
Programovanie 5

Technológie Java EE



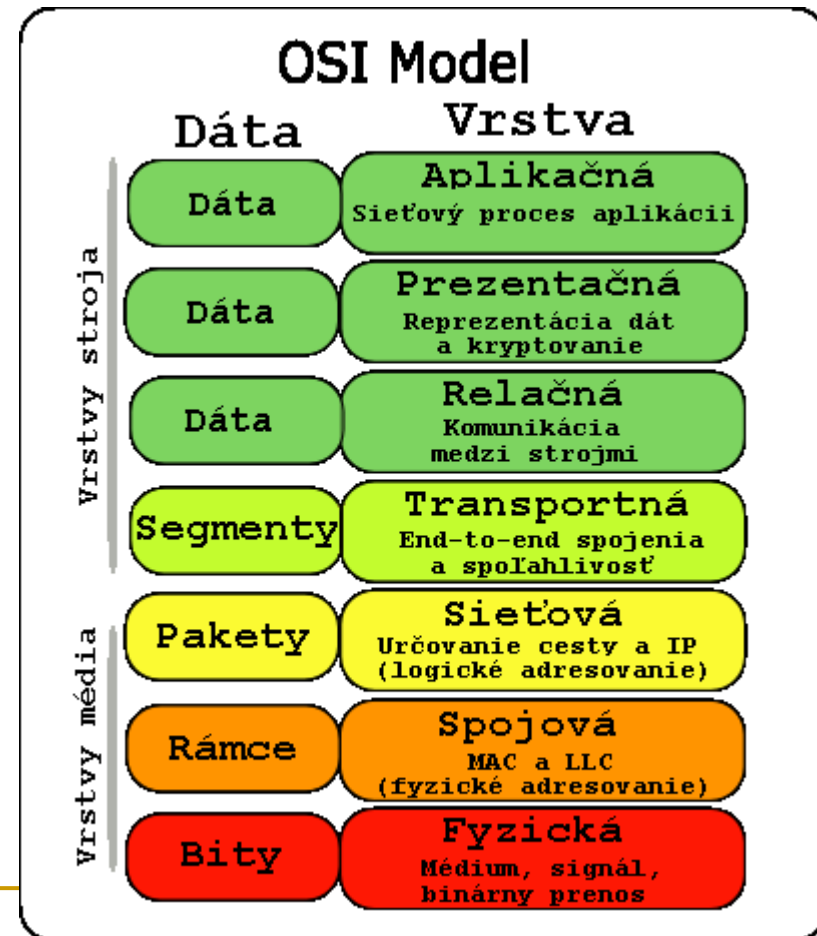
Pavel Petrovič, I24a, Peter Borovanský, I-18
petrovic@fmph.uniba.sk, borovan@ii.fmph.uniba.sk
<http://dai.fmph.uniba.sk/courses/java2>

Zámer

- dokončiť načaté/nenačaté témy o jazyku Java
- zorientovať sa v programovaní sieťových aplikácií, distribuované výpočty
- XML a Java
- Servlety: Glassfish/Tomcat
- JSP
- Databázy a Java
- Reflection Model
- Mobily – Symbian (midlet)
- Mobily - Android

