

2009 - 2010

Tehnologii de Programare in Internet (TPI / RST)

Titulari curs: **Mihnea Magheti**, Eduard-Cristian Popovici

Suport curs: <http://discipline.elcom.pub.ro/tpi/>

Moodle: <http://electronica07.curs.ncit.pub.ro/course/category.php?id=3>

Structura cursului

Continut curs TPI

1. Introducere in tehnologiile Internet

2. Introducere in tehnologiile desktop (SE) Java

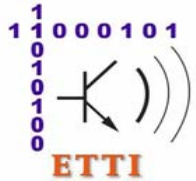
- 2.1. Elemente de baza. Tipuri de date referinta. Clase de biblioteca
- 2.2. Clase pentru fluxuri de intrare-iesire (IO)

3. Programarea la nivel socket in Java

- 3.1. Introducere in Protocolul Internet (IP) si stiva de protocoale IP
- 3.2. Socketuri flux (TCP) Java si programe multifilare (threads)
- 3.3. Socketuri datagrama (UDP) Java

4. Tehnologii Java de programare a aplicatiilor Web (EE) Java

- 4.1. Tehnologii client. Miniaplicatii Java (applet-uri)
- 4.2. Clase pentru interfete grafice cu utilizatorul (AWT, Swing)
- 4.3. Platforma Java EE. Arhitectura si tehnologiile implicate
- 4.4. Tehnologii server. Tehnologia Java Servlet
- 4.5. Tehnologia Java ServerPages (JSP)
- 4.6. Accesul la baze de date prin tehnologii Java (JDBC, Hibernate)
- 4.7. Tehnologii avansate (frameworks, componente EJB, Servicii Web)



Structura cursului

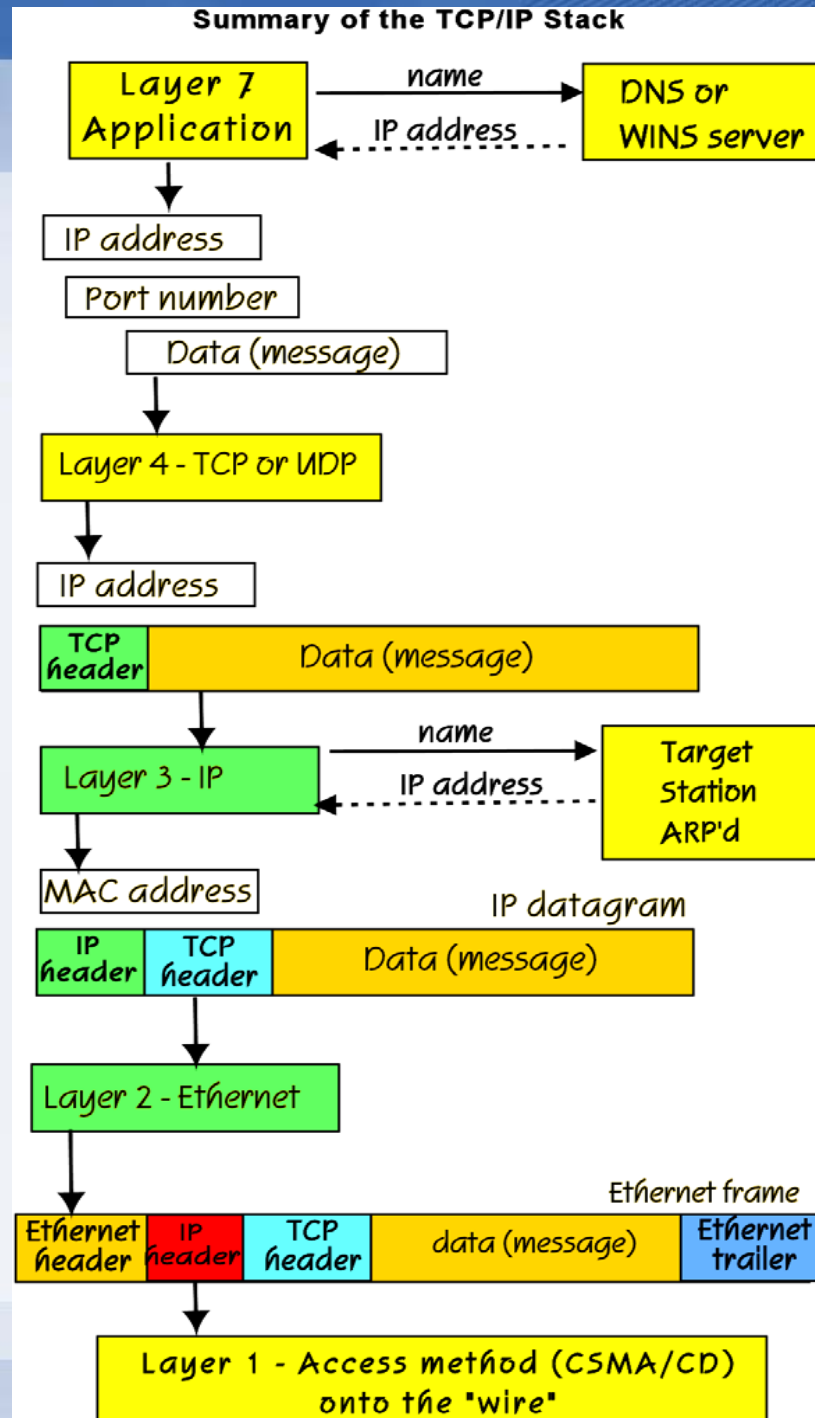


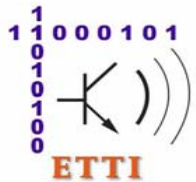
3. Programarea la nivel socket in Java

3.1. Introducere in Protocolul Internet (IP) si stiva de protocoale IP



Stiva de protocoale Internet (IP)

