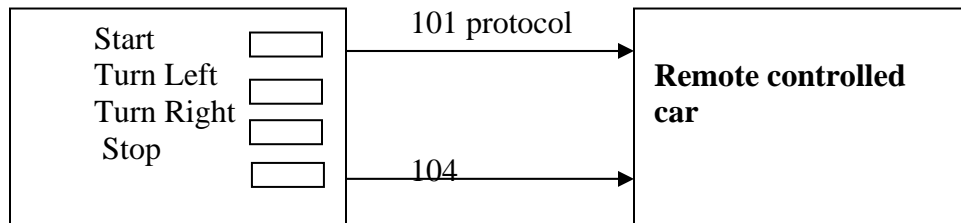


Networking: (java.net package)

How two different devices communicate over the networks: Network is that of multiple devices connected together by means of a physical media (Through network chords).

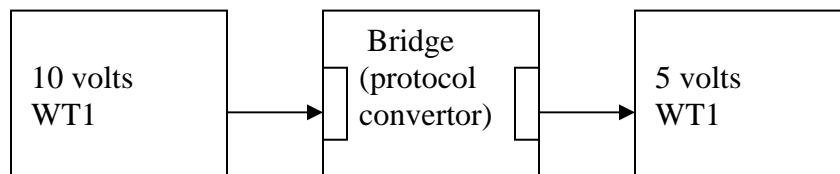


*Protocol is nothing but a set of rules, unless until there is no set of rules, two devices cannot communicate with each other (wireless Technology).

There are two issues here:

1. How the hardware is used to connect two devices i.e., through the chords & through wireless technologies.
2. How the top level technology (software) is used to connect the two devices i.e., 101...104

*The protocol used at the bottom level (hardware) is different for different devices.



*Bridges are used to fill the gap between two devices.

Diverse kinds of setups are used to convert the technologies (or) the top level protocols according to our requirements. They should stick to the particular protocol, when used diverse technology.

Enabling the communication between the diversing setups and the devices:

You are able to test your network programs without having the internet connection. Using the protocols we can do this. If this was tested in one computer, this will work well and good across the globe.

TCP/IP- Transfer control protocol/Internet protocol. This is the most widely used protocol. This is an open standard specification and cannot be controlled by any company.

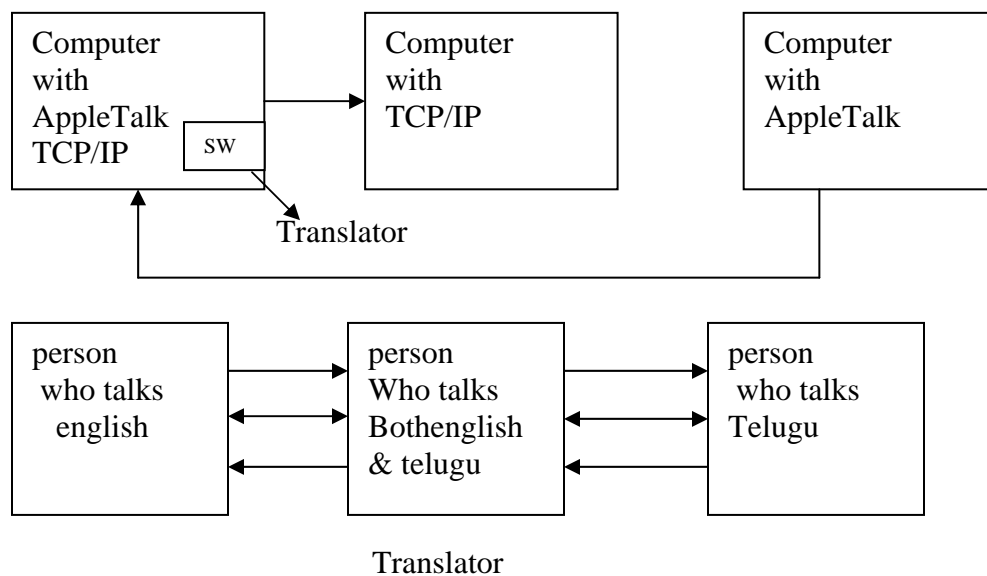
1. Net BUEI- protocol developed by Microsoft