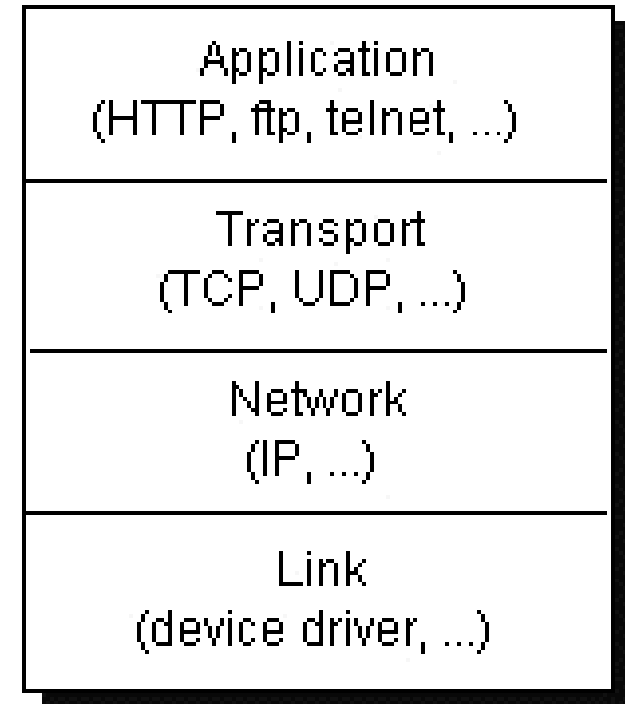
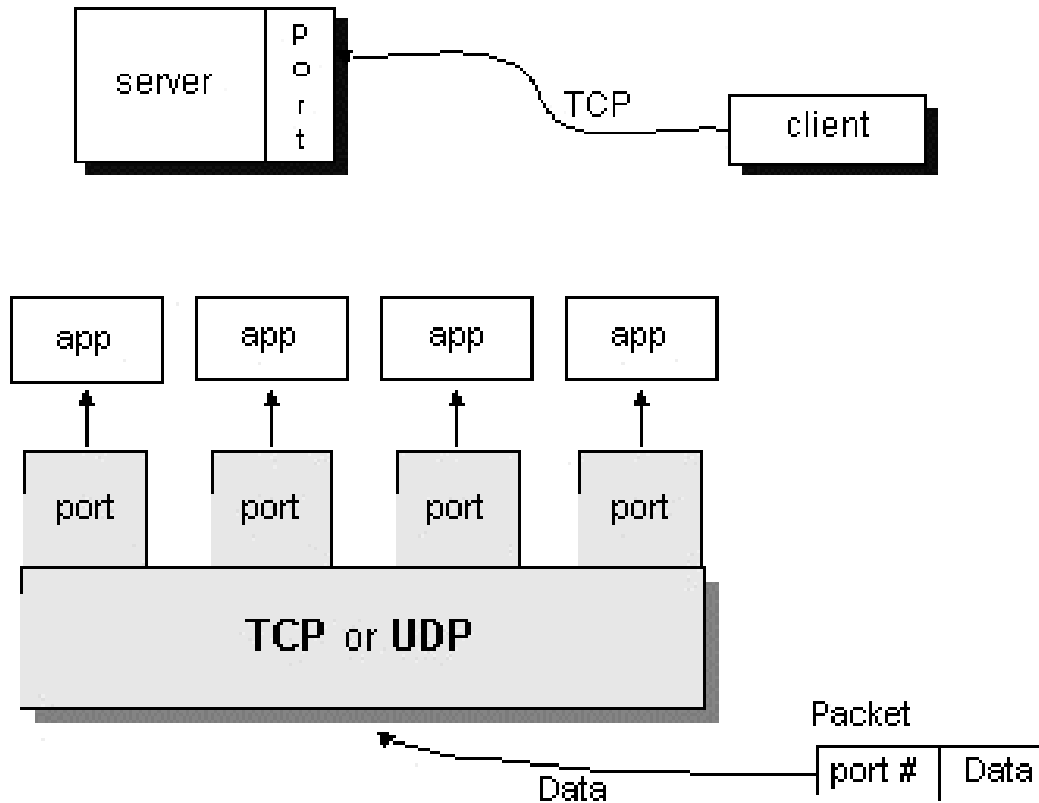
Internet

- IP adresa/menný priestor domén/DNS
- v4 vs. v6
- Model OSI (Open Systems Interconnection)
- Spojenie:
 - Jednorázové balíčky (UDP)
 - Trvalé spojenie (TCP)



Internet

- IP porty



Java Networking

- Aby sme sa nemuseli veľmi opakovať, vychádzame z výborného Java Tutorialu:

docs.oracle.com/javase/tutorial/networking/

- URL, Sockety, Datagramy, Klient/Server

S čím sme sa už stretli

- Applety (<APPLET>)

```
protected ImageIcon createImageIcon(String path, String description)
{
    java.net.URL imgURL = getClass().getResource(path);
    if (imgURL != null)
    {
        return new ImageIcon(imgURL, description);
    } else
    {
        System.err.println("Couldn't find file: " + path);
        return null;
    }
}
```

