Socket Programming

CS F303

BITS Pilani
Dubai Campus

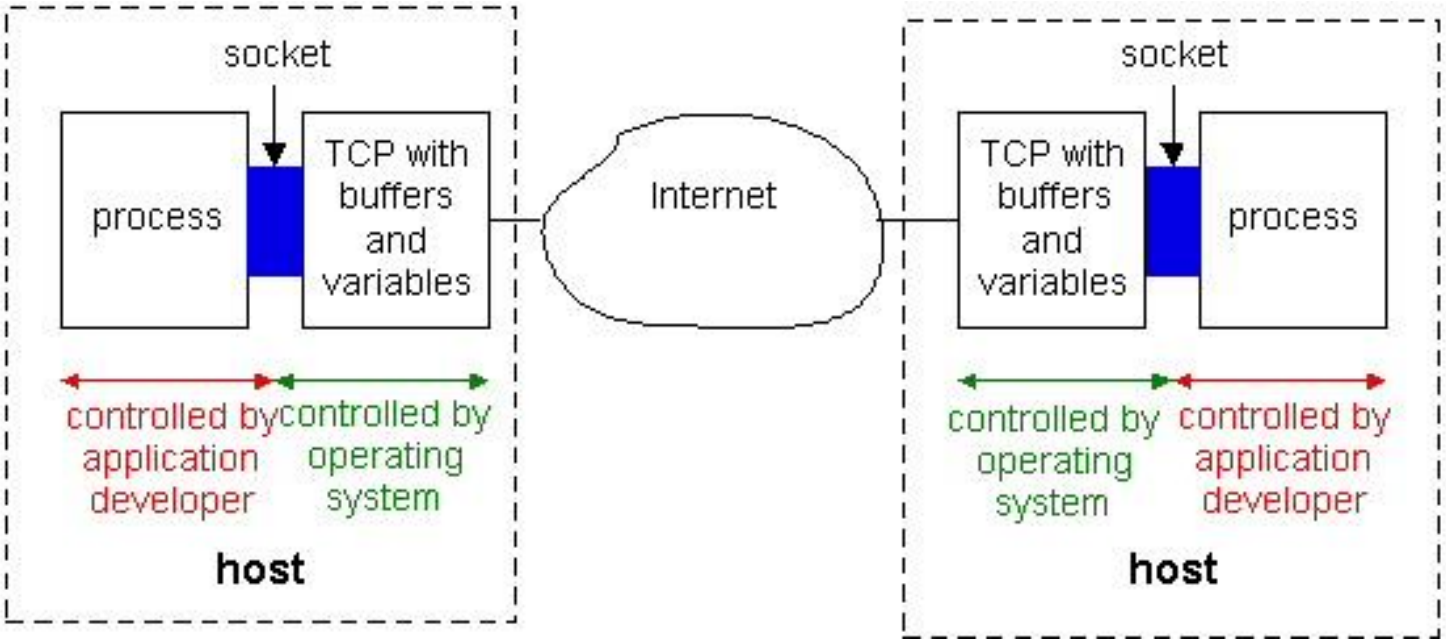
Dr. Pranav M. Pawar

Socket

- Socket
  - The combination of an IP address and a port number.
  - The name of the Berkeley-derived *application programming interfaces* (APIs) for applications using TCP/IP protocols.
  - Two types
    - Stream socket : reliable two-way connected communication streams
    - Datagram socket

- Socket pair
  - Specified the two end points that uniquely identifies each TCP connection in an internet.
  - 4-tuple: (client IP address, client port number, server IP address, server port number)