2. IPX/SPX- protocol developed by Novell
3. APPLETALK- protocol developed by Apple computers.

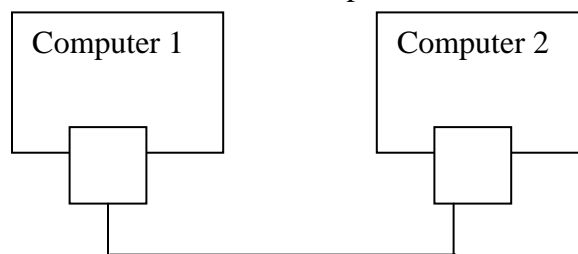
All these protocols are similar to the TCP/IP, but the set of rules that are used is different for different protocols.

In order to enable the communication between two devices, we require set of rules (protocol). In case of computer networks; we have different types of protocols like TCP/IP.

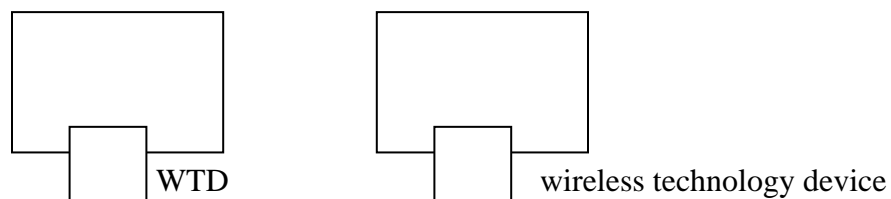
* In order to enable the communication between the two computers, both the computers have to follow the same protocol.



As a developer, we don't bother about how the data transfers from one computer to another computer. Your code in the o/s is responsible to do this.

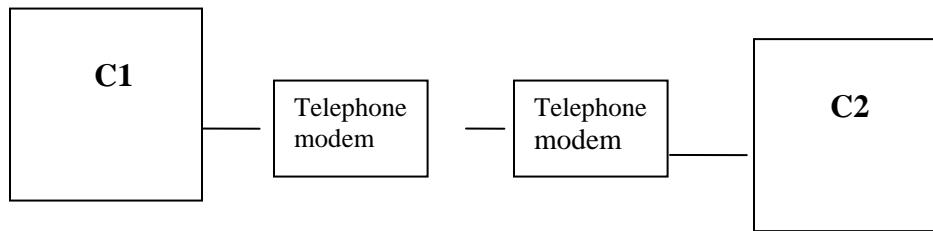


Wireless Technology H/W Devices: -



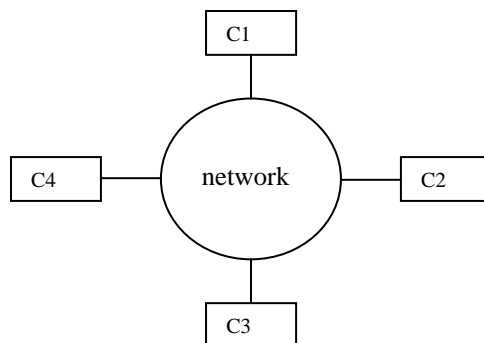
But there is a limitation with the distance.

Through Telephone Lines: -



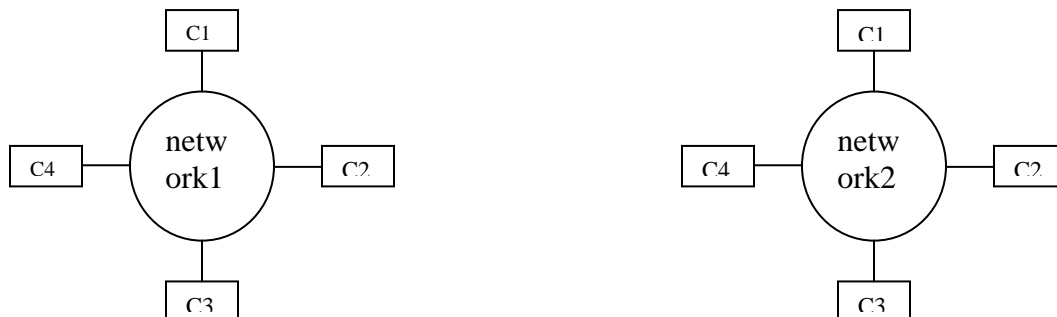
ISDN (Integrated Switching Digital Network):

