

Java Sockets

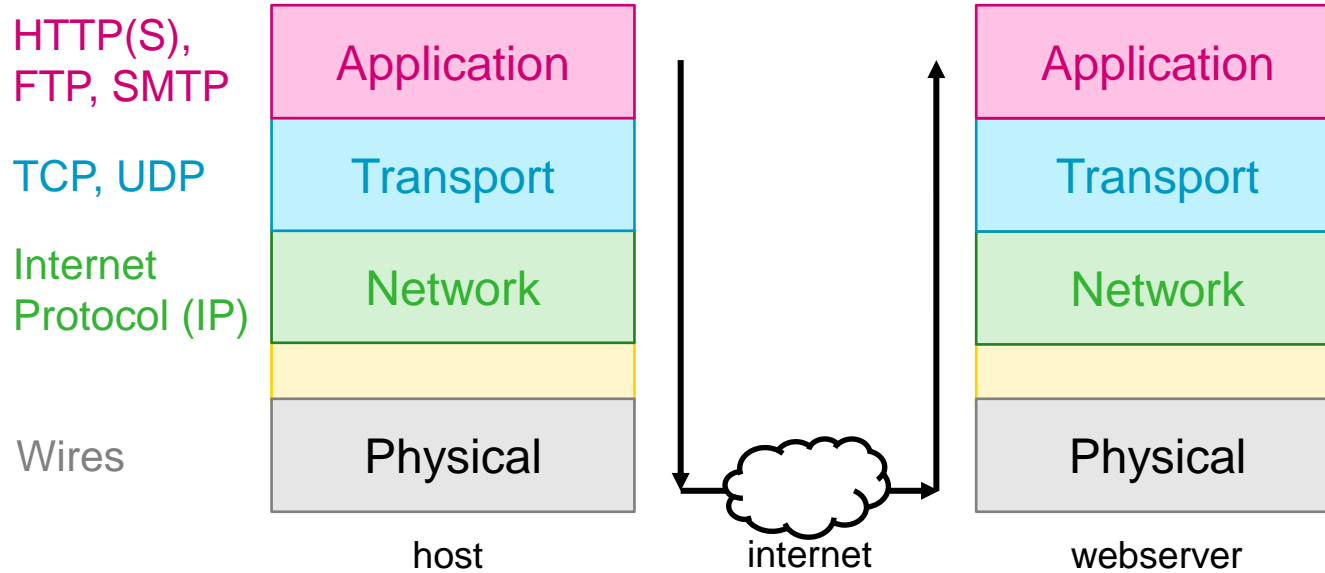
Topic of Software Systems (TCS module 2)

Lecturer: Ömer Şakar



NETWORKING

PROTOCOL LAYERS



HOST ADDRESS

- Part of the networking layer
- 130.89.3.231 or 192.168.1.3 (IPv4)
- 2001:0db8:85a3:0000:0000:8a2e:0370:7334 or 2001:0db8:85a3::8a2e:0370:7334 (IPv6)
- Domain names like utwente.nl or utoday.nl
- 127.0.0.1 or localhost

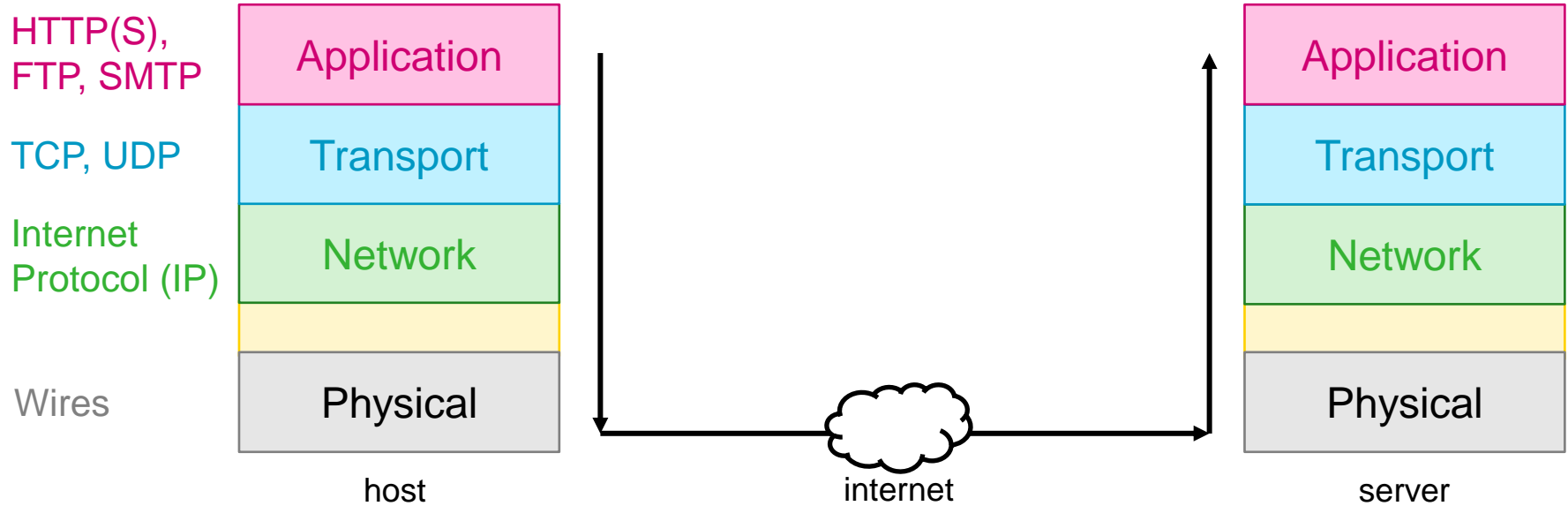
PORT NUMBER

- Part of the Transport Layer
- A number between 0 and 65535
- Ports 0 through 1023 are reserved
- Port number associated with an application