Socket Programming with TCP



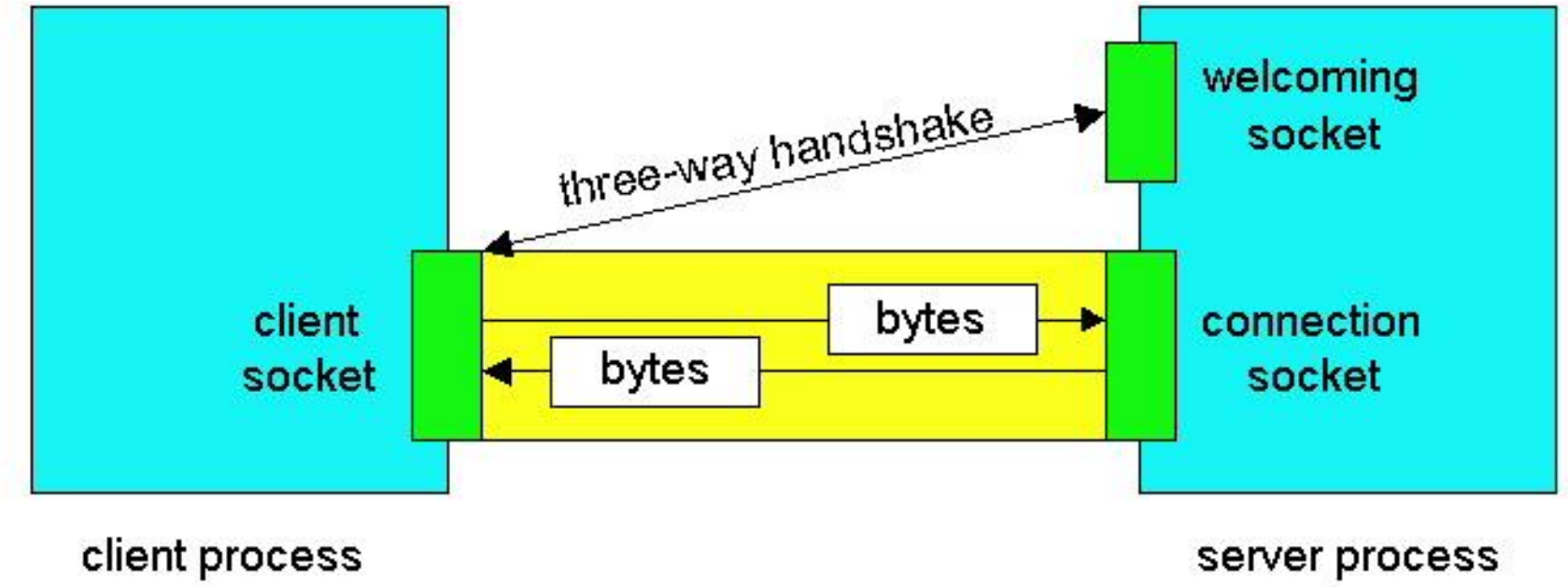
Socket for Client and Server

- Server
  - Welcoming socket
    - Welcomes some initial contact from a client.
  - Connection socket
    - Is created at initial contact of client.
    - New socket that is dedicated to the particular client.
- Client
  - Client socket
    - Initiate a TCP connection to the server by creating a socket object. (Three-way handshake)
    - Specify the address of the server process, namely, the IP address of the server and the port number of the process

