

ITSE322 Modern Programming Language: Advanced Java

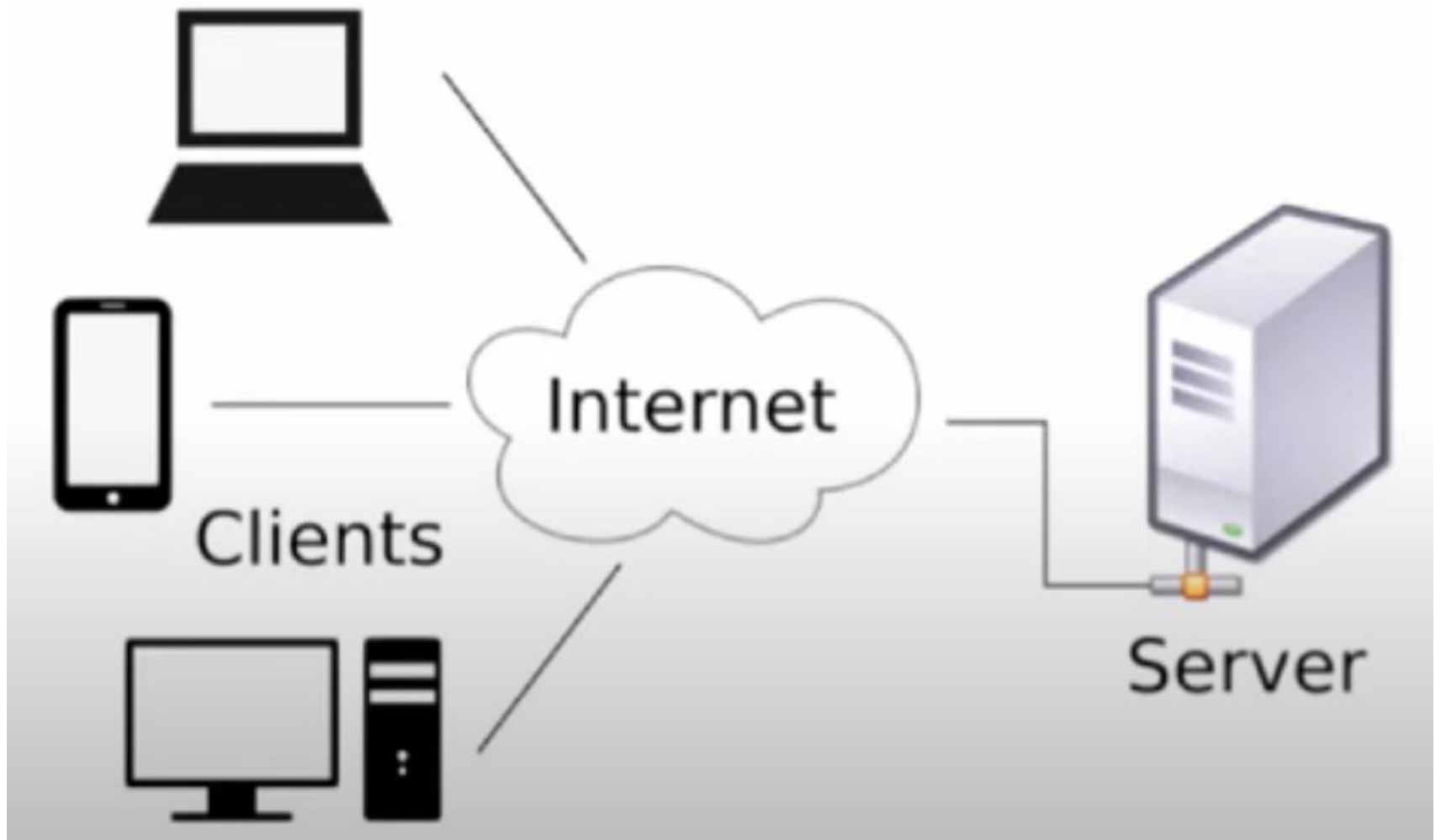
Java Network Programming

Lecture 7

Learning Objectives

- Understand the concept of Client-Server Applications
- Create Client-Server Applications
- Develop multithreaded server
- Send complex data structures to the server.

