

# Basics of Networking

Networking is the process of connecting two or more computing devices together so, that data and resources can be shared!

## Terminologies

1. LAN - MAN - WAN

2. WWW

3. IP Address

4. Port Number

5. URL

6. MAC Address

7. Socket

IP address, is a unique number assigned to a node of a n/w,  
for eg: 192.168.0.1.

• It is composed of octets that range from 0 to 255.

192.168.1.34  
└──┬──┘ └┐  
N/w ID Host ID.

• It is a logical address that can be changed.

Port Number, is used to uniquely identify different applications.

• It acts as a communication endpoint b/w applications.

• The Port number is associated with the IP-address for communication b/w 2-apps.

IP address.

200.23.56.8

Port No.

69

200.23.56.8

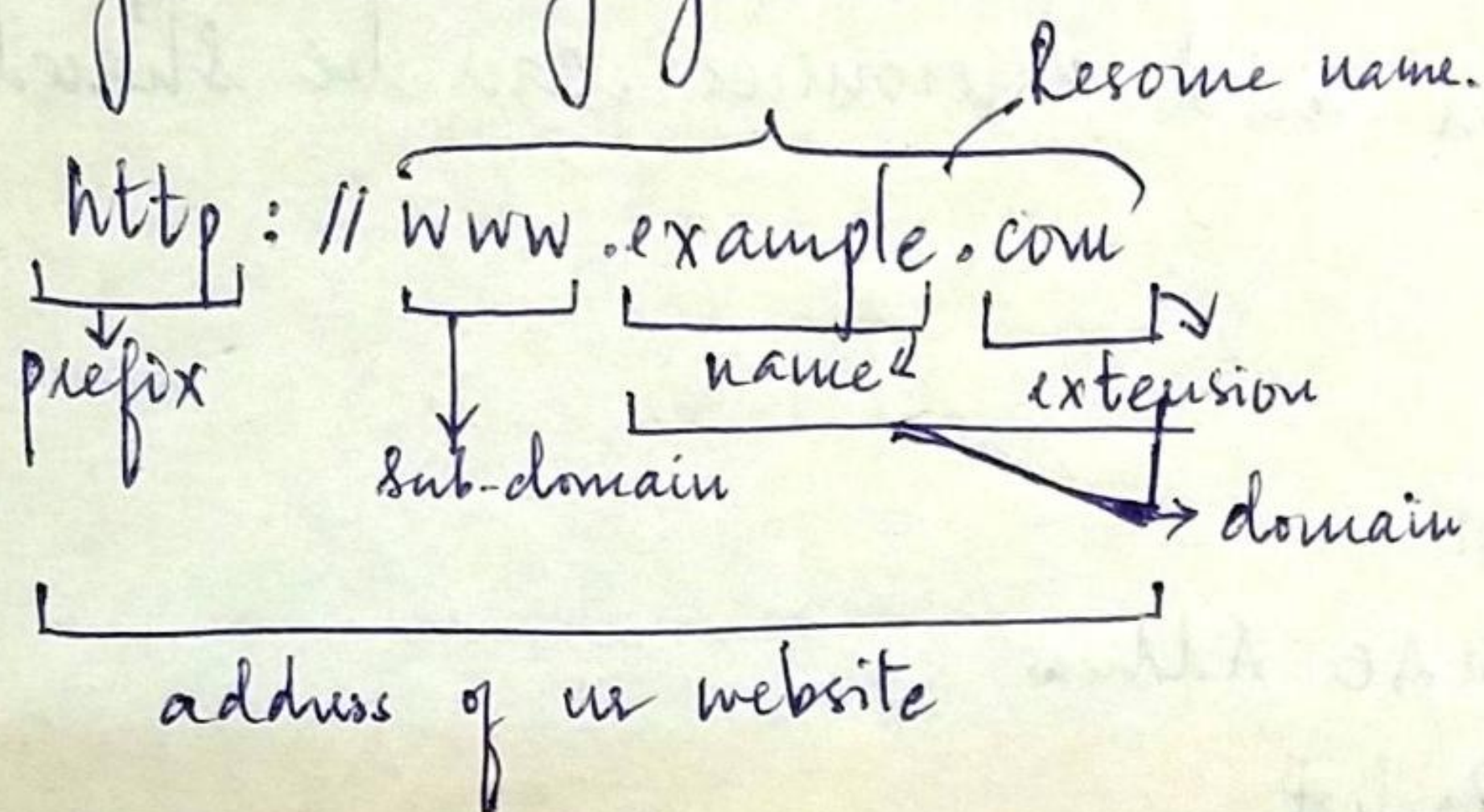
69

Socket Addr.