In this technology, the device should always directly connect to the telephone exchange, and this is connected to the server with a copper cable.



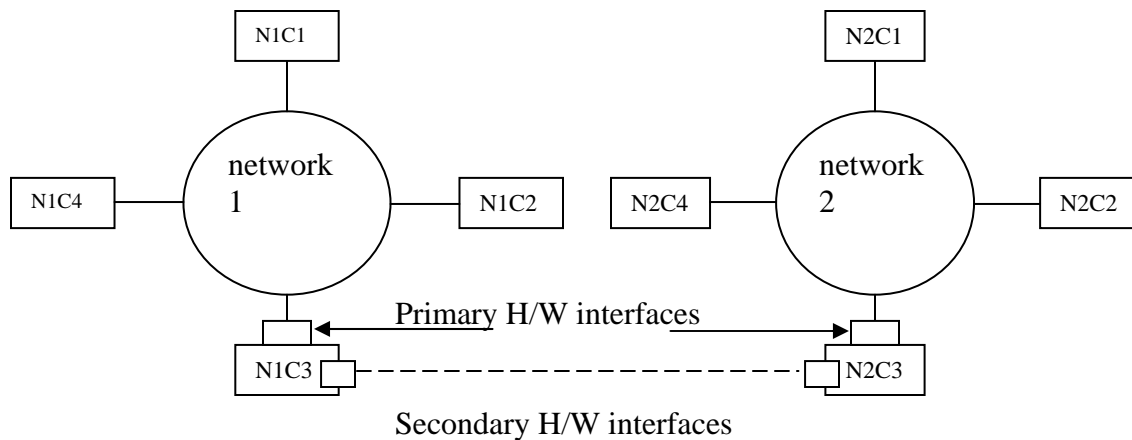
In a network, every computer will be given with an unique address. This address is used to transfer the data from a source to the destination.

Internet: is a collection of networks (or) network of networks.



To identify each computer in a particular network, we should follow the specific address (or) unique address.

Combining the two networks with less expense:



For a single computer .We can have multiple H/W devices.

N1C3 & N2C3 m/c's(machines) will acts as a gateway/passage of data from one m/c to an another m/c.

Gateway:- It is the passage to pass the data from one network to another network.

Rotting: It will perform the routing functions.

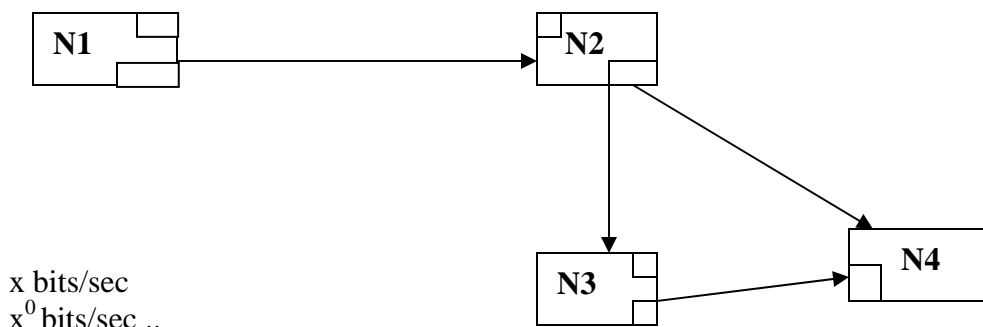
Note: Here we should remember two main points.

1. Addressing Scheme of a Computer.
2. Gateway/Router functions.

Network administrator is responsible to set the 'routing table' which internally sets up the roots of a network.