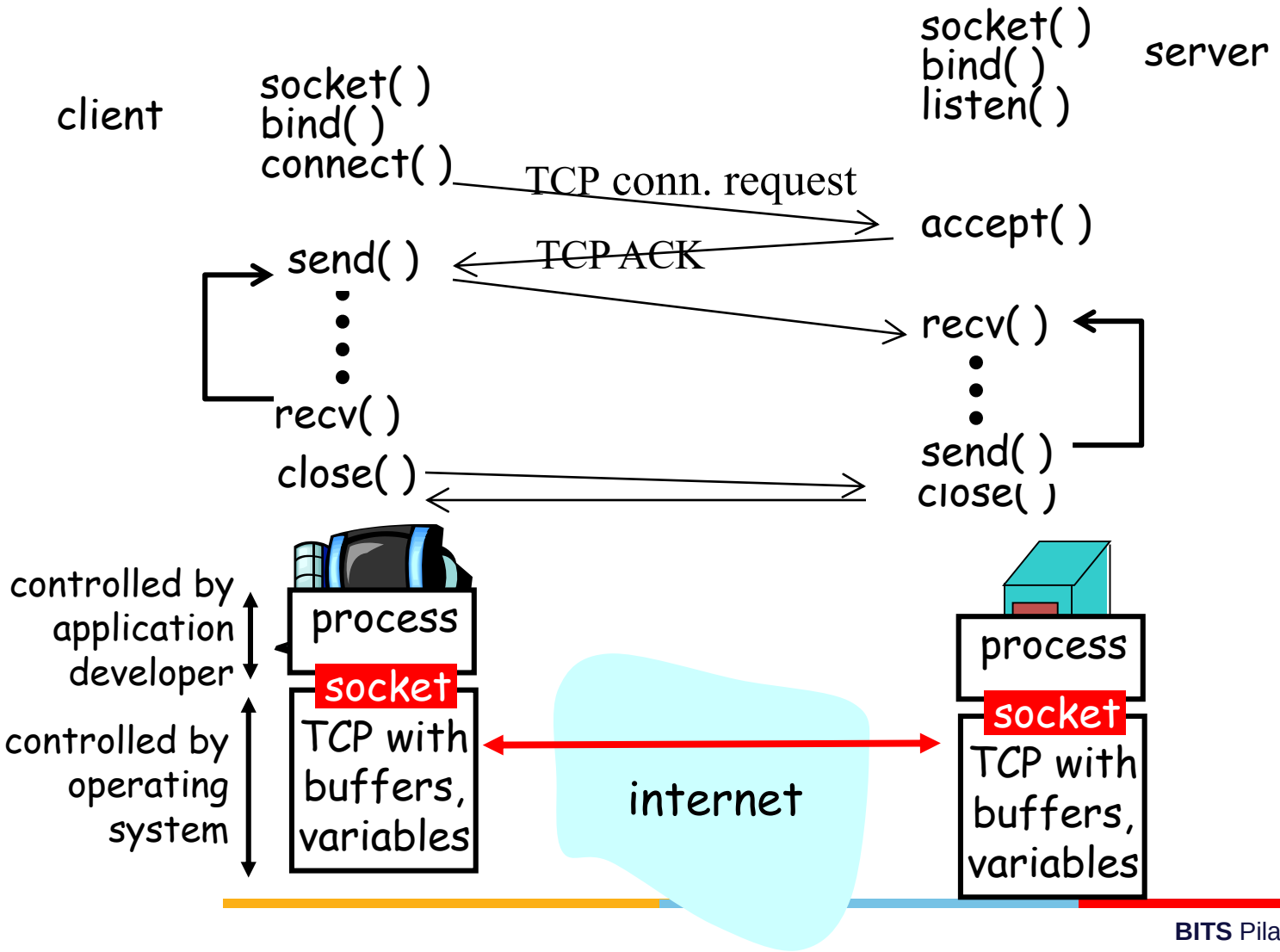
Socket Function Call

- socket (): Create a socket
- bind(): bind a socket to a local IP address and port #
- listen(): passively waiting for connections
- connect(): initiating connection to another socket
- accept(): accept a new connection
- Write(): write data to a socket
- Read(): read data from a socket
- sendto(): send a datagram to another UDP socket
- recvfrom(): read a datagram from a UDP socket
- close(): close a socket (tear down the connection)