URL, is the abbreviations of Uniform Resource Locator and is defined as the global address of documents and other resources on the WWW.

eg: `www.google.com`



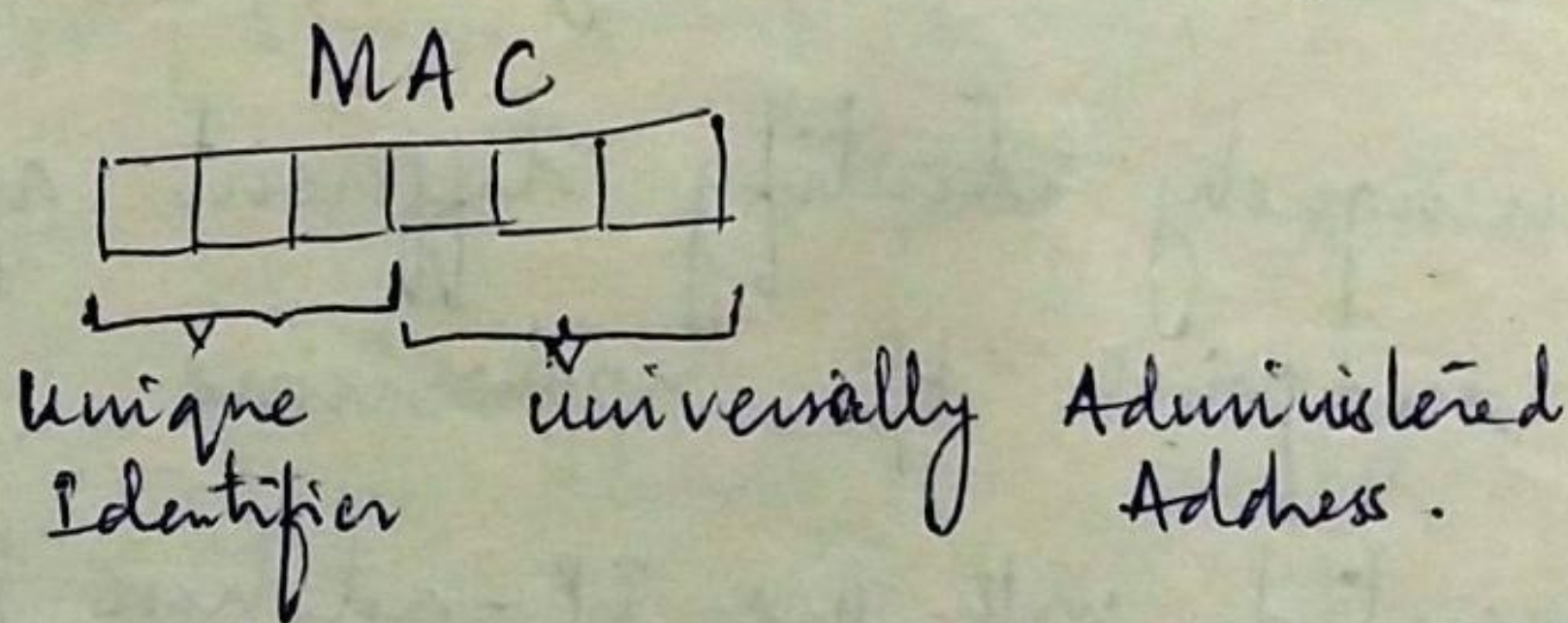
✓ first part of the URL, is called protocol identifier and indicates, what protocol to use.