Just by seeing the network topology, we can't say that, this the shortest root. This depends upon the bandwidth of the channel.

Network:



x bits/sec
x⁰ bits/sec ..

band width or capacity of your channel
100 MBPS data can be passed through the internet.

Internet Engineering Task Force (IETF):

This is a non-profitable organization; this should maintain the specifications of an internet.

There are several governing authorities like IAP (Internet Addressing Protocol), IANA (Internet Assigning Numbering Authority) and so on.

Which are responsible to maintain different processes of an Internet?

Internet Addressing Schemes (IAS)

The key point we should remember in the internet is, some networks are big and some are small. By keeping this in mind this IAS has developed.

1) Internet Address should have two parts, network and computer. Address should be assigned using one's and zero's only. The length of an address should be 32 bits. In this 16 bits are assigned to a network, and 16 bits are assigned to a machine/computer in a network. 16bits $\Rightarrow 2^{16}$ digits.

If we want to go for this IAS, then the size of all the Internets should be the same. But this is not possible because, some internets are small and some are big. In this world smaller networks are much more than the larger networks.

2) 8 bits for a Network & 24 bits for a computer.

2^8 digits for a Network and 2^{24} digits for a computer.

There are four IAS's by seeing the digits, we can able to say to which this network belong to.

Class A -----	0	Network 7 bits	LocalAddress 24 bits
Class B -----	10	14 bits	16 bits
Class C -----	110	21 bits	8 bits
Class D -----	1110	Multicast	28 bits

An IP address consists of 32 bits. A human being cannot remember 32 binary digits. To overcome this difficulty, internet designers have designed 'dotted-decimal-notation'.

Take 32 bits address and divide this into four equal parts. Convert these 8 bits into a decimal numbering system. Then write a decimal number system separating the 8 bits numbers with the dots.

EX: -

01110111 11000111 00011111 01111100 ---- 119.199.31.124----->dotted decimal notation

This belongs to the class A IAS.

* By seeing the first numbers i.e 0, 10, 110 and 1110, we can easily identify the class of an Internet Address Scheme.

EX:

190.12.12.12 to which class this IP Address belongs to?

$190_{(base)10} = 10111110_{(base)2}$

By seeing the first two digits, we can say that this belongs to class B IAS. Here to identify the class, you should not require to calculate the whole number into a binary system. Take the first part and convert this to an equivalent binary number system and see the first two, three or four digits, you can easily identify the class to which a particular IP Address belongs to.

Subnet Mask: Is used to divide a network into a multiple networks. This will be taken by the Network Administrator.

Default gateway: From which the data from this network to another network will be transferred.

Human beings cannot remember the numbers. But it is very easy for them to remember the names instead of the numbers. So IETF has designed the domain names instead of the numbering scheme. Like www.microsoft.com, www.yahoo.com, www.msn.com.....

Any body can register the domain names by consulting the internet registers. For owing a domain name one should not have the internet connection i.e. there is no need of having an internet connection.

If any one having a network of his own, in that every m/c in a network should have a unique IP Address and m/c name. If he has taken the internet connection then he can assign the domain names according to his wish like m/c1.domainname.com, m/c2.domainname.com and so on.

Domain Naming Server (DSN):

DSN server manages the mapping between the domain name and the IP address.

All the domain names and the IP addresses are stored in DNS server. The network administrator is responsible to tell that if one cannot find out the IP address of a particular company in a DNS. Then contact the near by DNS in another network. If it is not available in that network also go to the other near by network at any point of time you can find out that address.

In most of the cases m/c's are configured to reach multiple DNS's on a network.

How the TCP/IP developers develop the s/w: -

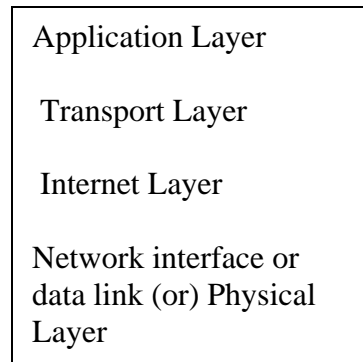
This s/w is designed in such that a way that as multiple layers. A layer is a set of functions and a layer is a logical thing.