Client-Server Model



Two Communication Models



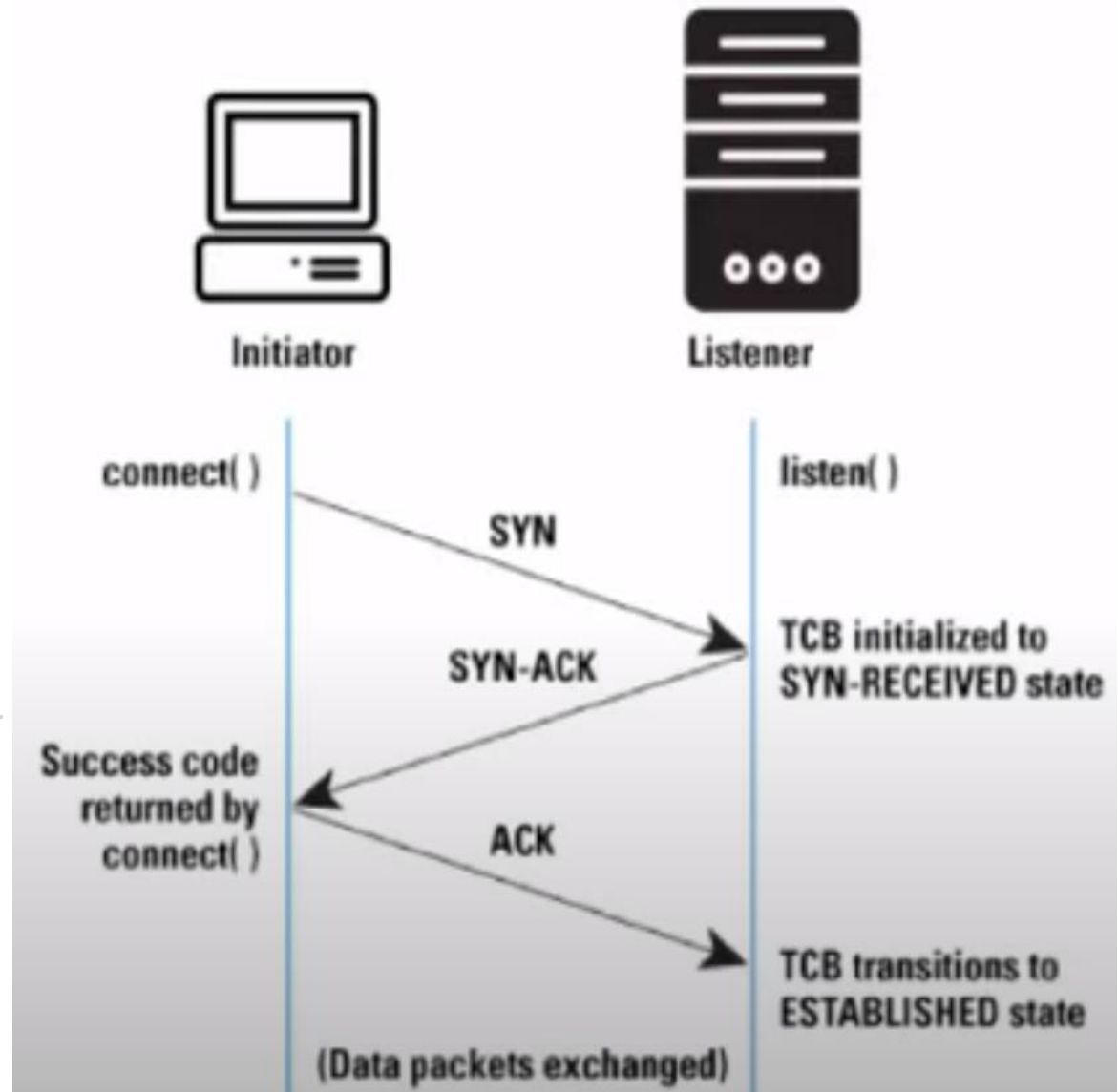
TCP Socket

- Connection-oriented
- Reliable
- Starts with handshake

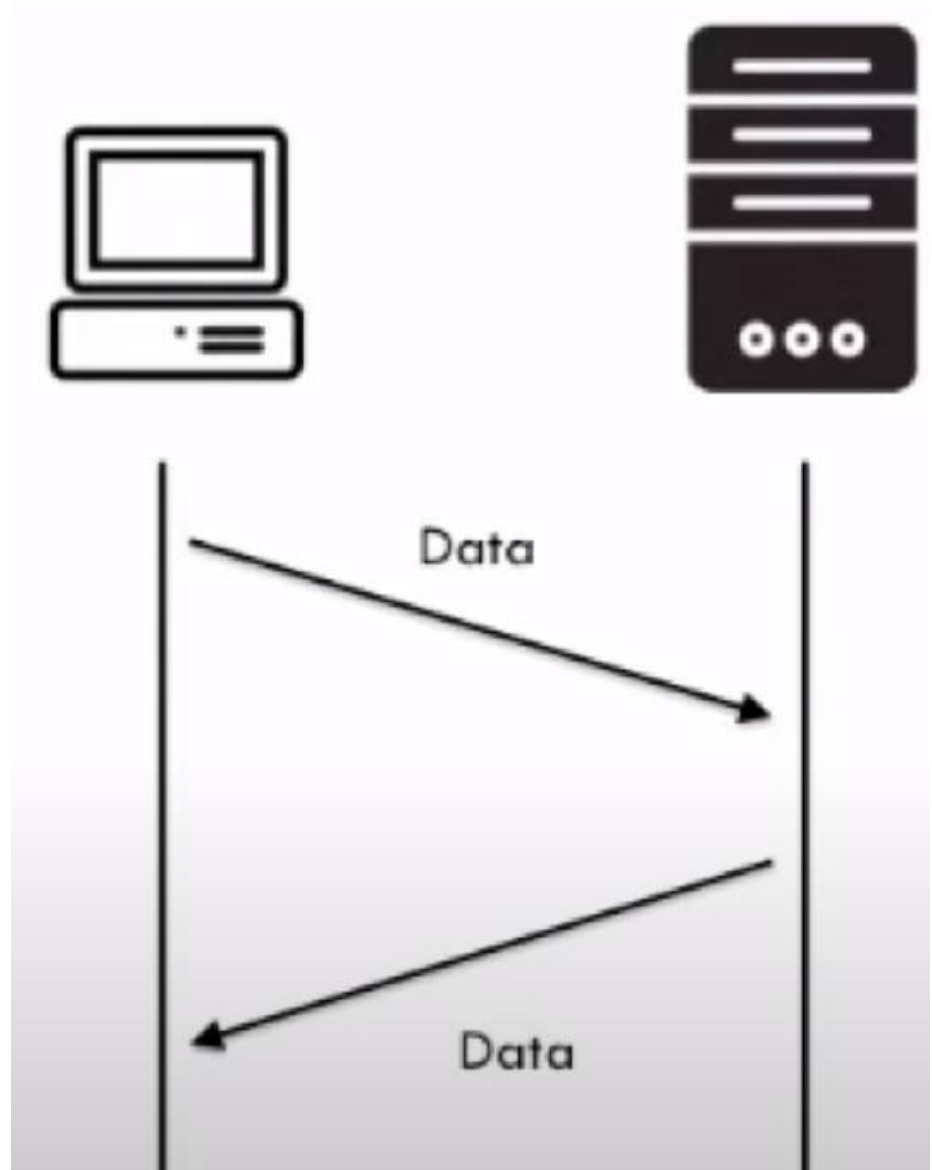
UDP Socket

- No connection
- No guarantees (not reliable)
- No handshake