Socket Communication



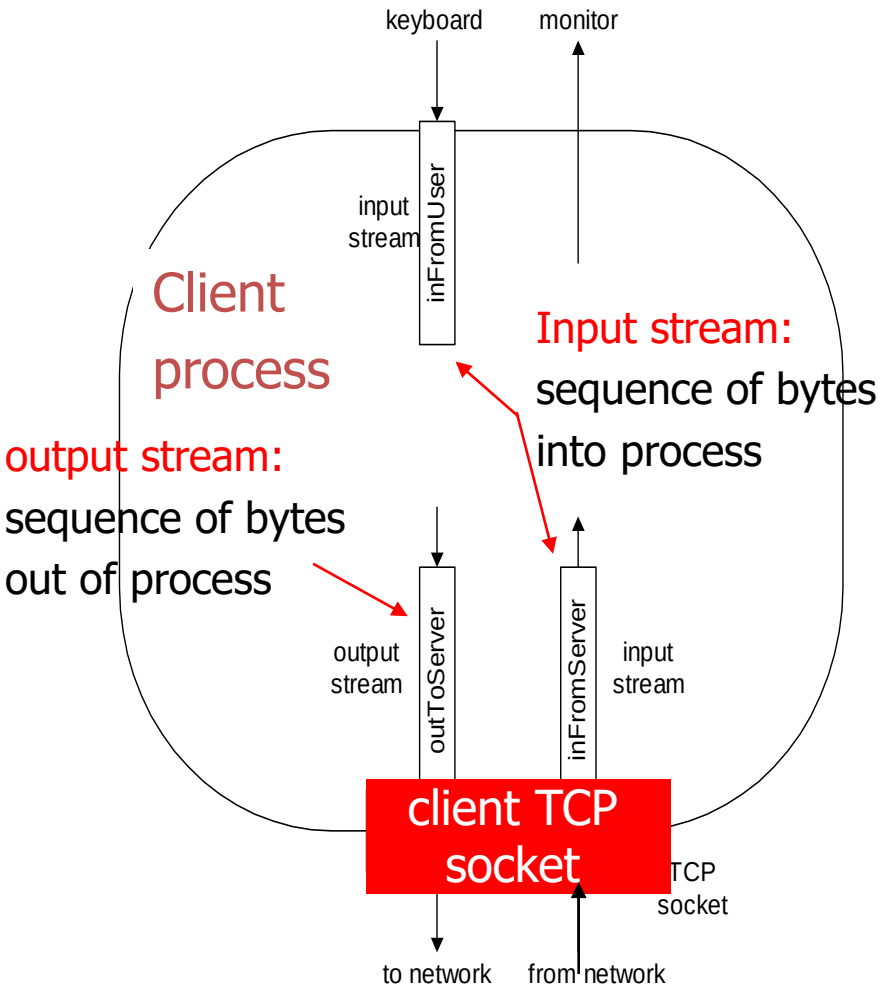
Socket-programming using TCP



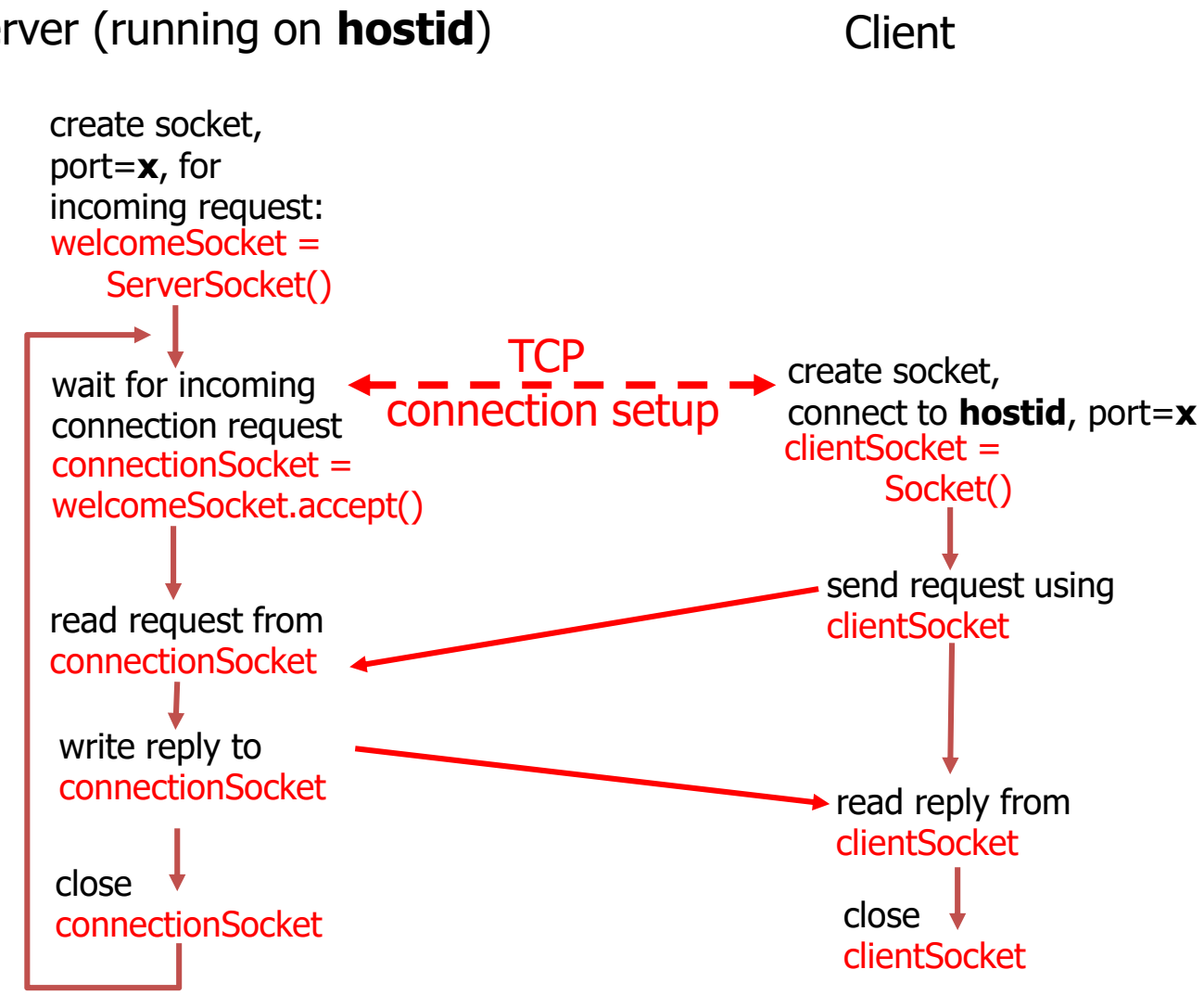
Socket programming with TCP

Example client-server app:

- client reads line from standard input (**inFromUser** stream) , sends to server via socket (**outToServer** stream)
- server reads line from socket
- server converts line to uppercase, sends back to client
- client reads, prints modified line from socket (**inFromServer** stream)