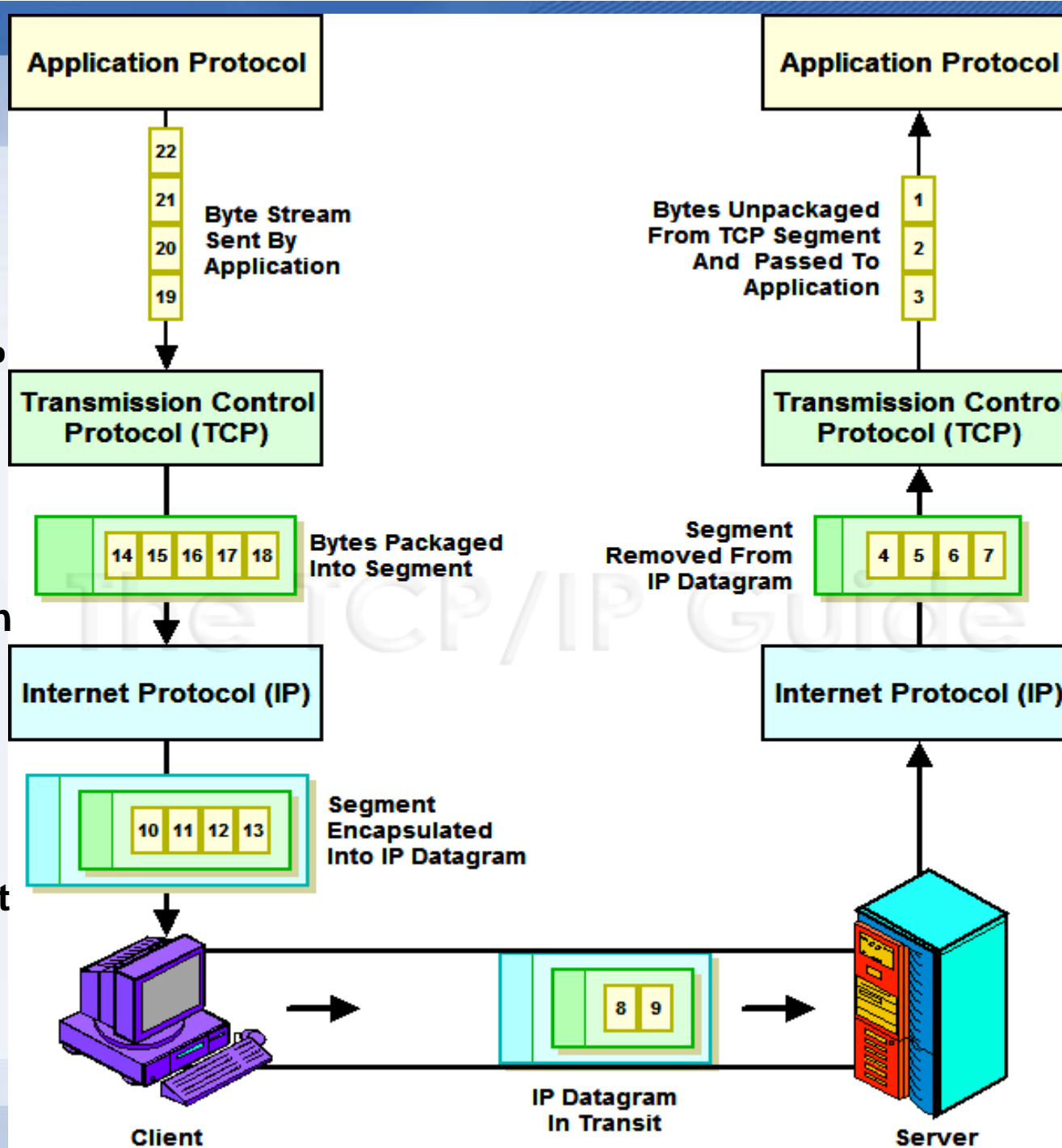




Impachetarea octetilor din fluxul de date ai Aplicatiei in segmente TCP

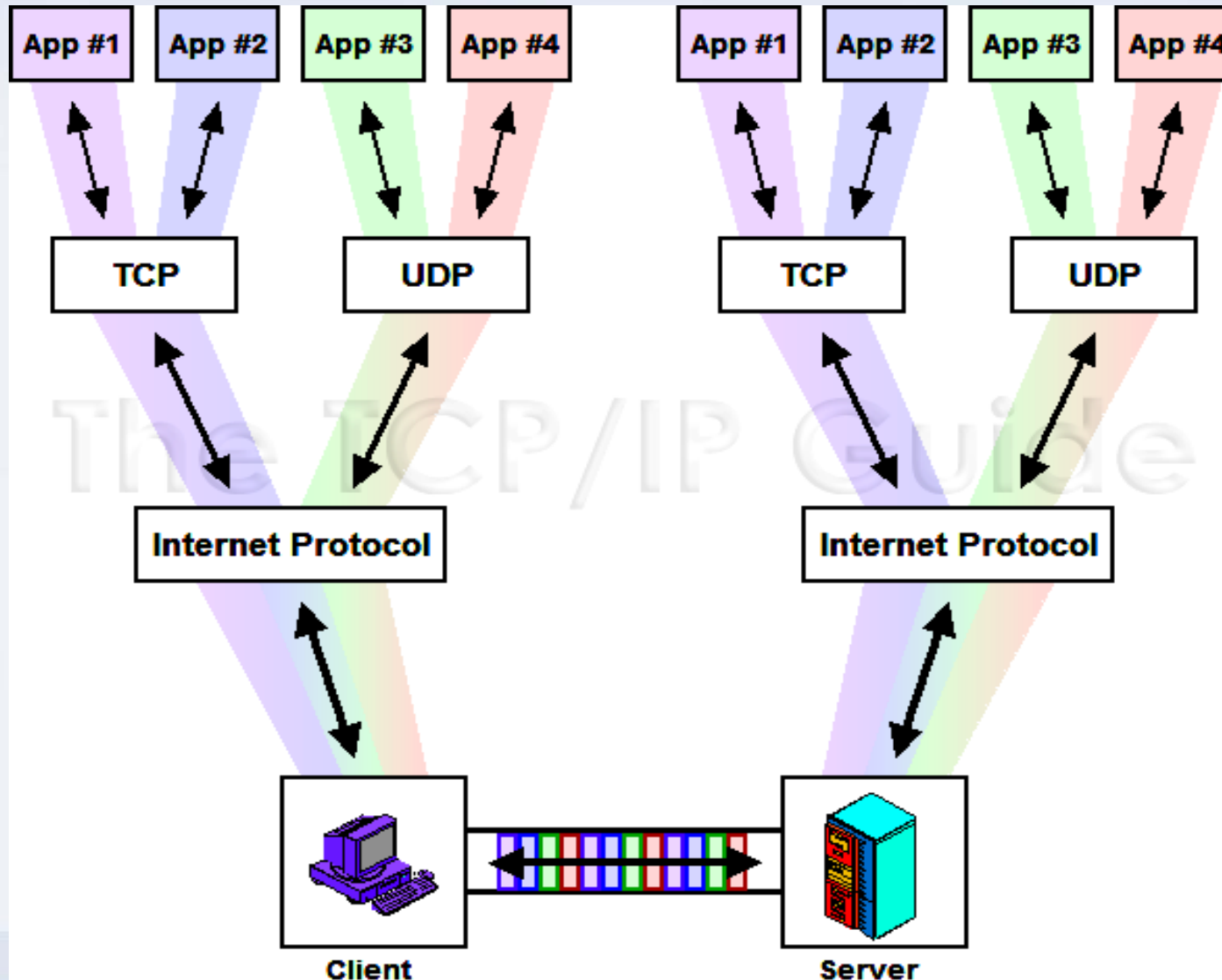