TCP 3-way Handshaking

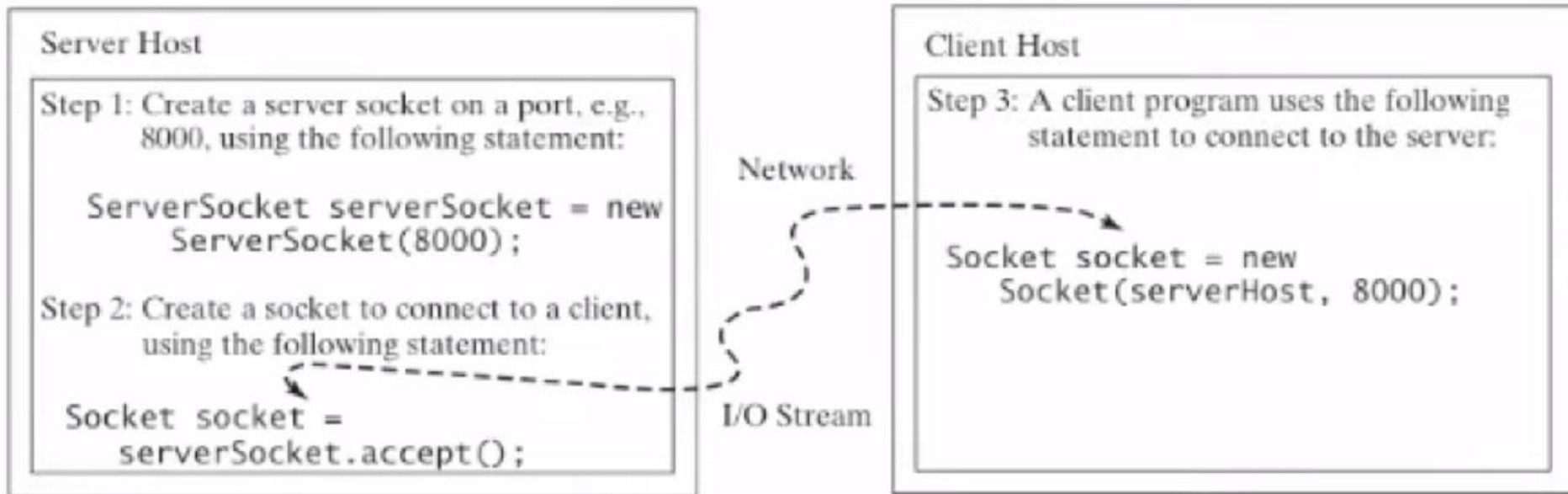


UDP



How to program in Java

Java TCP Client-Server model



- **ServerSocket:** create a TCP server
- **Socket:** create a TCP client

TCP Client/Server Example

Server

```
int port = 8000;
DataInputStream in;
DataOutputStream out;
ServerSocket server;
Socket socket;

server = new ServerSocket(port);
socket = server.accept(); ←
in = new DataInputStream
    (socket.getInputStream());
out = new DataOutputStream
    (socket.getOutputStream());
System.out.println(in.readDouble()); ←
out.writeDouble(aNumber); →
```