✓ second part is called resource name and it specifies the IP-address or the domain name, where the resource is located.

✓ The protocol identifier and the resource name are separated by a colon (:) and two forward slashes (//)

MAC address (Physical Address)  
(Medium Access Control) address is a unique identifier of NIC (Network Interface Controller).

• A network node can have multiple NICs, but each with unique MAC.





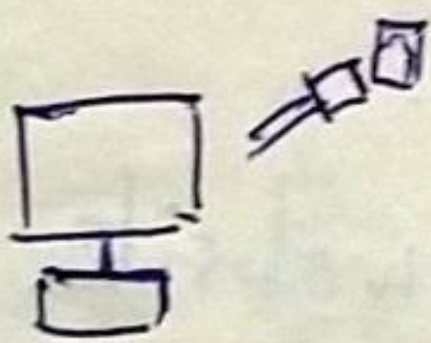
Socket, is an application program, responsible for communication b/w 2-end points.

- A socket is uniquely identified by an IP-address and a Port.

NOTE: A socket is a software element, no hardware.

Socket address

10.14.90.85 : 27017



Socket Address

10.14.90.80 : 8080



## Communication Protocols

1. Connection-less protocol (UDP - User Datagram Protocol)
2. Connection-oriented protocol (TCP - Transmission Control Protocol)
3. TCP-IP (Internet Protocol)
4. FTP
5. HTTP
6. HTTPS (Secure HTTP)
7. SMTP (Simple Mail Transfer Protocol)

## TCP-IP

TCP - it is known to provide reliable and error-free communication b/w end-systems.

- It performs sequencing and segmentation of data.
- It also has acknowledgement feature and ②



and controls the flow of the data through flow-controlled mechanism.

- it is very effective protocol, but has a lot of overhead due to such features. Increased overhead leads to increased cost.

IP - is responsible for delivering packets from the source host to the destination host by looking at the IP-addresses in the packet headers.

- IP, has 2-versions: IPV4 and IPV6.
  - IPV4, is the one that most of the websites are using currently.
  - But IPV6 is growing as the number of IPV4 addresses are limited in number when compared to the number of users.

FTP - is an application layer protocol, which moves the files b/w local and remote file systems. It runs on the top of TCP, like HTTP.

- to transfer a file, 2-TCP connections are used by FTP in parallel:
  - control connection and
  - data connection.
- control connection - user identification, password, store files.
- Data connection - sending actual file, port no. 20 is used for initiating Data connec<sup>n</sup>.



```

Pgm ① MyServer
import java.net.*;
class MyServer
{
    public static void main(String args[])
    {
        try
        {
            ServerSocket ss = new ServerSocket(43);
            S.o.pln("Server is started");
            Socket s = ss.accept();
            S.o.pln("Server is x started: ...");
        }
        catch (Exception e)
        {
            S.o.pln(e);
        }
    }
}

```

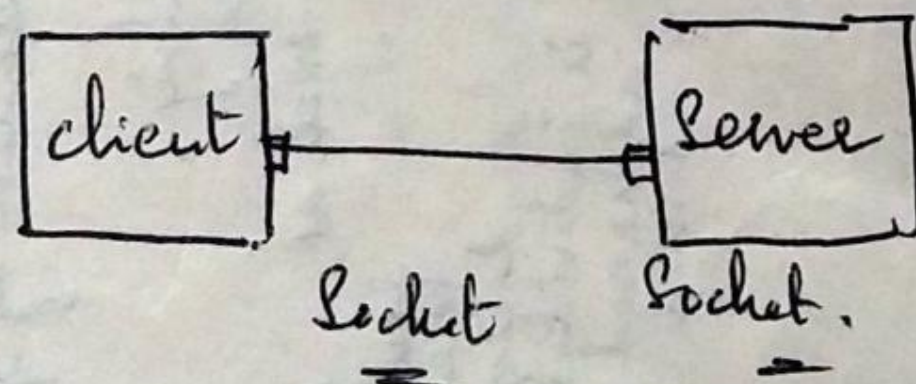
```

MyClient
import java.net.*;
class MyClient
{
    public static void main()
    {
        try
        {
            Socket s = new Socket("localhost", 43);
            S.o.pln(s);
        }
        catch (Exception e)
        {
            S.o.pln(e);
        }
    }
}

```

Simple pgm: connection established b/w client and server. m/cs.

