

2011 - 2012

Programare Orientata spre Obiecte (*Object-Oriented Programming*)

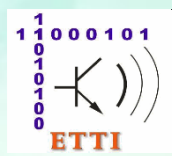
a.k.a. Programare Obiect-Orientata

Titular curs: Eduard-Cristian Popovici

Suport curs: <http://discipline.elcom.pub.ro/POO-Java/>

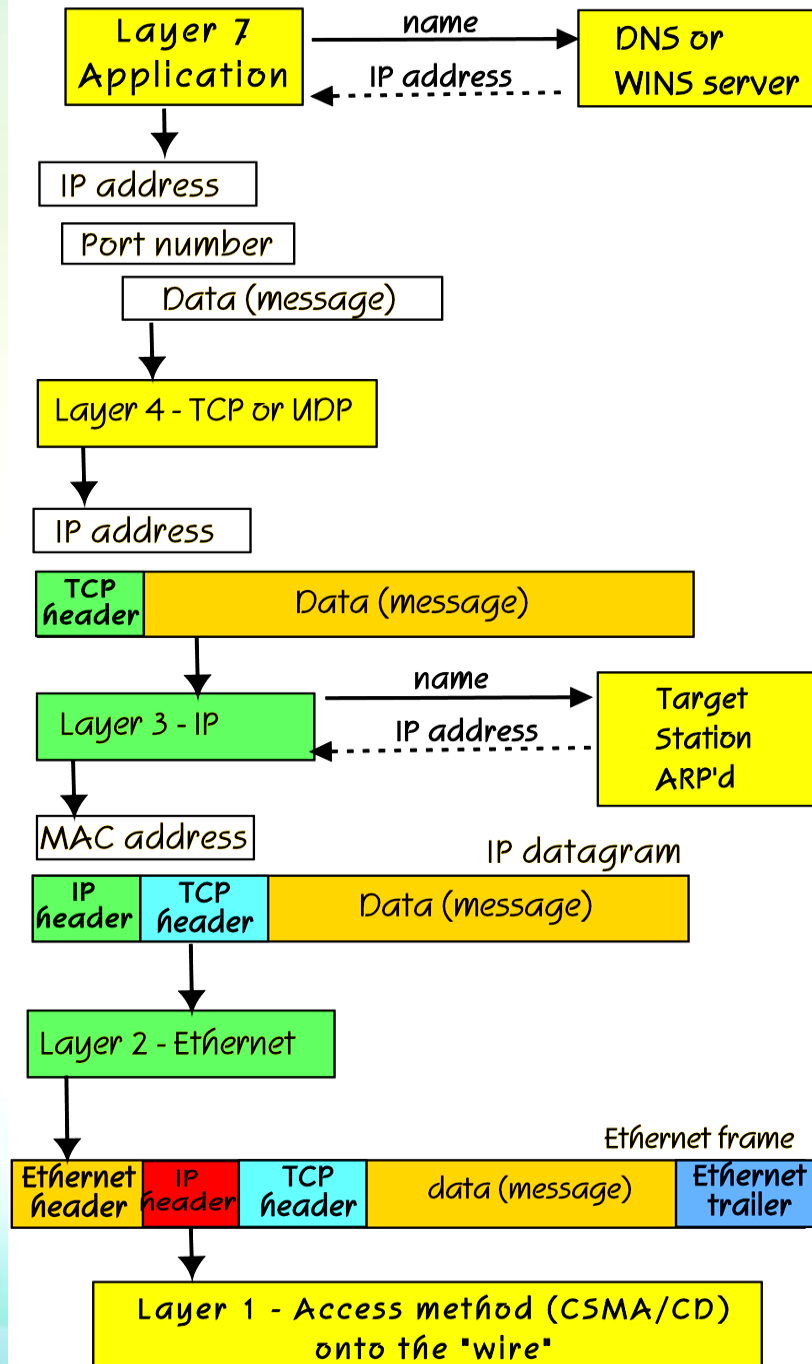
3. Programarea la nivel socket cu Java

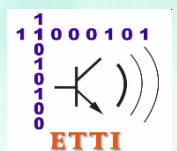
3.2. Introducere in Protocolul Internet (IP) si stiva IP