- URL

S čím sme sa už stretli

- Applety (<APPLET>)

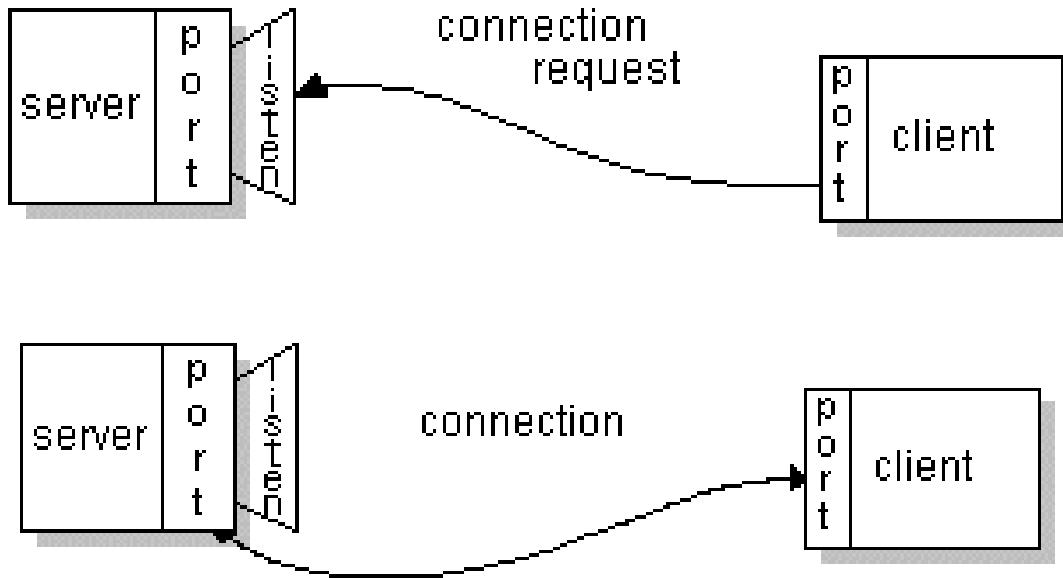
```
protected ImageIcon createImageIcon(String path, String description)
{
    java.net.URL imgURL = getClass().getResource(path);
    if (imgURL != null)
    {
        return new ImageIcon(imgURL, description);
    } else
    {
        System.err.println("Couldn't find file: " + path);
        return null;
    }
}
```

- URL

URL

1. Create a URL.
2. Retrieve the URLConnection object.
3. Set output capability on the URLConnection.
4. Open a connection to the resource.
5. Get an output stream from the connection.
6. Write to the output stream.
7. Close the output stream.