NOTE: ip config.



ip address of server  
but server = localhost.

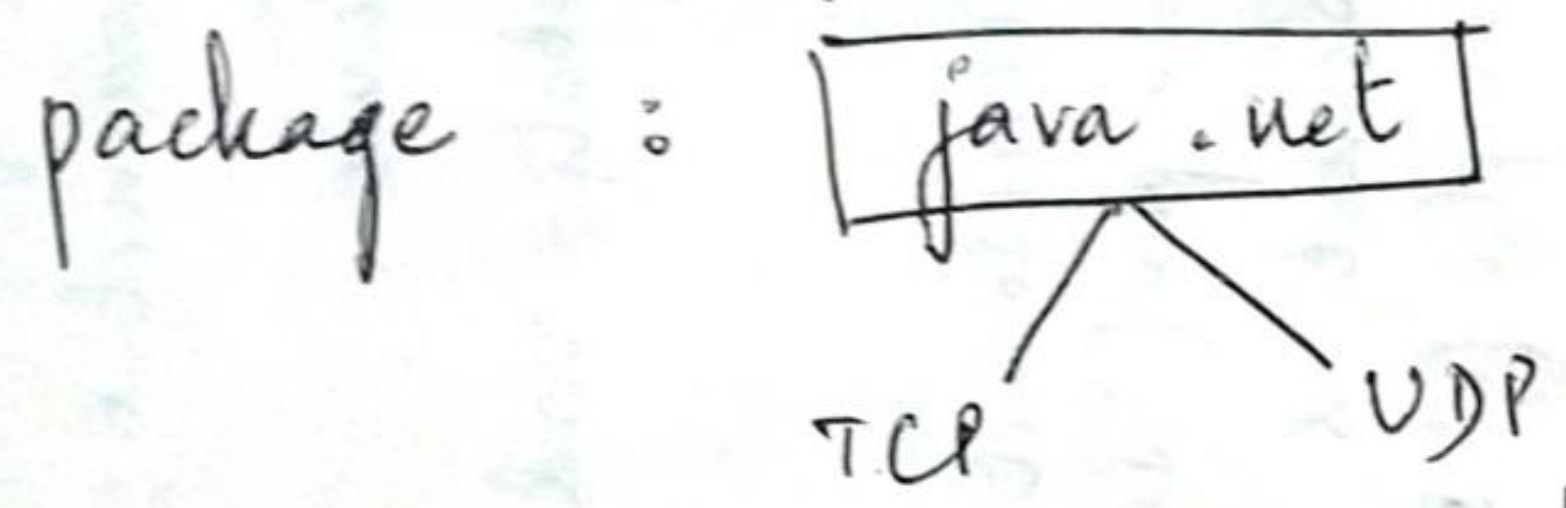


# Socket Programming

IP-addr

Port No.

Socket. → in java → thro' classes → obj → Buffer.



- Socket
- ServerSocket
- more reliable
- has extensive error checking
- require more resources.
- used by services HTTP, SMTP or FTP.

- DatagramSocket (for sending and receiving datagram packet)
- DatagramPacket → datagram packet is represented by DatagramPacket class
- less reliable
- has limited error checking.
- requires less resources.
- used by services such as VoIP.

other classes.

URL  
URLConnection  
HttpURLConnection  
InetAddress

each packet sent or received on a datagram socket is individually addressed and routed.