Stiva Internet

Summary of the TCP/IP Stack



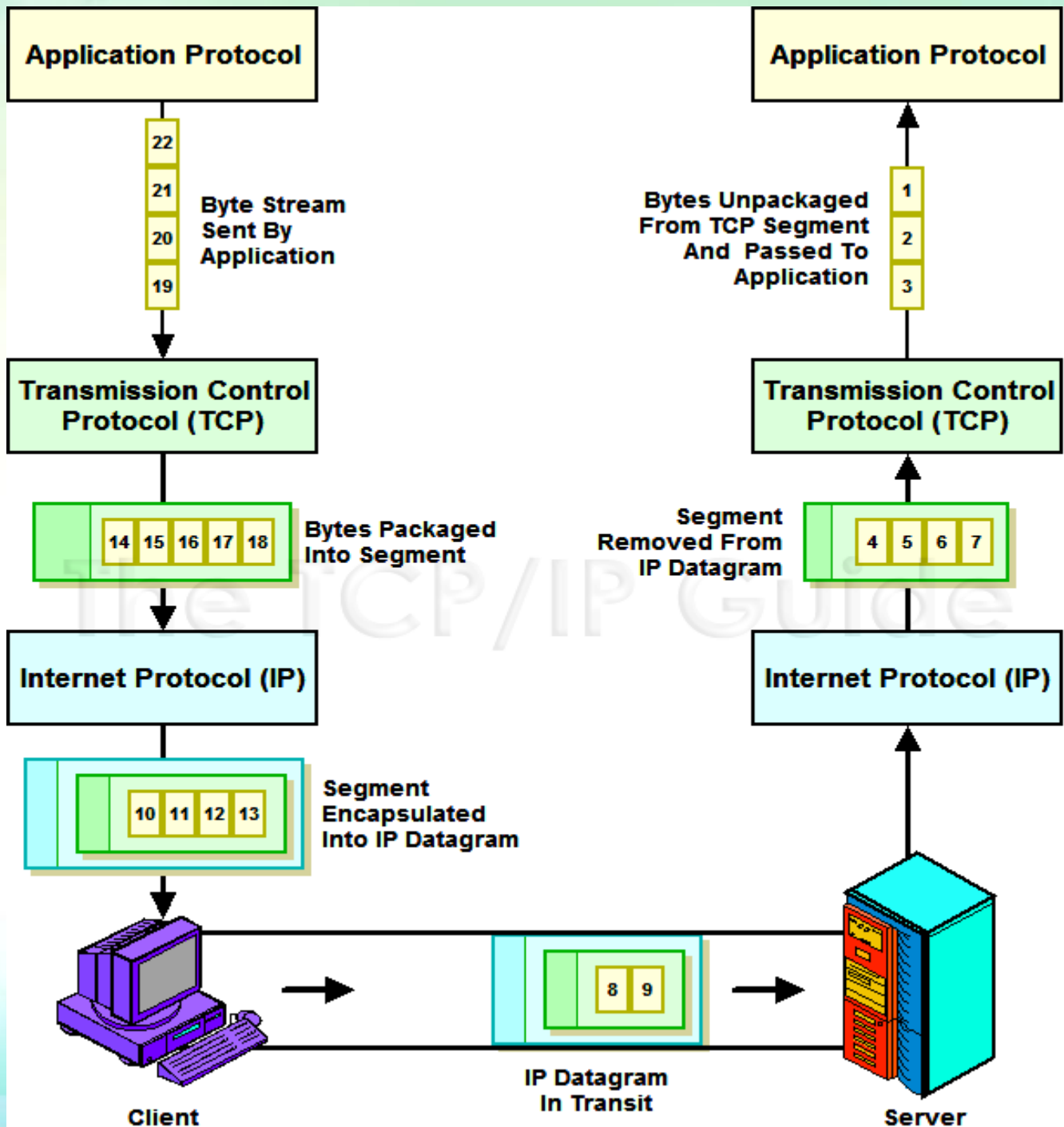


Stiva Internet