Incapsularea segmentelor in Datagrame IP

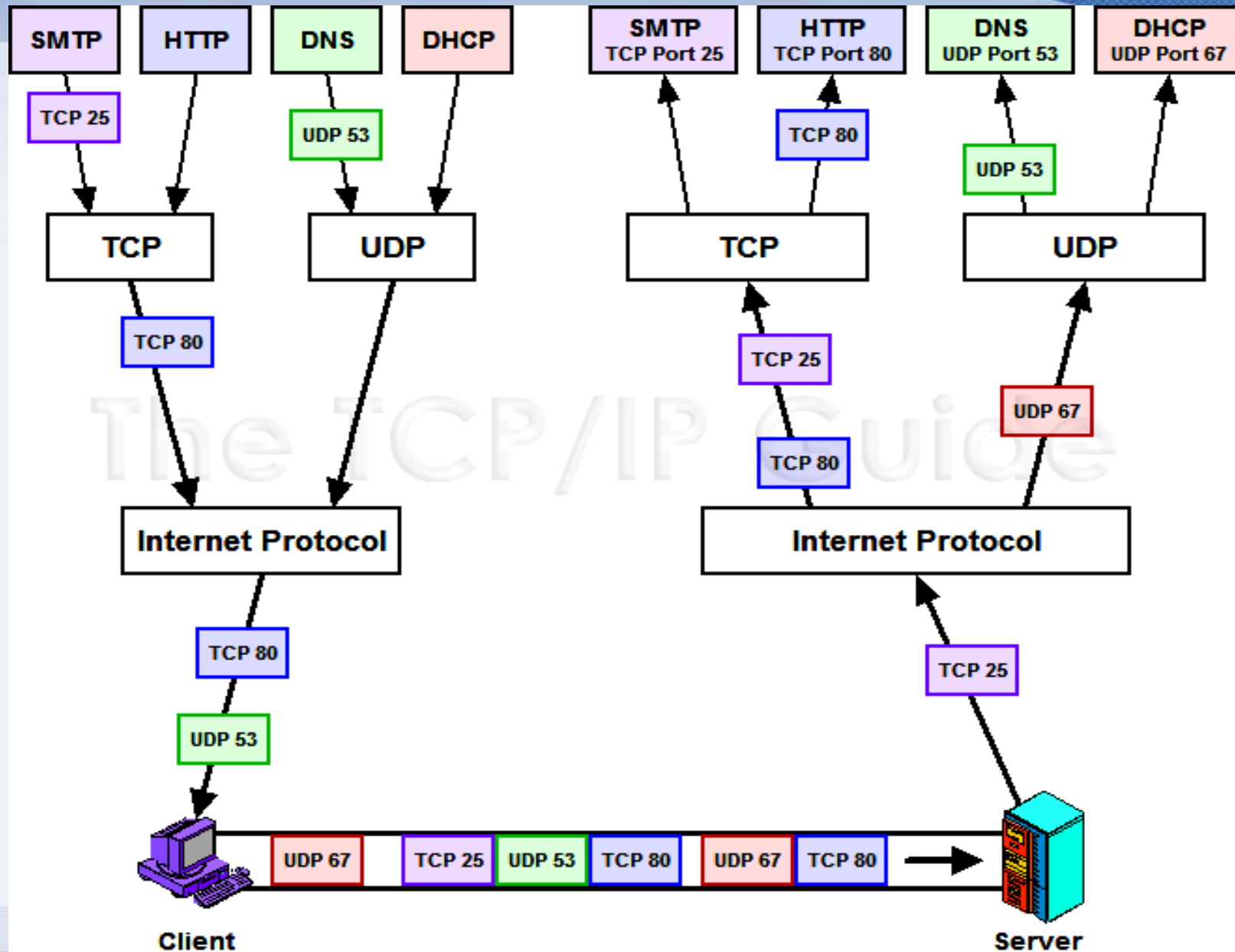
Transmisia Datagramelor IP prin Internet