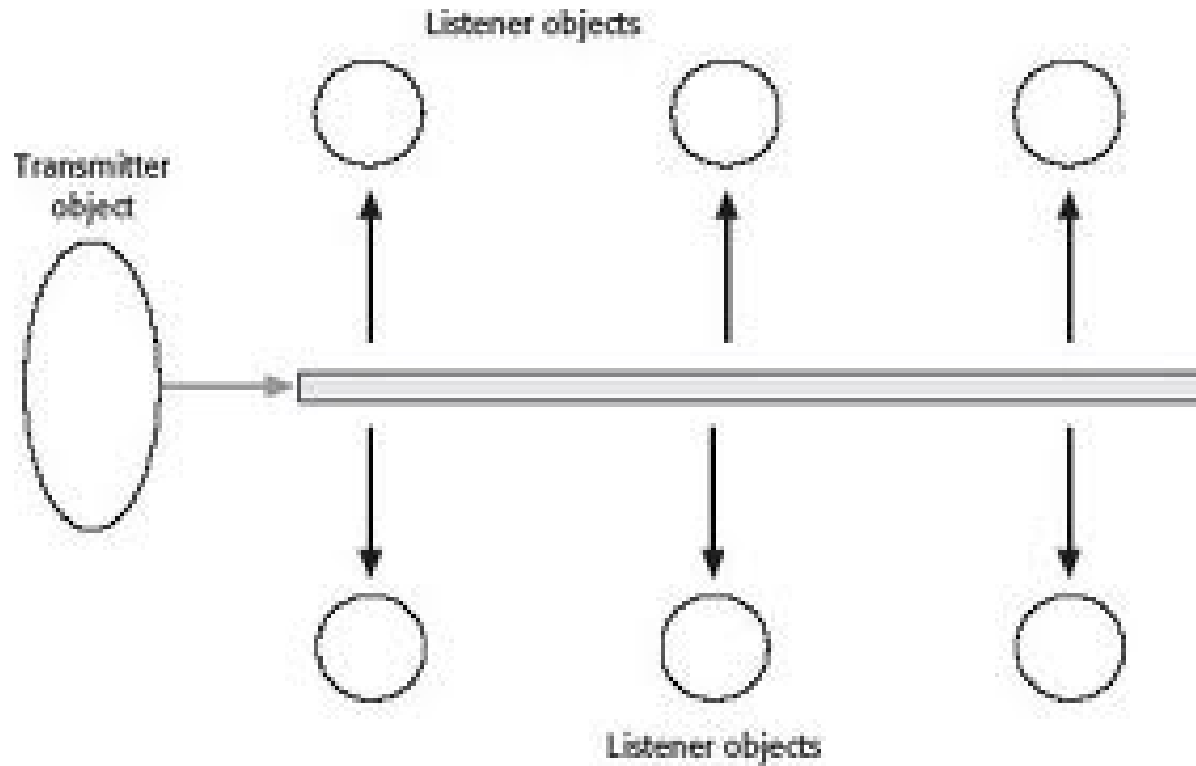
Sockety



`java.net.Socket`

Server „počúva“ (čaká na požiadavky na spojenie) na nejakom porte. Po prijatí požiadavky (`accept`) sa automaticky vytvorí nové spojenie (nový socket), ktorý slúži ako obojstranný komunikačný kanál.

Datagramy a MultiCast



Komunikačný protokol

Pravidlá komunikácie medzi jednotlivými uzlami (napr. aké požiadavky server akceptuje a ako na ne odpovedá)

Mal by byť zdokumentovaný v osobitnom súbore, aby sme nemuseli podrobnosti hľadať v zdrojovom kóde.

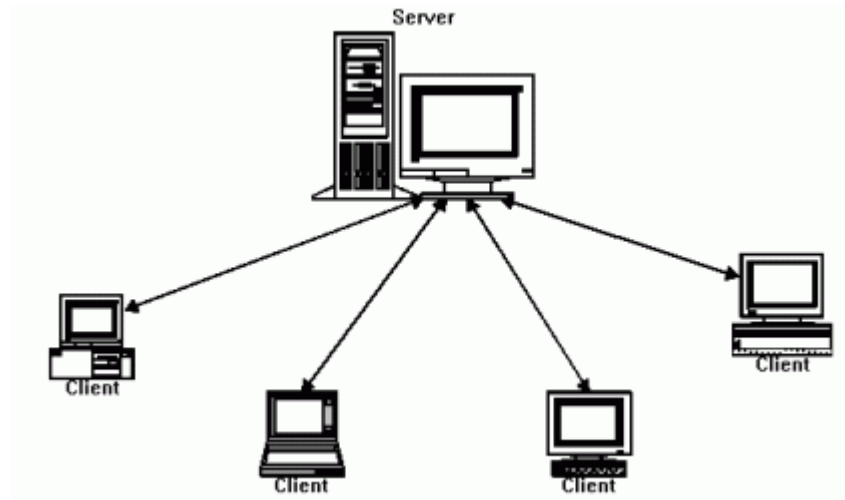
Príklad:

The communicating peers are exchanging packets in the following protocol:

length	type	description

1	byte	low byte of data length (X)
1	byte	high byte of data length (Y)
$DATA_LENGTH = X + Y*256$		
$DATA_LENGTH$	byte[]	data (string message)

Client/Server



<http://dai.fmph.uniba.sk/courses/java2/net/>