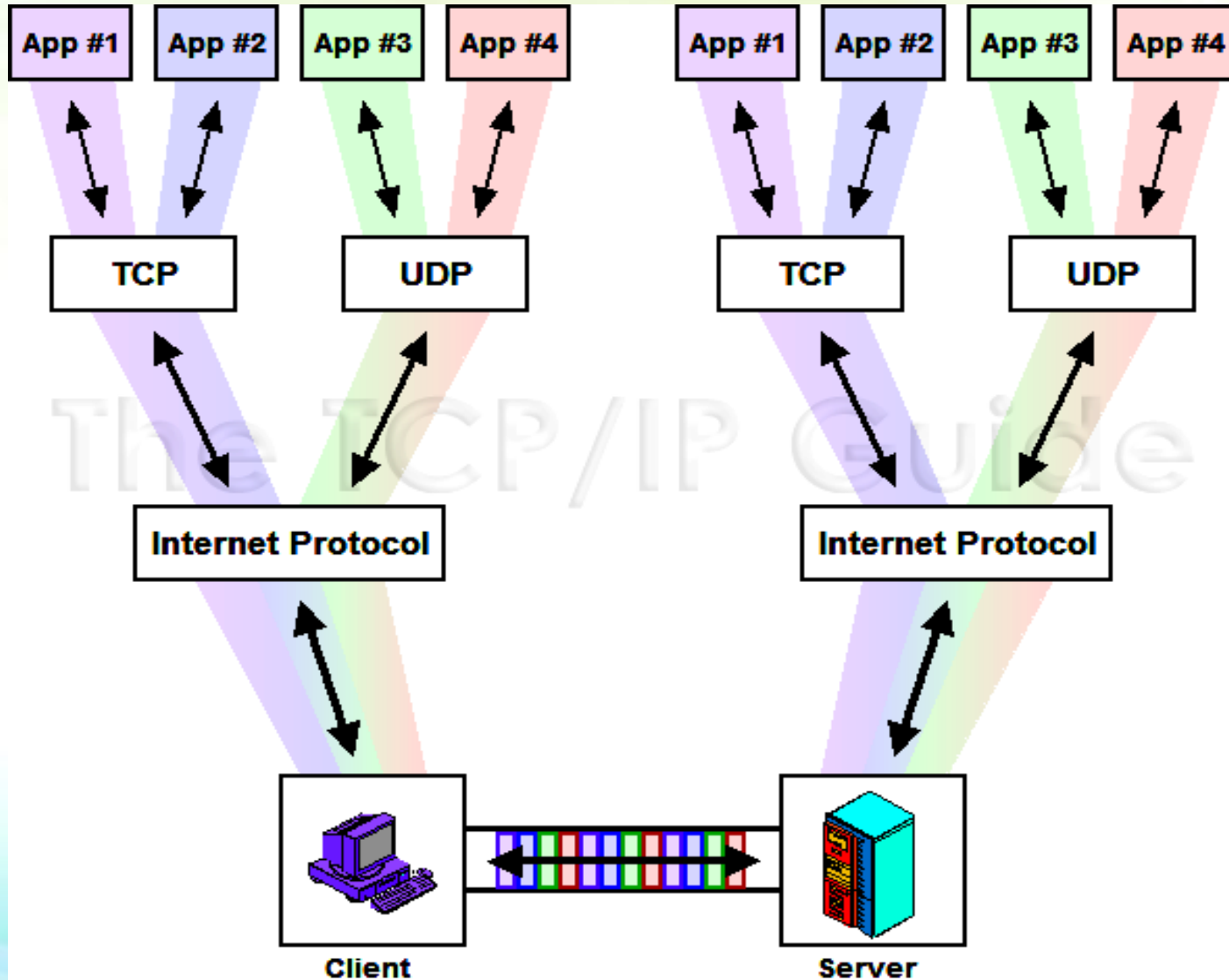
Impachetarea octetilor din fluxul de date ai Aplicatiei in segmente TCP