3.1. Introducere in stiva protocoale IP

Modul in care aplicatiile folosesc protocoalele TCP si UDP





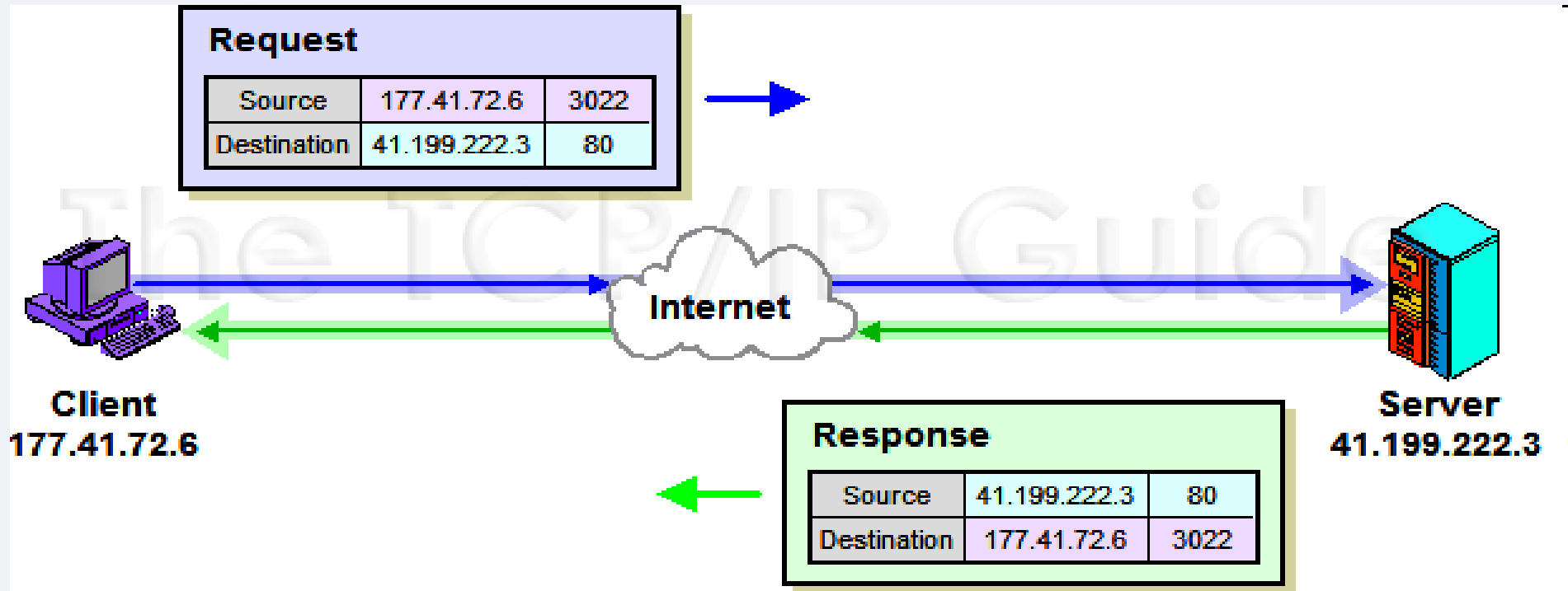
The TCP/IP Guide



3.1. Introducere in stiva protocoale IP

Utilizarea adreselor IP pentru a identifica entitatile din Internet

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

