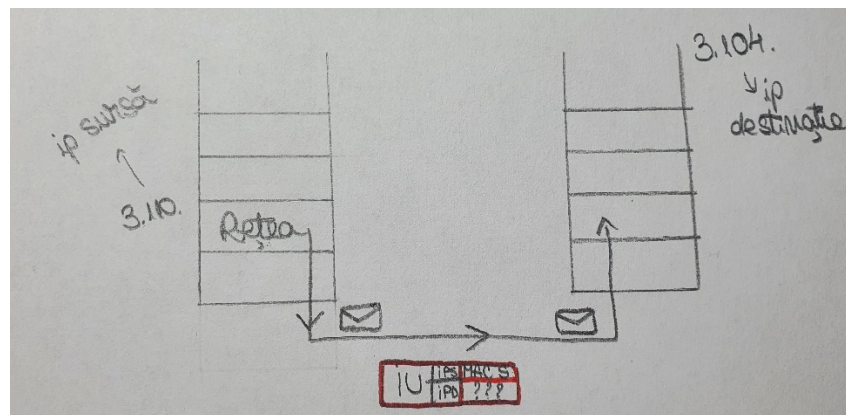
FF:FF:FF:FF:FF:FF

- MAC de broadcast
- util când vrem să comunicăm cu toate echipamentele de la nivelul unui LAN
- în situații în care expeditorul vrea să trimită frame-uri intenționat la toate echipamentele din rețea → se setează ca *MAC destinație* acest MAC de broadcast

### Protocoalele ARP și RARP

Două entități care vor să comunice se identifică prin nume, la nivel Aplicație. La nivel mai jos, ele se identifică prin IP (Rețea) și port (Transport). La nivel Data-Link, ele se identifică prin adrese MAC.

ping 192.168.3.104 – operează la nivel Rețea (merge printr-un protocol special, numit ICMP, undeva între Transport și Rețea)



**ARP (Address Resolution Protocol)** – ajută la determinarea adresei MAC pe baza adresei IP a destinației

Ca să determine adresa MAC a destinației, calculatorul trimite un frame special (**ARP request**) în care pune MAC-ul sursă, și la MAC destinație pune MAC-ul de broadcast. Acest request ajunge la toate calculatoarele și roagă calculatorul cu adresa IP la care vrem să ajungem să “spună” MAC-ul. Calculatorul va răspunde cu un **ARP reply**.

*arp -a* – arată perechile (ip, adresă mac) pe care le cunoaște calculatorul

**Default Gateway** = adresa IP a unei plăci de rețea din router; practic este adresa IP a router-ului căruia trebuie să îi trimit datele ca acestea să părăsească rețeaua locală și să meargă mai departe în Internet.

### **RARP – Reverse Address Resolution Protocol**

- se cunoaște adresa MAC, dar nu cea IP
- află adresa IP pe baza adresei MAC
- nu se mai folosește, a fost înlocuit de DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) – poate oferi mai multe setări de configurare

RARP este perfect pentru un device care tocmai a fost pornit, nu are configurat o adresă IP, și întreabă pe toată lumea din rețea printr-un **RARP request**, folosind adresa MAC de broadcast, ce adresă IP poate folosi.

**Ce setări *minimale* trebuie făcute unui device pentru a putea comunica în Internet?**

- IP
- Netmask (subnet mask)
- Default Gateway (GW)
- DNS (adresa IP a cel puțin unui server DNS)

### **RARP vs. DHCP**

RARP poate da doar adresă IP. DHCP poate configura toate cele patru setări.

**!!!** Serverul DHCP este un proces care rulează pe router (cel mai natural, pentru că router-ul e mereu *up and running*).

**!!!** Serverul DHCP nu își poate da adresă IP lui însuși.