Client/server socket interaction: TCP



JAVA TCP Sockets

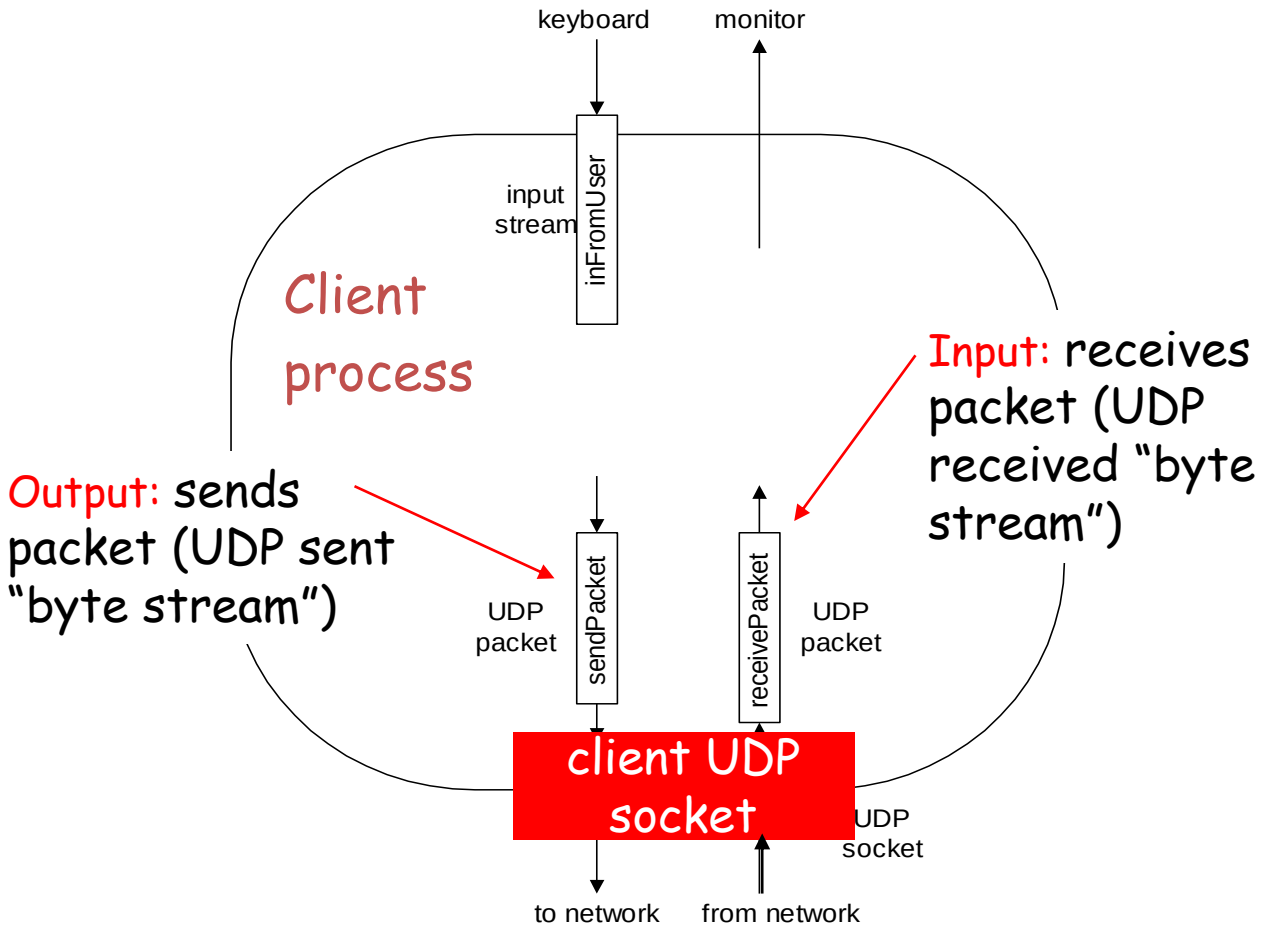
- In Package java.net
  - java.net.Socket
    - Implements client sockets (also called just "sockets").
    - An endpoint for communication between two machines.
    - Constructor and Methods
      - Socket(String host, int port): Creates a stream socket and connects it to the specified port number on the named host.
      - InputStream getInputStream()
      - OutputStream getOutputStream()
      - close()
  - java.net.ServerSocket
    - Implements server sockets.
    - Waits for requests to come in over the network.
    - Performs some operation based on the request.
    - Constructor and Methods
      - ServerSocket(int port)
      - Socket Accept(): Listens for a connection to be made to this socket and accepts it. This method blocks until a connection is made.