Connection
Request

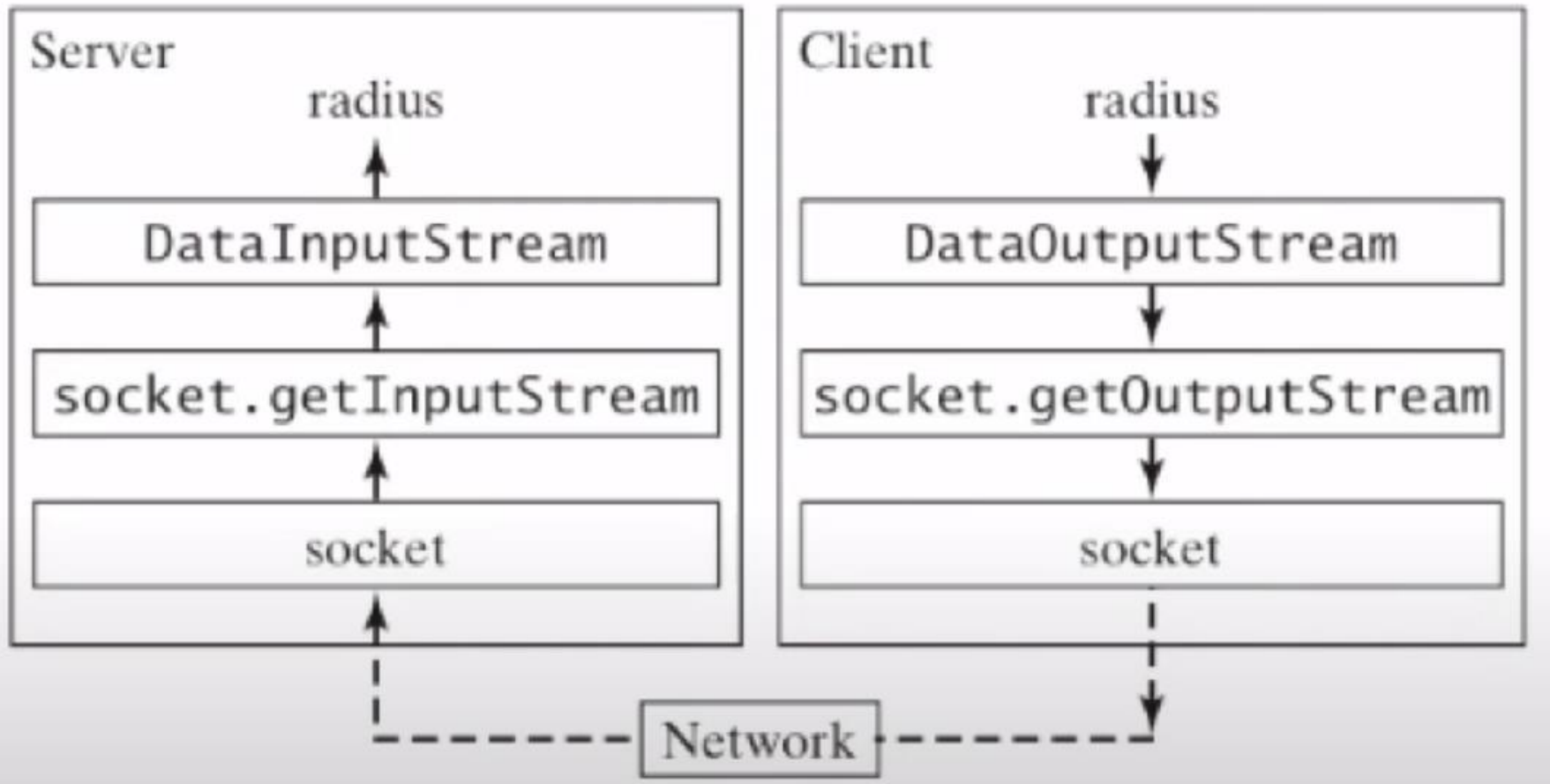
I/O
Streams

Client

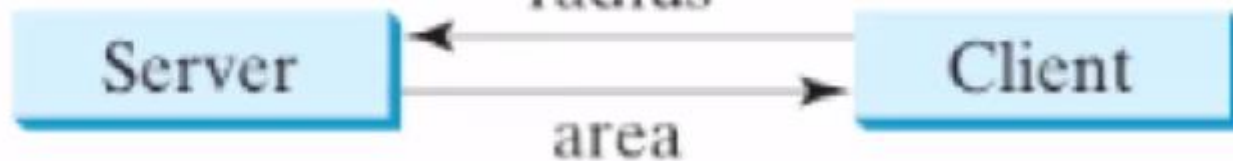
```
int port = 8000;
String host = "localhost"
DataInputStream in;
DataOutputStream out;
Socket socket;

socket = new Socket(host, port);
in = new DataInputStream
    (socket.getInputStream());
out = new DataOutputStream
    (socket.getOutputStream());
out.writeDouble(aNumber);
System.out.println(in.readDouble());
```

