Network Commands

1. **ping <IP address>**

   Sends test packets to a specific IP address to check if the target device is reachable and how long the packets take to travel.

   Ex - ping 8.8.8.8

2. **pathping <IP address>**

   Combines `ping` and `tracert` (trace route). It shows the path packets take to reach the target and measures packet loss at each hop (intermediate router).

   Ex - ping 8.8.8.8

3. **ipconfig**

   Displays network configuration info for your system.

   Ex - ipconfig

Network Commands

1. **getmac**

   Displays the **MAC address** (physical address) of your computer's network adapters.

   Ex - gatemac

2. **pathping <IP address>**

   Performs a **DNS query** to find the IP address(es) associated with a domain name.

   Ex - nslookup google.com

3. **ipconfig**

   Shows **network statistics** and active connections on your system.

   Ex - netstat

Socket Programming

Problem Definition

Write a Java program using socket programming where client and server both send and receive messages in realtime.

Algorithm

Server

```
Connect to server at localhost:6666

Set up:
    - input from server
(DataInputStream)
    - output to server
(DataOutputStream)
    - input from client user
(BufferedReader)

Initialize str = "", str2 = ""

WHILE str ≠ "stop" DO
    prompt "Client: "
    str ← read input from user
    send str to server
    flush output

    str2 ← read message from server
    print "Server says: " + str2

END WHILE

Close all streams and socket
```

Client

```
Start server on port 6666
Wait for client connection

Set up:
    - Input from client
    - Output to client
    - Input from server user
(console)

Initialize str = "", str2 = ""

WHILE str ≠ "stop"
    Read str from client
    Display str to server user

    Prompt for str2 (server input)
    Read str2 from console
    Send str2 to client

END WHILE

Close all streams and sockets
```

Implementation/Code

Server

```
import java.net.*;
import java.io.*;

class MyServer {
    public static void main(String[]
args) throws Exception {
        ServerSocket ss = new
ServerSocket(6666);
        Socket s = ss.accept();

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

        BufferedReader br = new
BufferedReader(new
InputStreamReader(System.in));

        String str = "", str2 = "";

        while (!str.equals("stop")) {
            str = din.readUTF();

System.out.println("Client says: " +
str);

            System.out.print("Server:
");

            str2 = br.readLine();
            dout.writeUTF(str2);
            dout.flush();
        }

        din.close();
        dout.close();
        s.close();
        ss.close();
    }
}
```

Client

```
import java.io.*;
import java.net.*;

public class MyClient {
    public static void main(String[]
args) {
        try {
            Socket s = new
Socket("localhost", 6666);

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

            BufferedReader br = new
BufferedReader(new
InputStreamReader(System.in));

            String str = "", str2 =
"";

            while
(!str.equals("stop")) {

System.out.print("Client: ");
                str = br.readLine();
                dout.writeUTF(str);
                dout.flush();

                str2 = din.readUTF();

System.out.println("Server says: " +
str2);
            }

            dout.close();
            din.close();
            s.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```