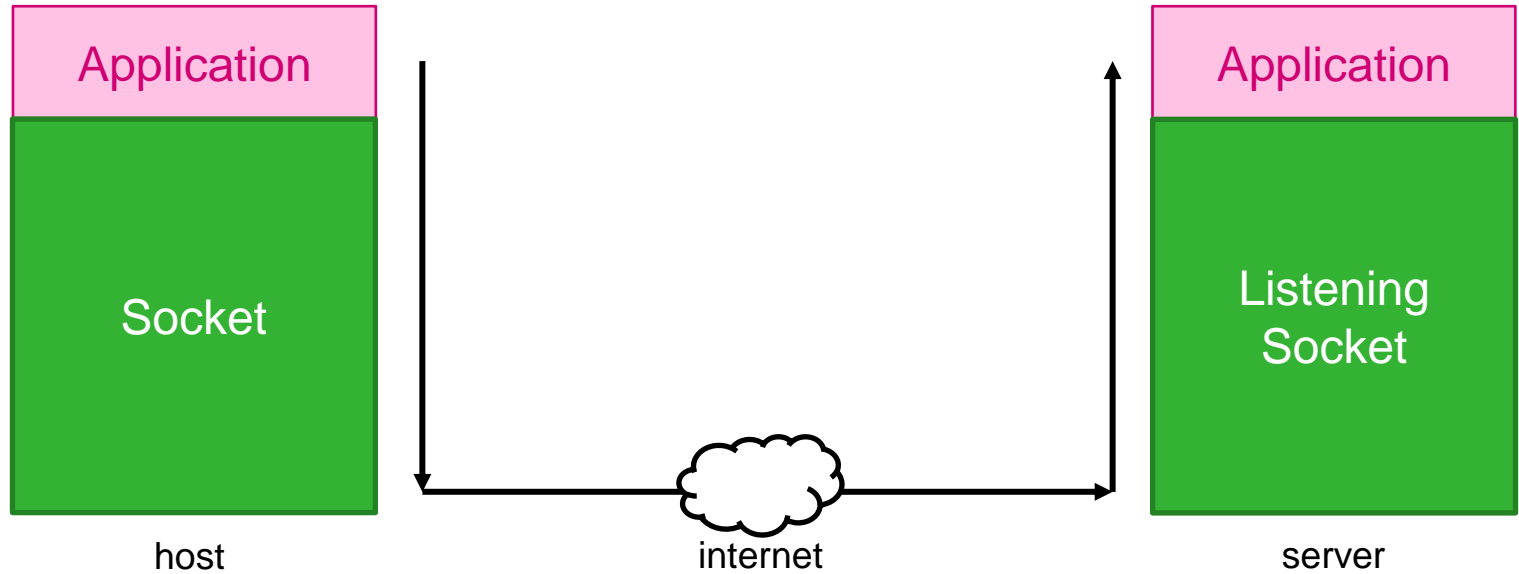
SOCKETS

- Socket is a common abstraction of the transport layer
 - Combines host address and port number
- Provides communication between program parts
 - Server waits on a port at an Internet address
 - Client tries to connect with port at an Internet address

PROTOCOL LAYERS



PROTOCOL LAYERS



CONTENTS

- Basic Networking
- Java Sockets

HOST ADDRESS IN JAVA

- Class `java.net.InetAddress` encapsulates an IP address
- String `getHostName()` gives host name (e.g., "utoday.nl")
- String `getHostAddress()` gives address (e.g., "130.95.72.134")
- Factory method `InetAddress getByName(String hostName)`
 - `hostName` can be "utwente.nl" or "130.95.72.134"
- Localhost `InetAddress getLocalHost()`

JAVA SOCKET

- `java.net.Socket` (TCP)
- `java.net.ServerSocket` (TCP)
- `java.net.DatagramSocket` (UDP)

SERVERSOCKET

- Listens on fixed port for incoming connections
 - Creates a connection on a certain port for each incoming request
- `ServerSocket(int port)`: constructor; if port is 0 chooses a free port
- `Socket accept()` returns a `Socket` if the attempt is successful
- `Socket close()` closes the server
- `InetAddress getInetAddress()` returns the local IP address
- `int getLocalPort()` returns the port the server is listening on.

SOCKET

- Bi-directional communication between sender and receiver
- `Socket(String remoteHost, int port)` starts a connection with remote host at port
- `InputStream getInputStream()` allows data from the other party to be received
- `OutputStream getOutputStream()` allows data to be sent to the other party

EXAMPLE

Client

```
s = new Socket  
("server", 9080);
```



```
s.getInputStream()
```

```
s.getOutputStream()
```

example
port numbers

example
port numbers

2037 -----> 9080

2037 <----- 1583

2037 -----> 1583

Server ("server")

```
ServerSocket ss =  
new ServerSocket(9080);  
...  
Socket s = ss.accept();
```



```
s.getOutputStream()
```

```
s.getInputStream()
```