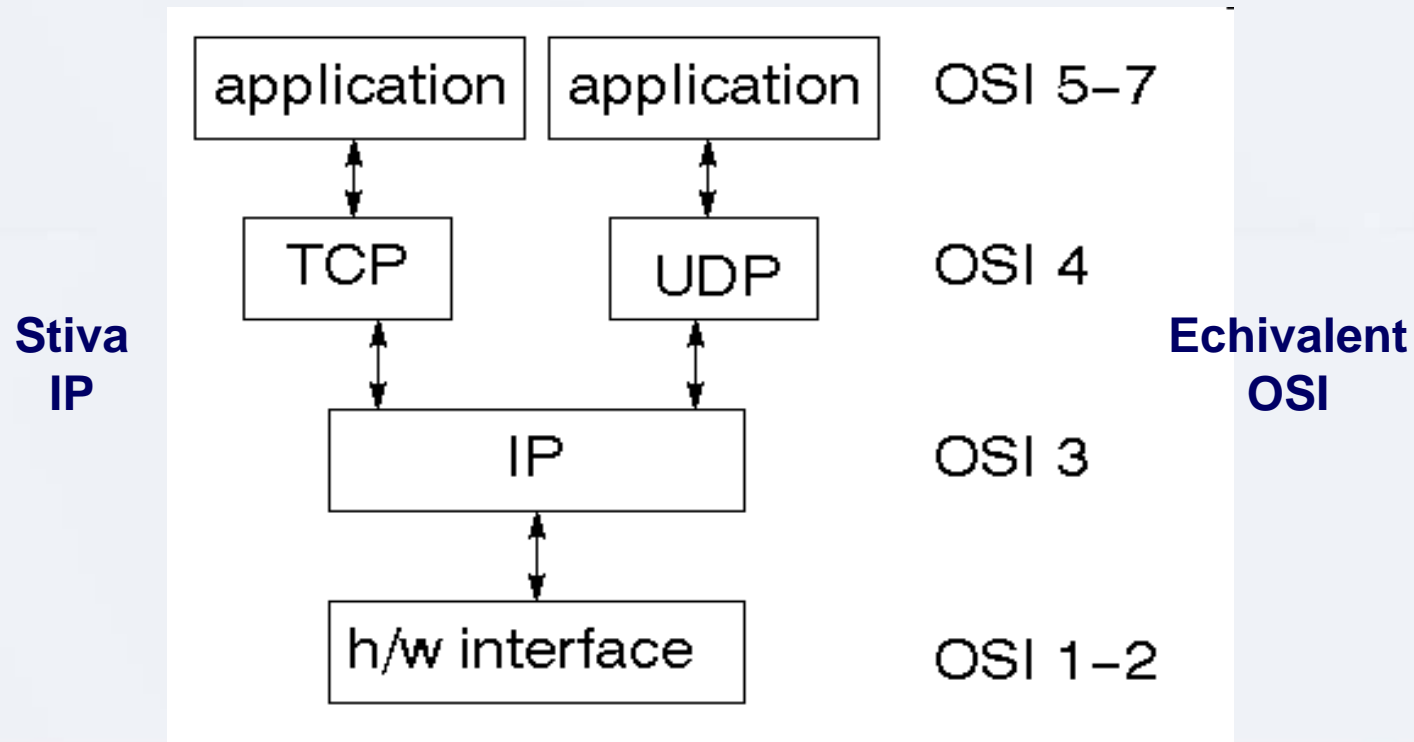


3.1. Introducere in stiva protocoale IP

Stiva Internet si rolul *socket-urilor*

Socket-ul

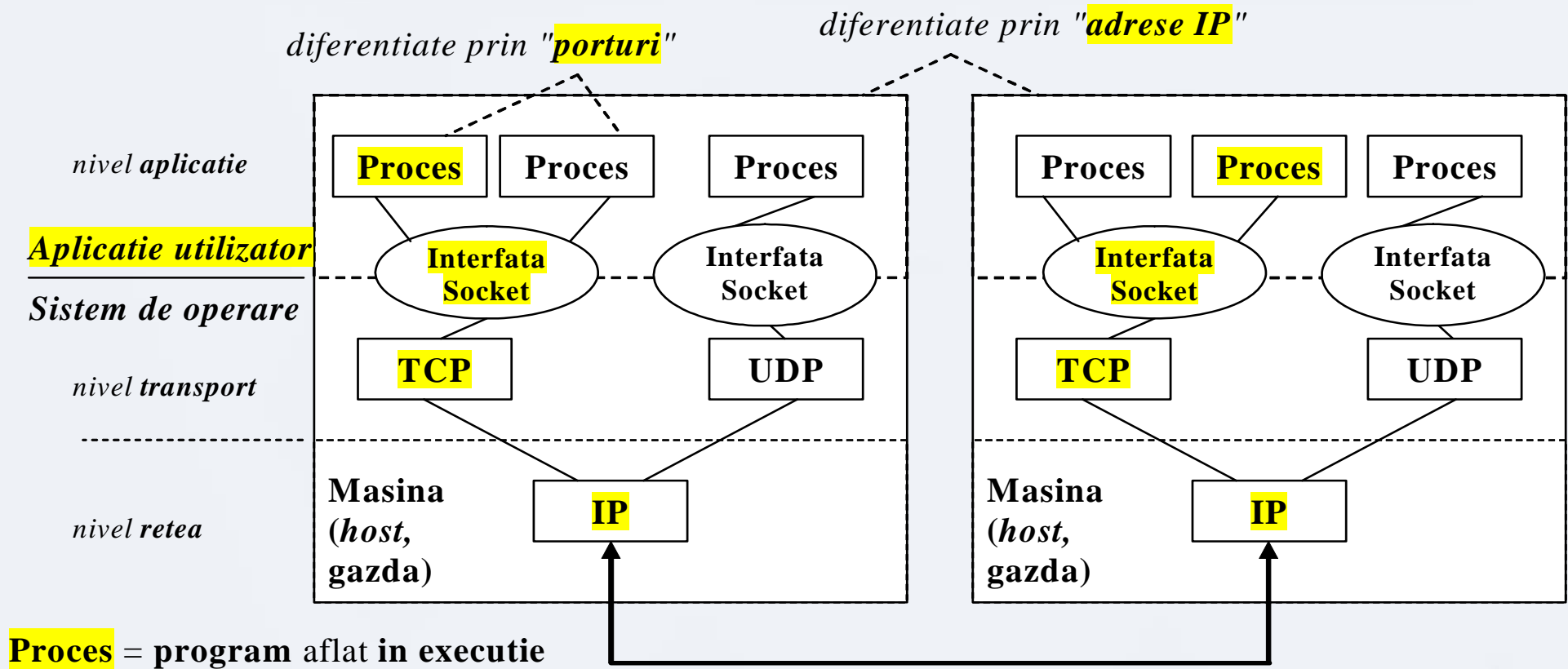
- este **punct final** al unei **comunicatii** intre **procese**
- ofera un **punct de acces** la **servicii de nivel transport (TCP sau UDP)** in **Internet**