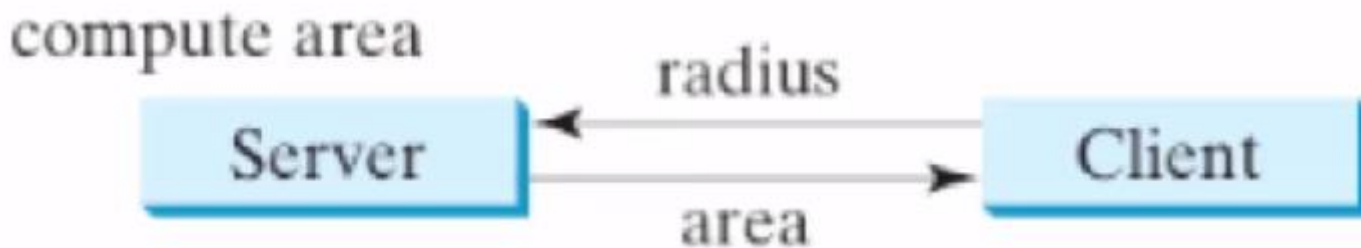
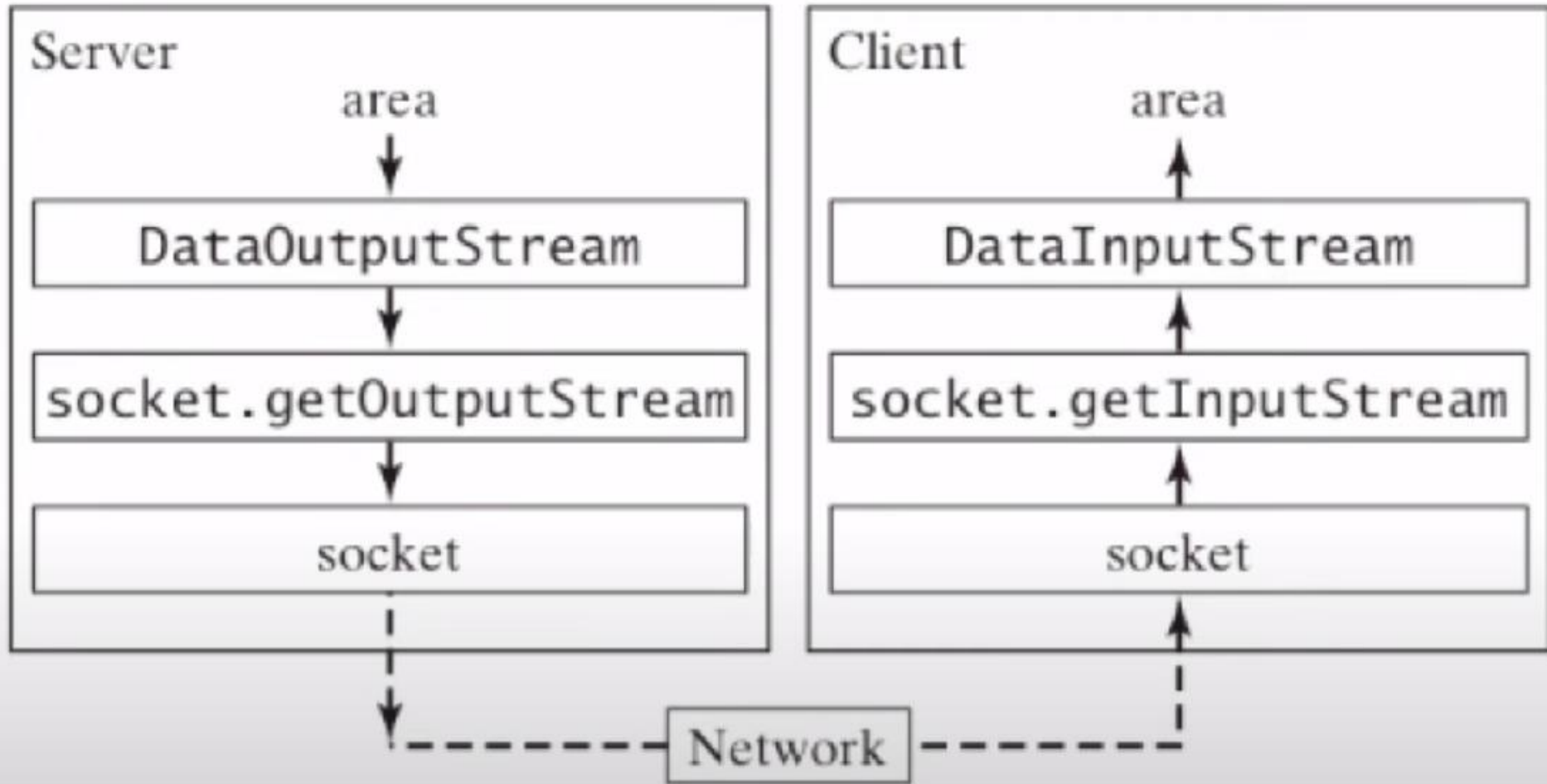
InputStream/OutputStream



compute area



InputStream/OutputStream



UDP

- **DatagramSocket:**

- create a UDP server/client
- DatagramSockets send and receive DatagramPackets.

- **DatagramPacket:**

- create a datagram packet
- objects store packets of data that are to be sent or that are received by an application.