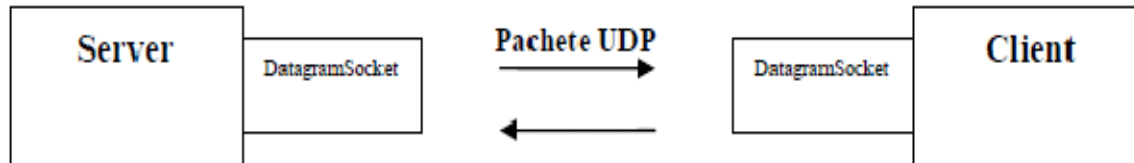


## Aplicație simplă client/server neorientată spre conexiune

Atât clientul cât și serverul folosesc obiecte de tipul **DatagramSocket**.



Un **server** care folosește protocolul UDP trebuie:

1. Să creeze un obiect **DatagramSocket** asociat cu un număr de port specificat

*Exemplu:*

```
DatagramSocket socket = new DatagramSocket(8400);
```

2. Să creeze un obiect **DatagramPacket** utilizând constructorul pentru datagrame pentru recepționarea datelor

*Exemplu:*

```
byte[] buf = new byte[256];
DatagramPacket mesaj = new DatagramPacket(buf, buf.length);
```

3. Cu ajutorul metodei **receive()** a clasei **DatagramSocket** se salvează o datagramă în obiectul de tip **DatagramPacket**

*Exemplu:*

```
socket.receive(mesaj);
```

4. Se procesează cererea

5. Să creeze un obiect **DatagramPacket** utilizând constructorul pentru datagrame pentru transmiterea datelor.

- a) Se identifică adresa și portul de la care vine cererea

*Exemplu:*

```
InetAddress adresa = mesaj.getAddress();
int port = mesaj.getPort();
```

- b) Se construiește răspunsul

*Exemplu:*

```
byte[] buf2 = new byte[256];
buf2 = "...".getBytes();
```

- c) se completează datagrama care va fi trimisă

*Exemplu:*

```
DatagramPacket raspuns =
    new DatagramPacket(buf2, buf2.length, adresa, port);
```

6. Cu ajutorul metodei **send()** a clasei **DatagramSocket** se trimite datagrama completată anterior.

*Exemplu:*

```
socket.send(raspuns);
```

**La partea de client**, se implementeaza aceleasi etape, cu mentiunea ca mai intai se trimite o datagrama, dupa care se asteapta raspunsul serverului.

#### **1. Se creaza un obiect DatagramSocket**

a) Se specifica adresa IP si portul pe care ruleaza serverul

*Exemplu:*

```
InetAddress adresa =  
InetAddress.getLocalHost().getHostName();  
int port=8400;
```

b) Se instantiaza clasa DatagramSocket

*Exemplu:*

```
DatagramSocket socket = new DatagramSocket();
```

**2. Se creaza un obiect DatagramPacket** utilizand constructorul pentru datagrame pentru transmisia datelor.

a) Se construiesc raspunsul

*Exemplu:*

```
byte[] buf = new byte[256];  
buf = ("...").getBytes();
```

b) se completeaza datagrama care va fi trimisa

*Exemplu:*

```
DatagramPacket mesaj =  
new DatagramPacket(buf, buf.length, adresa, port);
```

**3. Cu ajutorul metodei send()** a clasei DatagramSocket se trimite datagrama completata anterior.

*Exemplu:*

```
socket.send(mesaj);
```

#### **4. Se asteapta raspunsul serverului**

**5. Sa creeze un obiect DatagramPacket** utilizand constructorul pentru datagrame pentru receptionarea datelor

*Exemplu:*

```
byte[] buf = new byte[256];  
DatagramPacket raspuns= new DatagramPacket(buf,  
buf.length);
```

**6. Cu ajutorul metodei receive()** a clasei DatagramSocket se salvează o datagramă în obiectul de tip DatagramPacket

*Exemplu:*

```
socket.receive(raspuns);
```