Incapsularea segmentelor in Datagrame IP

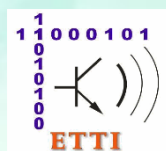
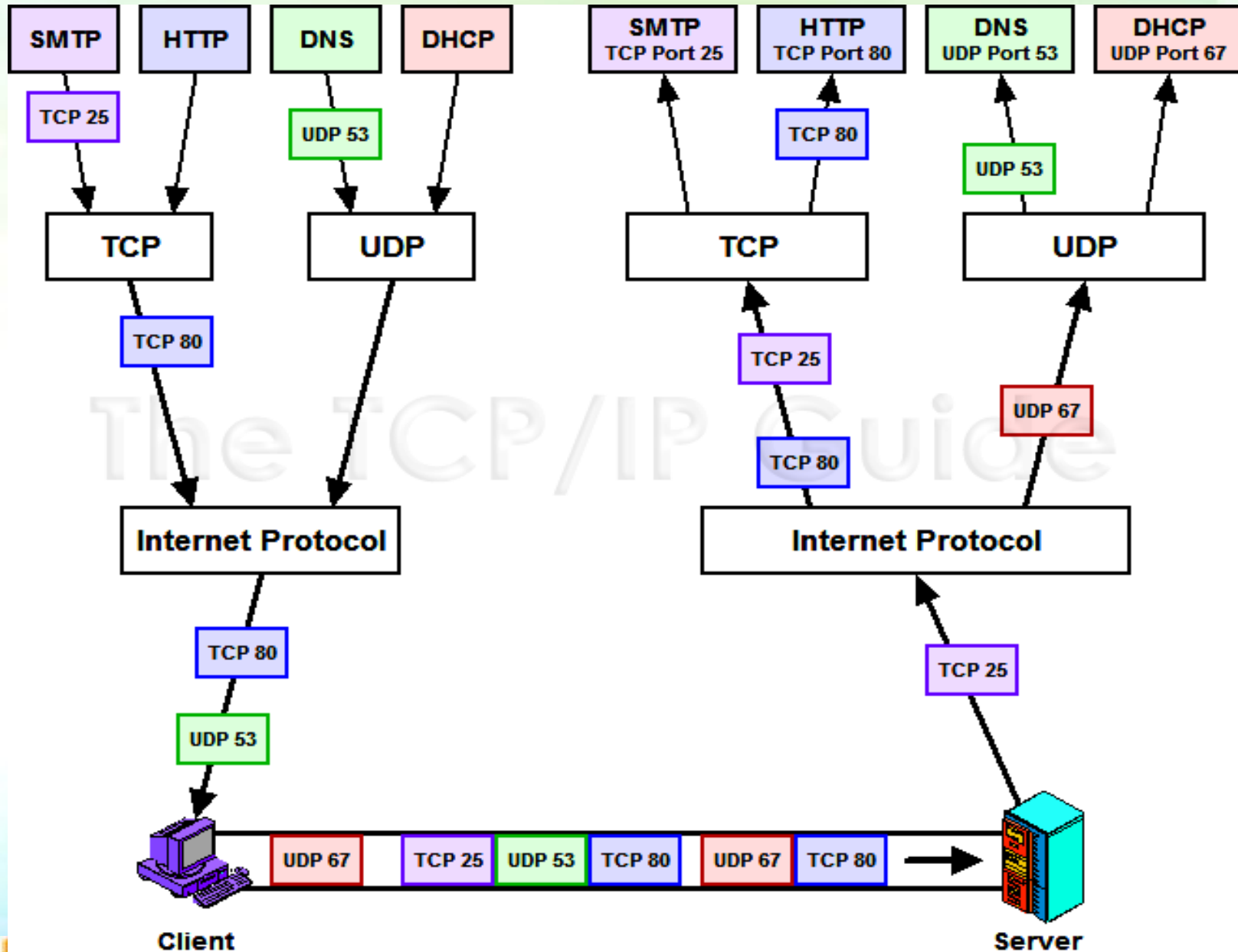
Transmisia Datagramelor IP prin Internet