3.1. Introducere in stiva protocoale IP

Socket-urile

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

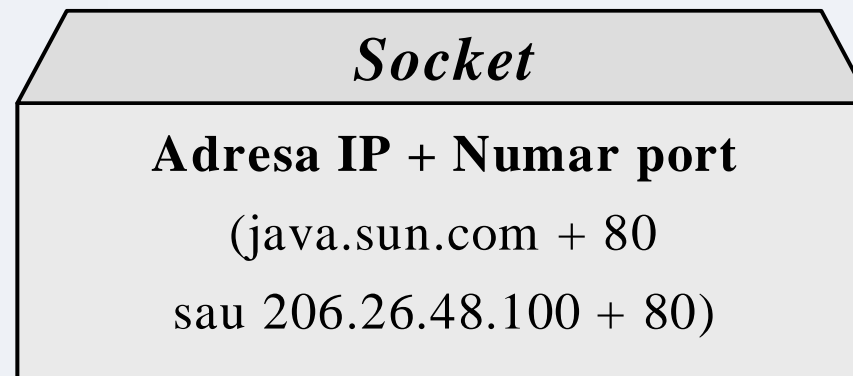


3.1. Introducere in stiva protocoale IP

Socket-urile

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

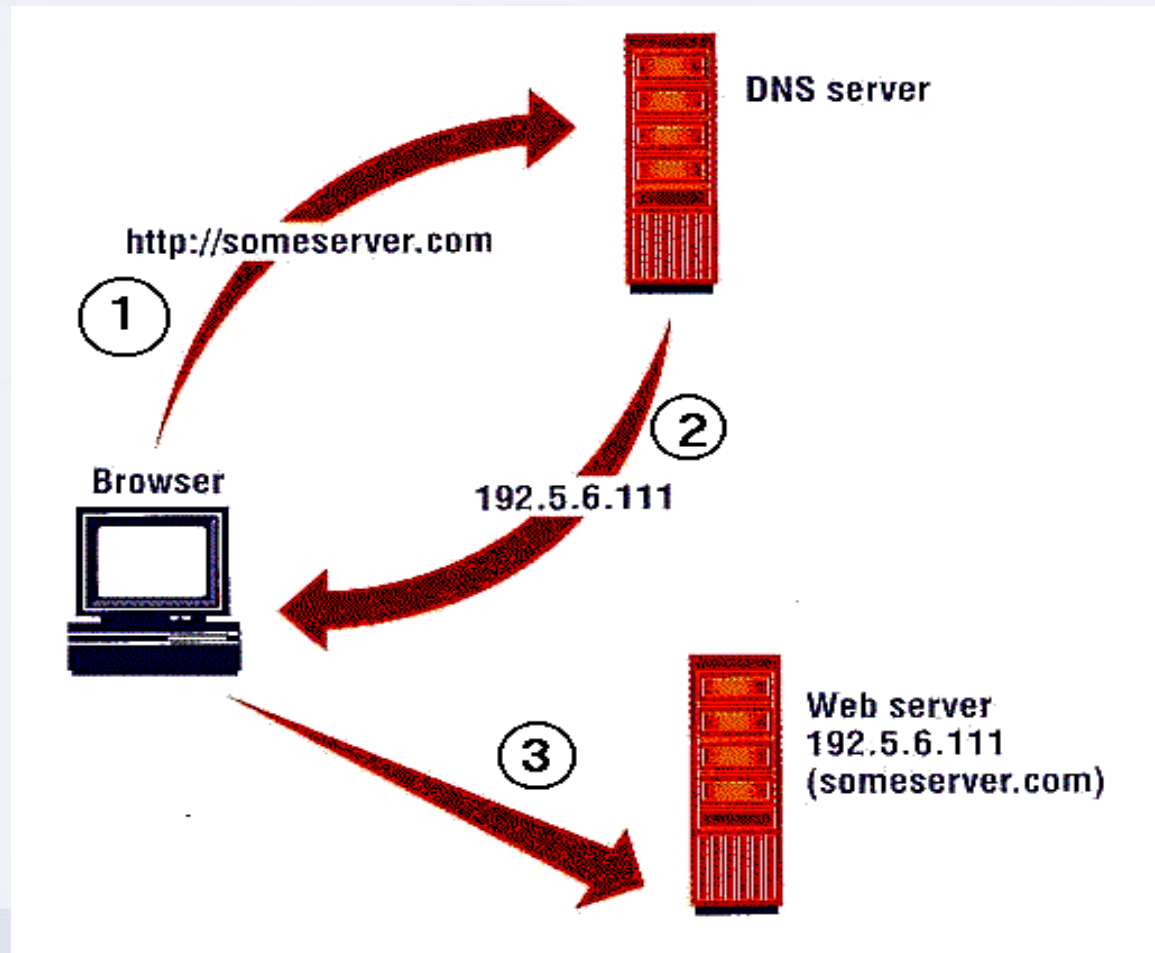
- **adresa IP**, pe 32 biti (4 octeti), reprezentata ca
 - **sir de 4 valori intre 0 si 255** despartite prin puncte (ex. 206.26.48.100)
 - sau ca **alias** numele masinii si domeniului (ex. java.sun.com).
- **numarul de port** (identificatorul portului), pe 16 biti (2 octeti), **distinct pentru fiecare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



