

## CURS 4

192.167.1.17 – format zecimal cu punct

Reprezentarea în memorie a adresei IP:

- **IPv4** – pe 4 octeți (cu valori între 0 și 255)
- **IPv6** – pe 16 octeți

**INET\_ADDR** – convertește adresa IP dată ca *string* (în format zecimal cu punct) în reprezentarea internă.

EX: `inet_addr("192.168.1.17")`

`{"192", "168", "1", "17"} => (atoi) {192, 168, 1, 17}`

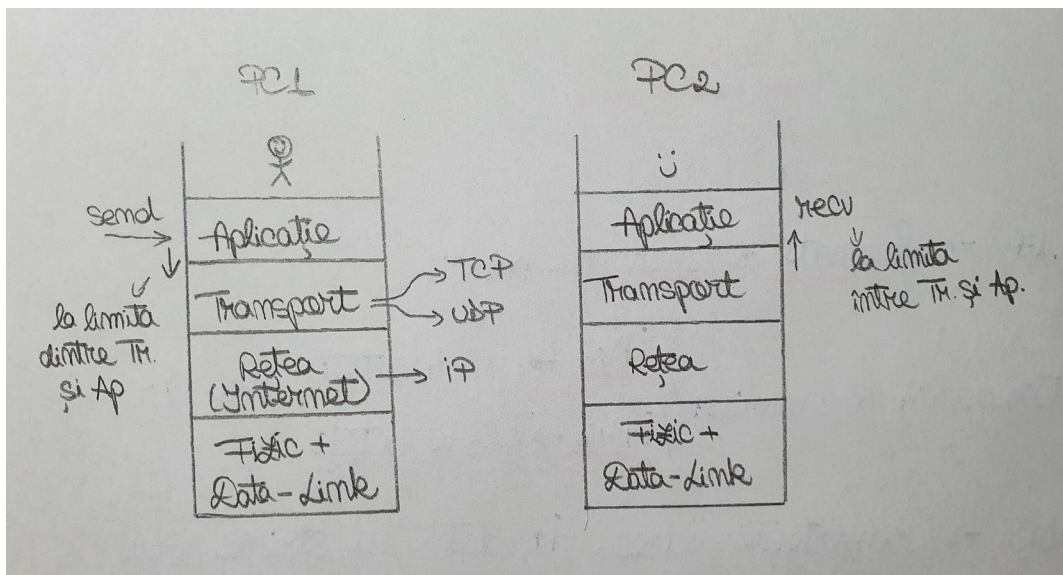
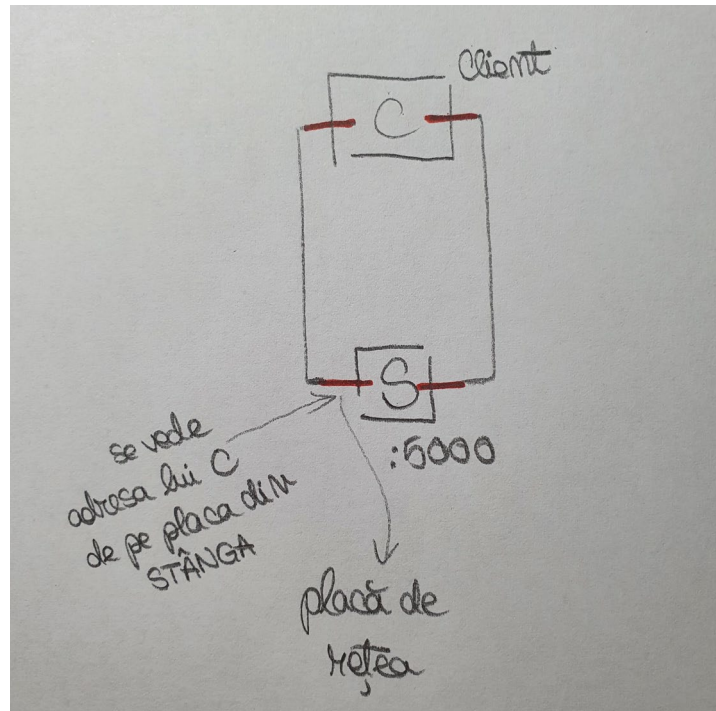
$\rightarrow 192 * 256^3 + 168 * 256^2 + 1 * 256 + 17$