Stiva Internet - Modul in care aplicatiile folosesc protocoalele TCP si UDP

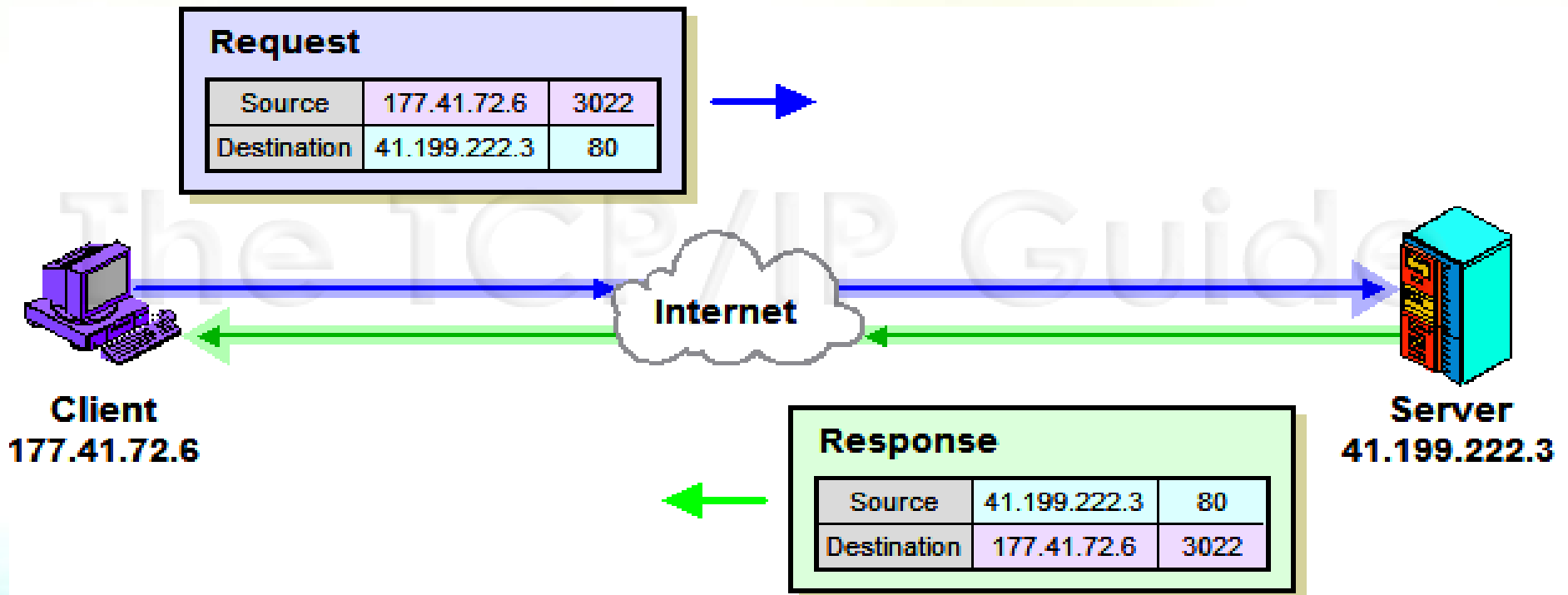


Introducere in Protocolul Internet (IP) si stiva IP



Utilizarea adreselor IP pentru a identifica entitatile din Internet

Porturile (80 al server-ului si 3022 al client-ului) permit identificarea proceselor / protoacoalelor la nivel aplicatie