DIAGRAM:12

The big advantage with this setup is your l1 can use any functions of l0. But it is not possible use the functions of l1 in l0. Any s/w (complex) will be designed like this only.

Another big advantage of this setup is we can selectively replace a layer.

The TCP/IP s/w is also designed by using a layered architecture. In any layered architecture each and every layer is responsible for carrying out a specific set of tasks.



ADVANTAGES:

Now we are using Ethernet card for an internet connection, in the future we can able to replace with the latest application protocols (Like Wireless).

TCP/IP is not only two protocols, it is a suite of protocols, out of which TCP/IP is most popular protocols.

This is designed to work for multiple networks.

When you are connected to the internet, the ISP provides you a block of IP addresses according to the charge taken by them (how many systems). These IP addresses are unique, and each and every m/c will be given one or more IP address. No two m/c's should have the same IP address.

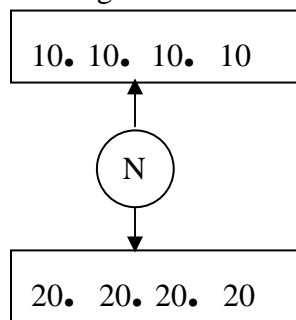
Isolated Network: The network which is not connected to an internet.

Network Administrator is responsible to assign the IP addresses specifically to a local network m/c's or isolated network m/c's. Because if in future if you are connected to the internet, this IP address should not conflict with the internet IP addresses.

Ex: Even though 24 m/c's are available in a network, we have taken the connection from ISP for 14 m/c's only. And we internally connect the internet to the remaining 10 m/c's through the network. In this case

1. 14 m/c's are directly connected to the internet.
2. 10 m/c's are not directly /not logically connected to the internet.

With this m/c's you can not directly access the internet and you cannot write network programs using these m/c's.



Socket API: -(In 1970 Burkely University)

Socket is end point of communication.

Socket API is a set of functions written in 'C' language, for allowing c language programmers to write network programs on UNIX o/s. This was developed by the Burkely University. Later on every o/s vendor has started providing the implementation of C-language libraries with these functions.

This is a very popular API available for almost all the platforms. JavaSoft has created a set of classes as part of "java.net" package, which uses socket API internally.

JAVA programmers can write network programs using TCP, UDP protocols.

DIAGRAM:15

Server: Server is a program which provides some services to the client.

Client: Client is a program which requests server for some services i.e. for data from the database.

Programmers can write functions, he can write single byte or thousand bytes at a time. But we are not sending the whole data in a single shot to a client.

The data is converted into several smaller packets by TCP layer, and then it is going to add the headers (i.e. source address, destination address, packet numbers and so many other things like error checking code and the data).

Error code is a number.

TCP - Connection Oriented reliable protocol

UDP - Connectionless unreliable protocol

IP layer is responsible to take the routing/gateway for transferring or routing the data. TCP is responsible for transferring the data i.e. how to transfer the packets from the source to the destination.

When the data is received by the client it will be received as it is the data is handed it over to an application. When the data is reached to a client, this will be checked by an error code whether the data is correct or not and the top level protocol id which is used to send the data.

TCP protocol is responsible for sending the acknowledgement back when it has received a packet. When a TCP layer is unable to deliver the packet, it will simply return error message to an application.

When you are using UDP, there is a little bit of chance of losing the data. If we are ready to bare the losses, it is ok to proceed with this kind of protocol.

In TCP there are two techniques available to receive and to send the data.

