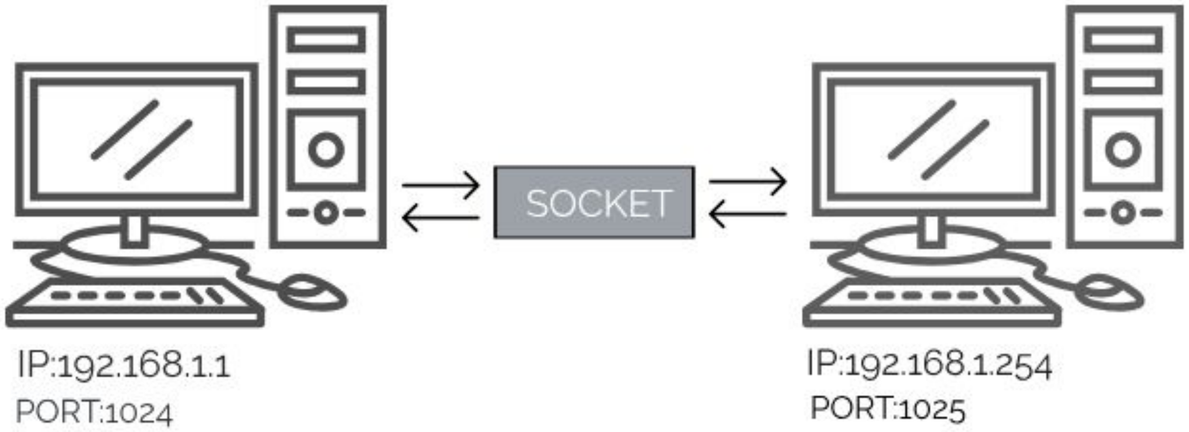
CSE 2100
LAB-5 : Into to Java Networking

Download the materials:
shorturl.at/cov16

What is Networking?

- Networking is a concept that allow processes to communicate with each other across a computer network
- Java socket programming provides facility to share data between different computing devices



Sockets

- What are sockets?
  - End-point of interprocess communication
  - An interface through which process can send or receive information
- What exactly creates a socket?
  - <IP address, Port #> tuple
- **IP address** is a unique address that identifies a device on a network
- **Port number** is used to uniquely identify different applications

Client-Server Model

- Most interprocess communication uses client-server model
- Client & server are two processes that wants to communicate with each other
- The client process connects to the server process, to communicate with the server
- Once the connection is established, they can start send information

Client-Server Model

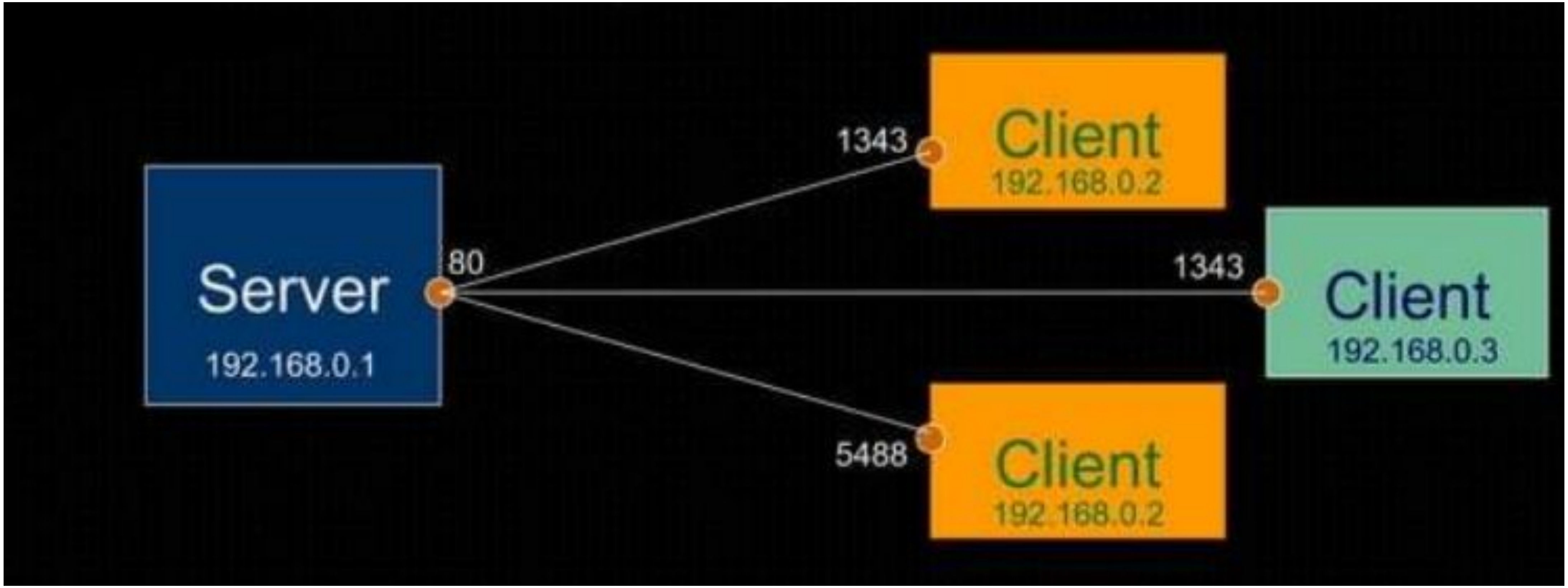


Fig. Example of a client-server model

A Generic Client-Server Application

Algorithm for TCP server

1. Find the IP address and port number of server
2. Create a TCP socket
3. Connect the socket to server
4. Bind the server socket to server IP and Port number
5. Accept a new connection from client

A Generic Client-Server Application

Algorithm for TCP client

1. Find the IP address and port number of server
2. Create a TCP socket
3. Connect the socket to server
4. Send/receive data with server using socket
5. Close the connection

A Generic Client-Server Application

- All the classes related to sockets are in java.net package
- All the input/output stream classes are java.io package

Creating a server

```
ServerSocket server = new ServerSocket(port_number);
Socket s = server.accept();
```

Creating a client

```
Socket s = new Socket(machine_name,  port_number);
```

A Generic Client-Server Application

How to create a input/output stream?

In the client and server side a DataInputStream and DataOutputStream class to create input/output stream to send and receive information

```
DataInputStream din = new DataInputStream(s.getInputStream());
DataOutputStream dout = new DataOutputStream(s.getOutputStream());
```

Closing a socket

You should always close input and output stream befor closing socket

```
din.close();
dout.close();
s.close();
server.close();
```

Task

- Create a real-time chatting application using Java Socket

END