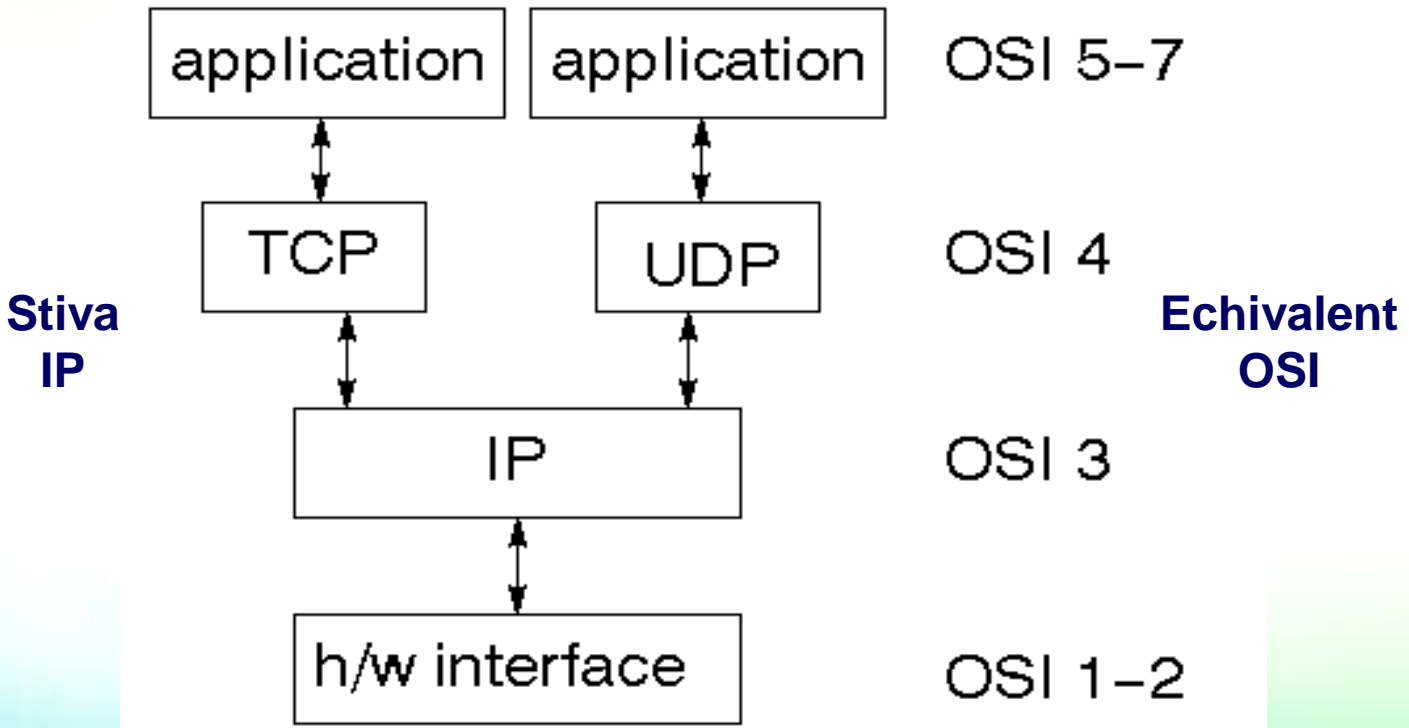




Stiva Internet si rolul *socket-urilor*

Socket-ul

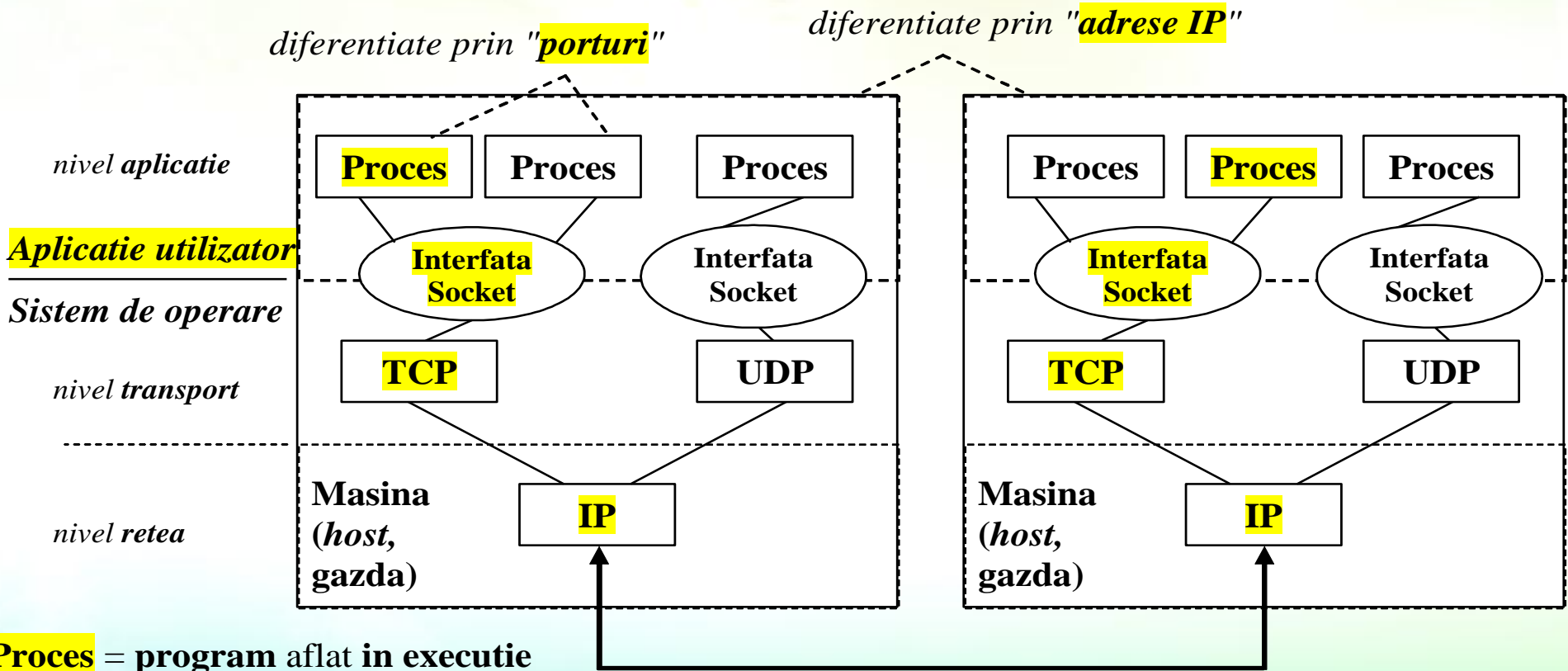
- este **punct final** al unei **comunicatii** intre procese
- ofera un **punct de acces** la **servicii de nivel transport (TCP sau UDP)** in Internet