$\rightarrow 192 \ll 24 + 168 \ll 16 + 1 \ll 8 + 17$  (shiftări pe biți)

**INET\_NTOA** – primește ca parametru adresa în *reprezentarea internă* și o convertește la string (**N**etwork **T**O **A**SCII)

*Comenzi terminal:*

- **nc** (**n**etwork **c**at) – arată conținutul rețelei
  - `nc -l 5000` – face un server pe portul 5000 care stă și așteaptă ceva
- **netstat -lnp** – arată toate procesele server care așteaptă pe calculator
  - `-l` : arată toate socket-urile în starea *listen*
  - `-n` : arată *numărul portului* în loc de numele protocolului și *adresa IP* în locul numelui calculatorului
  - `-p` : arată *PID-ul* procesului
- **telnet localhost 5000** – face un connect la serverul de pe localhost cu portul 5000



Orice proces, orice client, server, este localizat la nivelul *Aplicație* a stivei TCP/IP.

Când se trimit pachete, acestea coboară de la nivel la nivel, fiecare “pasând” (folosind *memcpy*) informația utilă de la nivelul superior la nivelul inferior. Fiecare nivel inferior adaugă niște antete/câmpuri în fața informației utile (**headere**). La nivel *minimal*:

- Data-Link: adaugă (MAC sursă, MAC destinație)

- Rețea: adaugă (IP sursă, IP destinație)
- Transport: adaugă (port sursă, port destinație)

!!! Informația utilă nu se copiază de la un nivel la altul

!!! Fiecare nivel vede informația utilă ca fiind informația utilă originală + toate celelalte headere adăugate de nivelurile anterioare

### Ce se întâmplă când se face *send* de mai mulți octeți? (ex. 10k octeți)

Nivelul Transport are grijă să “rupă” acel send în mai multe “pliculețe” (pachete de informație). Dimensiunea unui pachet depinde de mai mulți factori:

- **MTU** (Maximum Transfer Unit): pentru cablul UTP – 1500 octeți (dimensiunea informației utile + toate headerele)
- **Protocolul de la nivel Transport** (64 kilo)
  - nivelul transport rupe în plicuri de maxim 64 kilo
  - nivelul data-link ia plicul, îl rupe în mai multe pachetele mai mici, și la fiecare pachet adaugă header-ul care se adaugă la nivel DL

### Diferența dintre TCP și UDP

#### TCP

- protocolul orientat pe conexiune
- asigură livrarea sigură datelor de la sursă la destinație (*reliable*)
- la nivelul Transport, se împart datele în pachete și se “numerotează”
- se trimit datele la nivelul Rețea și tot mai jos, până ajung în cealaltă parte, la nivel Transport; datele trebuie pregătite, la nivel Transport, deoarece la receptor, tot la nivelul Transport, trebuie rearanjate exact în ordinea trimisă
- se asigură că informația este aranjată exact în ordinea în care a fost trimisă
- dacă lipsește un pachet, receptorul anunță ce pachet nu a primit, prin niște confirmări (implicite sau cumulate) → emițătorul retrimite pachetele neprimite (nu instant, se mai trimit alte pachete anticipat, ca să se verifice dacă nu cumva acel pachet pierdut nu ajunge cu întârziere)

#### UDP

- datele se pot pierde (*nu este reliable*)
- nu se fac toate verificările ce se fac la nivel Transport în cazul TCP
- este mai rapid decât TCP, datorită faptului că nu se fac atâtea verificări

Protocoale care folosesc UDP: cele care se bazează pe mecanism cerere-răspuns (ceea ce cere clientul serverului încapă într-un singur plic, și răspunsul serverului încapă și el într-un singur plic) – *NTP* (Network Time Protocol), *DNS* (Domain-Name Server) – ambele sunt de la nivel Aplicație