NIELIT Gorakhpur

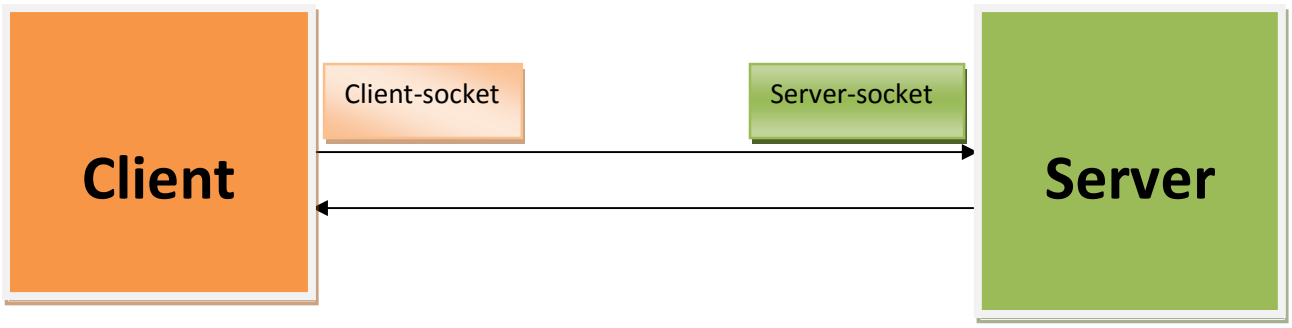
Course Name: A Level (2nd Sem) **Subject: JAVA**
Topic: Socket Programming **Date: 18-05-20**
(Networking in Java)

Java Networking/Socket programming:

In Java, networking is performed using Transmission Control Protocol/Internet Protocol (TCP/IP) or the User Datagram Protocol (UDP).
In Java, TCP connections are established using sockets and server sockets. These connections allow us to send and receive data to a Web Site. On the other hand, broadcast messages known as datagrams use the UDP.

Generally, network communication comes in two flavors: Transport Control Protocol (TCP) and User Datagram Protocol (UDP). TCP and UDP are used for different purposes and both have unique constraints:

- TCP is relatively simple and reliable protocol that enables a client to make a connection to a server and the two systems to communicate. In TCP, each entity knows that its communication payloads have been received.
- UDP is a connectionless protocol and is good for scenarios where we do not necessarily need every packet to arrive at its destination, such as media streaming and online-gaming etc.



Socket:

A socket forms the interface between the protocol and client for communication. A Java socket forms the base for data transmission between two computers using TCP/IP. The socket specifies the site address and the connection port. When packets reach a client or server, they are routed to the destination port specified in packet header.

The java.net Package:

The package **java.net** contains classes and interfaces that provide a powerful infrastructure for networking in Java. These include:

- The URL class for basic access to Uniform Resource Locators (URLs).
- The URLConnection class, which supports more complex operations on URLs.
- The Socket class for connecting to particular ports on specific Internet hosts and reading and writing data using streams.
- The ServerSocket class for implementing servers that accept connections from clients.
- The DatagramSocket, MulticastSocket, and DatagramPacket classes for implementing low-level networking.
- The InetAddress class, which represents Internet addresses.

Socket Class:

The java.net.Socket class is used to create a socket so that both the client and the server can communicate with each other easily. A socket is an endpoint for communication between two computers. The Socket class inherits the Object class and implements the Closeable interface.

Some commonly used public methods of the Socket class are:

1. synchronized void close()
2. Socket (String host, int port) throws UnknownHostException, IOException
3. Socket()
4. OutputStream getOutputStream()
5. InputStream getInputStream()

Server Socket Class:

The ServerSocket class represents a socket that listens for connection requests from clients on a specified port. When a connection is requested, a Socket object is created to handle the communication.

Some Important methods of the ServerSocket class are:

1. public Socket accept()
2. public synchronized void close()

Programs:

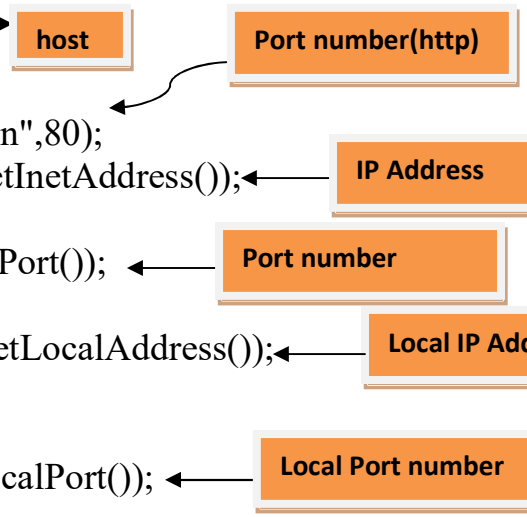
1)Basic Example to know IP Address,port number,local ip address,local port number etc(system should be connected to Internet)

```
import java.io.IOException;
import java.net.*;
public class Socket1 {
        public static void main(String args[]) throws UnknownHostException
        {
                try
                {
Socket mySocket = new Socket("student.nielit.gov.in",80);
System.out.println("Connection to: " + mySocket.getInetAddress());

System.out.println("Port Number: " + mySocket.getPort());

System.out.println("Local Address: " + mySocket.getLocalAddress());


System.out.println("Local Port: " + mySocket.getLocalPort());
                }
                catch (UnknownHostException e)
                {
                        System.out.println("Site not found!");
                }
                catch (SocketException e)
                {
                        System.out.println("Socket error");
                }
                catch ( IOException e)
                {
                        System.out.println("An I/O Exception Occurred!");
                }
        }
}
```



Exercise:

1. What is socket programming?
2. What are sockets and socket class? Why are they important?