Analogy:  
chatting module }  
online chat. }

Internal model. - Socket

pgm 2

MyServer

```

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

public class MyServer {
    public static void main ( ) {
        try {
            ServerSocket ss = new ServerSocket(43);
            ss.println("Server started..");
            Socket s = ss.accept();
            ss.close();
            DataInputStream di = new DataInputStream(
                s.getInputStream());

            String msg = (String) di.readUTF();
            ss.println(msg);
        } catch (Exception e) {
            ss.println(e);
        }
    }
}
  
```

MyClient

```

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

public class MyClient {
    public static void main ( ) {
        try {
            Socket s = new Socket("localhost", 43);
            s.close();
            DataOutputStream ds = new DataOutputStream(
                s.getOutputStream());
            ds.writeUTF("hello server");
            ds.close();
        } catch (Exception e) {
            s.println(e);
        }
    }
}
  
```



## JAVA NETWORKING.

- For the networking with Java, it provides 2-types of sockets,
  - stream sockets and
  - datagram sockets.

### ① Stream sockets

- with stream sockets, a process establishes a connection to another process.
- while the connection is in place, data flows between the processes as continuous streams.
- here, stream sockets are said to provide a connection-oriented service.
- the protocol used is TCP

### ② Datagram sockets

- with datagram sockets, individual packets of information are transmitted.
- the transmission of packets follows a connection less service.
- the protocol used is UDP.



## HTTP & HTTPS Hyper Text Transfer Protocol

- it is the underlying protocol used by the world wide web.
- this protocol defines how the messages are formatted and transmitted, and what actions web-servers and browsers should take in response to various commands.

HTTPS means HTTP Secure.

- secure version of HTTP.
- it means, all communications b/w your browser and the website are encrypted.



## classes for Networking in JAVA

- URL
- URLConnection
- HttpURLConnection
- InetAddress
- DatagramSocket
- ServerSocket
- Socket

package : java.net

1. class URL

eg: https://nptel.ac.in/course.php → object.

A URL contains many information

- protocol : https.
- server name or IP address : nptel.ac.in is the server name.
- port no. : optional attribute  
eg: https://nptel.ac.in:80/, 80 is the port no.

if port no. not mentioned in the URL, it will return -1.

- file name or directory name : course.php.

java.net.URL class provides many methods,

public String	getProtocol()	- returns the protocol of the URL
"	getHost()	" host name "
"	getPort()	" port no. "
"	getFile()	" file name "
" URLConnection	openConnection()	" Instance of URLConnection associated with URL. (4)



## ② class `URLConnection`

- represents a communication link b/w the URL and the application.
- this class can be used to read and write data to the specified resource referred by the URL.

\* - methods      `openConnection()`  
                         `getInputStream()`

Syntax : `public URLConnection openConnection() throws IOException { }`

## ③ class `HttpURLConnection`

- Java `HttpURLConnection` class is http specific `URLConnection`.
- it works for HTTP protocol only.
- This class can be used for extracting the following "info" of any HTTP URL,
  - Header information
  - Status code
  - Response code etc.

Syntax      `public URLConnection openConnection() throws IOException { }`  
methods :      `openConnection()`  
                         `getHeaderFieldKey`  
                         :



## class InetAddress

- Java InetAddress class represents an IP-address.
- The java.net.InetAddress class provides methods to get the IP of any host name,  
eg: www.google.com  
www.wikipedia.org...
- The java.net.InetAddress class provides many methods.

eg: InetAddress getByName(String host) throws  
UnknownHostException.

InetAddress getLocalHost() throws UnknownHostException

getHostName()

getHostAddress()...

URLConnection.openConnection()

## class DatagramSocket

- represents a connectionless socket for sending and receiving datagram packets.
- a datagram is an information, but there is no guarantee of its contents, arrival or arrival time. ⑤



constructors,

`DatagramSocket()` throws `SocketException`

`DatagramSocket(int port, InetAddress address)`  
throws `SocketException`.

### class ServerSocket

- `java.net.ServerSocket` class is used by server applications to obtain a port and listen for client requests.

#### constructors

- `ServerSocket(int port)` throws `IOException`
- `ServerSocket(int port, int backlog)` throws `IOException`.
- `ServerSocket(int port, int backlog, InetAddress address)` throws `IOException`.

#### Methods

- `accept()`
- `bind()`
- `getLocalPort()`



class Socket

connect(SocketAddress host, int timeout) throws IOException

InetAddress getAddress()

getPort()

close()