3.1. Introducere in stiva protocoale IP

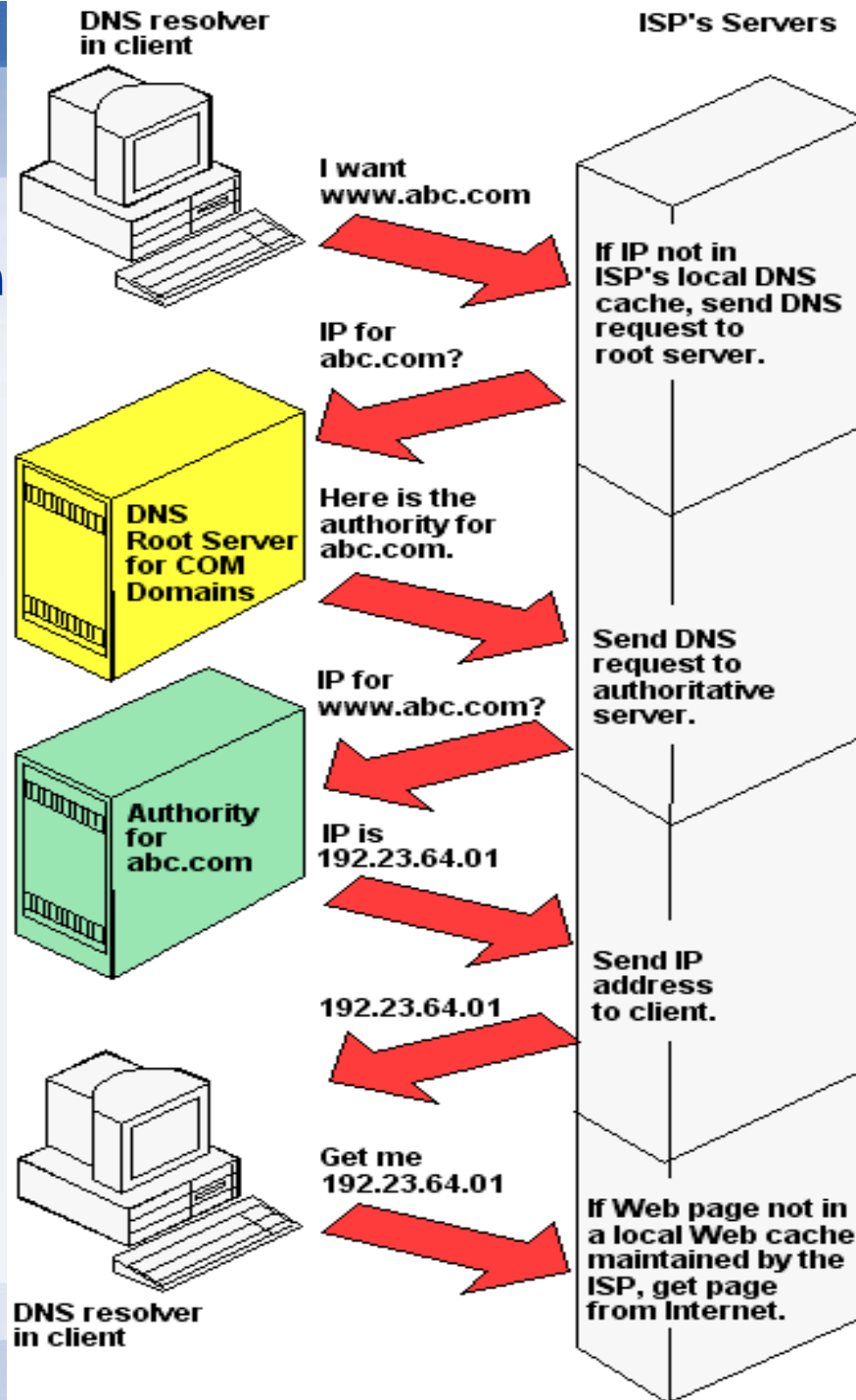
Modelul comunicatiilor in Internet

Utilizarea serviciului DNS (*Domain Name Server*) pentru obtinerea adresei numerice IP echivalenta numelui masinii de calcul



Modelul comunicatiilor in Internet

Utilizarea serviciului DNS



3.1. Introducere in stiva protocoale IP

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()`

3.1. Introducere in stiva protocoale IP

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**".