Socket-urile

- puncte finale al comunicatiilor intre procese
- puncte de acces la servicii de nivel transport (TCP sau UDP) in Internet

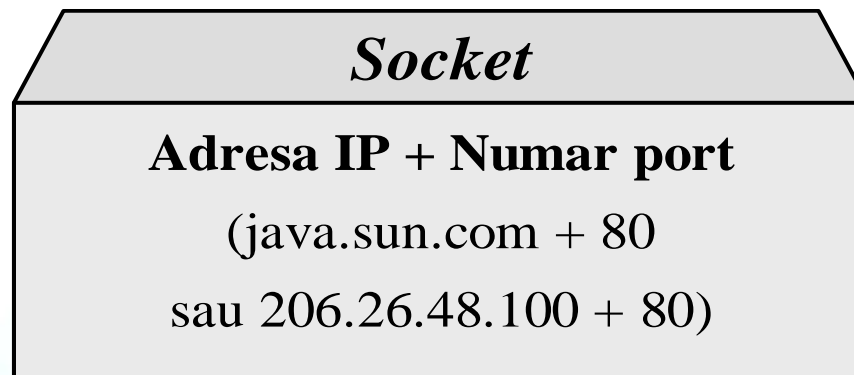




Socket-urile

Adresa socket intr-o retea bazata pe IP consta din **doua parti**:

- **adresa IP**, pe 32 biti (4 octeti), reprezentata ca
 - **sir de 4 valori intre 0 si 255** despartite prin puncte (ex. 206.26.48.100)
 - sau ca **alias** numele masinii si domeniului (ex. java.sun.com).
- **numarul de port** (identificatorul portului), pe 16 biti (2 octeti), **distinct pentru fiecare tip de protocol** (TCP si UDP)





Incapsularea adreselor IP in limbajul Java

Clasa **InetAddress** **incapsuleaza o adresa IP** intr-un obiect

Obiectul **poate intoarce informatia utila** daca ii invocam metodele

- de exemplu, **equals()** intoarce adevarat daca doua obiecte reprezinta aceeasi adresa IP

Clasa **InetAddress** **nu are constructor public**

- **pentru a crea obiecte** ale acestei clase trebuie invocata una dintre **metodele de clasa** (declarate **static**)

getByAddress() sau

getByName().



Incapsularea adreselor IP in limbajul Java

O adresa IP speciala este **adresa IP *loopback***

- tot ce este trimis catre aceasta adresa IP se intoarce si devine intrare IP pentru gazda locala
- cu ajutorul careia pot fi testate local programe care utilizeaza *socket-uri*.

Pentru a identifica adresa IP *loopback* sunt folosite numele "**localhost**" si valoarea numerica "**127.0.0.1**".

Pentru a obtine `InetAddress` care incapsuleaza adresa IP *loopback* pot fi folosite apelurile echivalente:

```
InetAddress.getByName (null)
InetAddress.getByName ("localhost")
InetAddress.getByName ("127.0.0.1")
```

Metoda `getAddress ()` returneaza octetii adresei IP incapsulate, ceea ce poate fi util pentru filtrarea adreselor.