Socket Programming with UDP

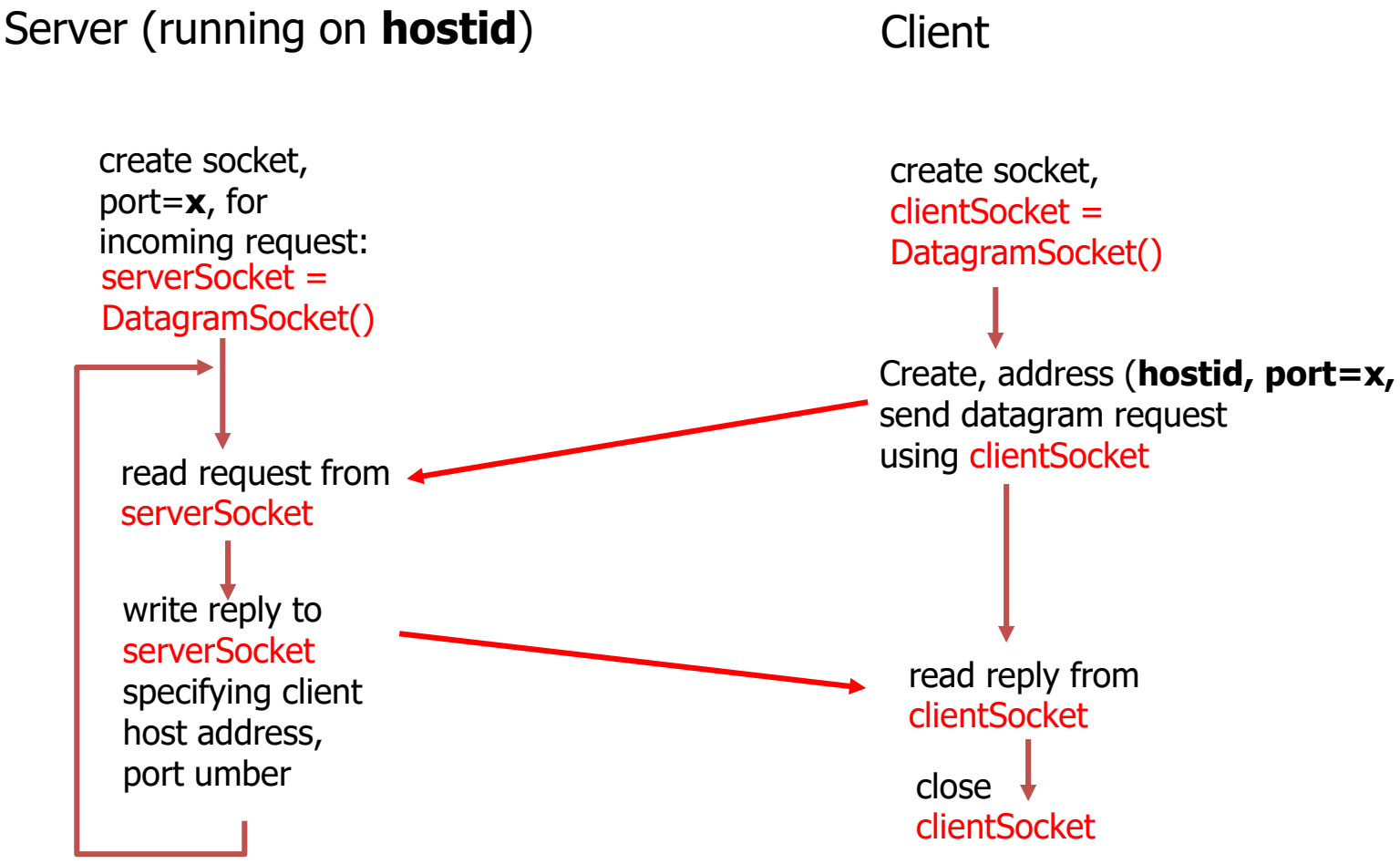
- UDP
  - Connectionless and unreliable service.
  - There isn't an initial handshaking phase.
  - Doesn't have a pipe.
  - transmitted data may be received out of order, or lost

- Socket Programming with UDP
  - No need for a welcoming socket.
  - No streams are attached to the sockets.
  - the sending hosts creates "packets" by attaching the IP destination address and port number to each batch of bytes.
  - The receiving process must unravel to received packet to obtain the packet's information bytes.

Example: Java client (UDP)



Client/server socket interaction: UDP



JAVA UDP Sockets

- In Package java.net
  - java.net.DatagramSocket
    - A socket for sending and receiving datagram packets.
    - Constructor and Methods
      - DatagramSocket(int port): Constructs a datagram socket and binds it to the specified port on the local host machine.
      - void receive(DatagramPacket p)
      - void send(DatagramPacket p)
      - void close()

Sample TCP Socket Program

- GreetingClient.java
- GreetingServer.java

Practice Statement

- Create a simple client server application where server should return current data and time to the client. (Use TCP socket)

Sources

- https://www.codejava.net/java-se/networking/java-socket-server-examples-tcp-ip
- https://www.javatpoint.com/socket-programming
- https://www.tutorialspoint.com/java/java_networking.htm#:~:text=Sockets%20provide%20the%20communication%20mechanism,its%20end%20of%20the%20communication.
- https://www.it.uu.se/edu/course/homepage/distrinfo/ht11/schedule/Java_Socket_Programming.ppt

innovate achieve lead

BITS Pilani
Dubai Campus

Thank You!