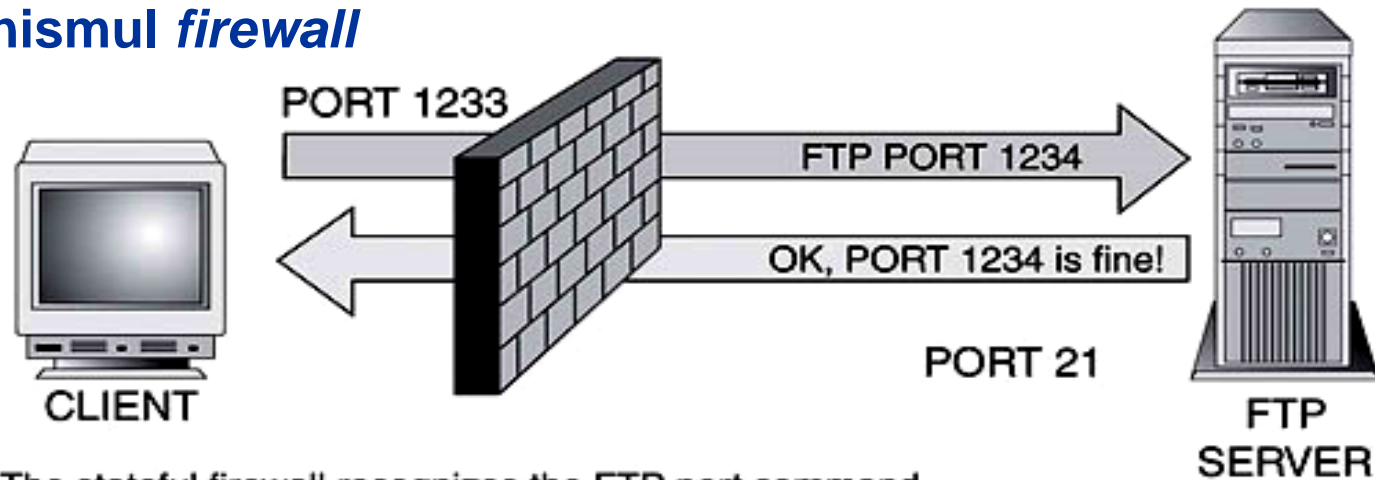
Pentru a obtine **InetAddress** care incapsuleaza adresa IP *loopback* pot fi folosite apelurile echivalente:

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

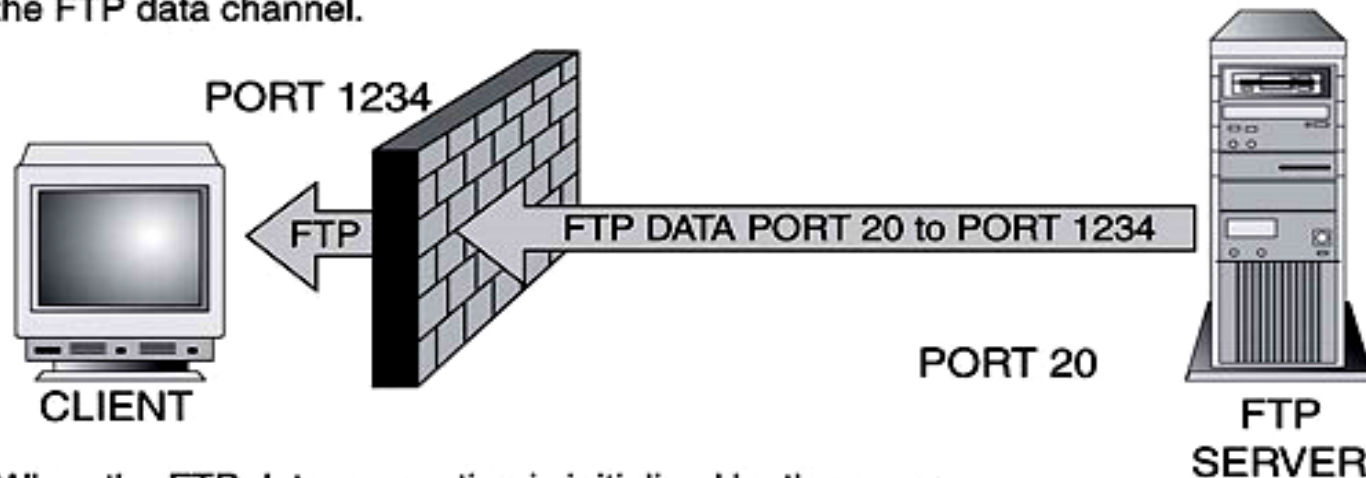
Metoda [getAddress\(\)](#) returneaza octetii adresei IP incapsulate, ceea ce poate fi util pentru filtrarea adreselor.

3.1. Introducere in stiva protocoale IP

Mecanismul *firewall*



The stateful firewall recognizes the FTP port command and records its value to securely facilitate the creation of the FTP data channel.



When the FTP data connection is initialized by the server, the firewall recognizes the IP address and port combination used and allows the inbound connection to pass.