1. Input buffers
2. Output Buffers

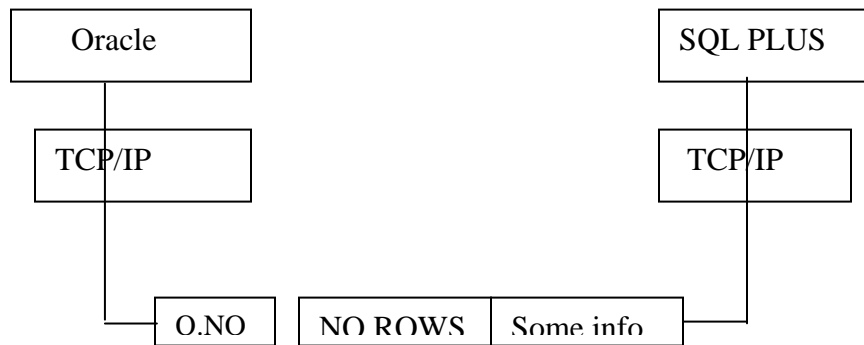
These two are the memory areas managed by the TCP.

Normally, when an application writes less amount of data, the TCP s/w collects the data in the output buffer for pre-defined amount of time or till pre-defined amount of data is collected. In some cases a programmer can give an instruction to the TCP layer to pass the data immediately after writing the data.

According to our requirements, we can give an instruction to TCP to send the data immediately char by char or byte by byte or at a time. These are known as "socket options ".

In some cases the programmer may not be ready to receive the data. In such a case the TCP s/w stores the data in an input buffer till the program reads it.

Oracle supports multiple protocols i.e. oracle database as a server and SQL-PLUS as a client. Here Oracle acts as server as well as a client.

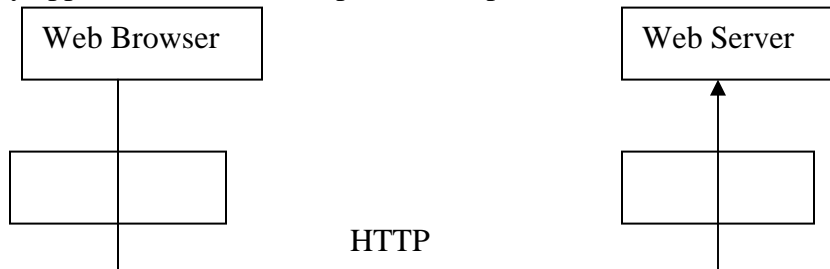


SQL-server also works as similar to oracle. But the way in which oracle server transfers the data to the SQL-PLUS is different between the SQL-server transfers the data.

Oracle server has its own protocol.

Oracle sever has its own format, such that oracle server expects the data sent and the client to receive the data in their own format or using specific rules.

Every application has its own protocols (specified format) to send and receive the data.



Web server acts as a server and Web browser acts as a client.

Most popular Web servers in the world are:

1. Apache Webserver
2. IIS Webserver
3. Netscape

HTTP (Hyper Text Transfer Protocol) is used in the webserver to transfer the data according to its format.

HTTP is an open protocol i.e. this is not maintained by any company. A non-profitable organization W3C (World Wide Web Consortia, w3c.org) maintains the details of protocols i.e. web related specifications are managed by this organization.

Application Level Protocols: All the specific formats of each and every particular protocol is documented in the application level protocol.

Whenever a network application has to be implemented, we have to decide about the protocol that has to be used in both the client and server. These protocols are known as "Application Level Protocols"

Ex: web browser and a webserver uses a protocol called HTTP in order to talk to each other.

How the JAVA language represents IP addresses:

In java language, we have a class with the name "InetAddress" which is used to represent the internet addresses/IP addresses.

Multi_homed host: If a single m/c is configured with multiple IP addresses, then it is known as a "multi_homed host".

Host m/c: m/c that is not connected to a network is known as "Host m/c" in a network terminology.

Loop pack address: Always refers to the local m/c or host m/c. This is same as for every host m/c. 127.0.0.1

ex Address.java

InetAddress.getByName("Localhost") gives the IP address of a host m/c.

Hosts file: In almost all the o/s's, when we use getByName() method, first it will check for a local file named 'hosts' for IP addresses. If it fails to find a match in this file, then it contacts the nearest DNS sever, and maps the IP address and the name.

WinNT allows to configure multiple IP addresses, but Win98 doesn't allow multiple IP addresses to a single h/w interface.

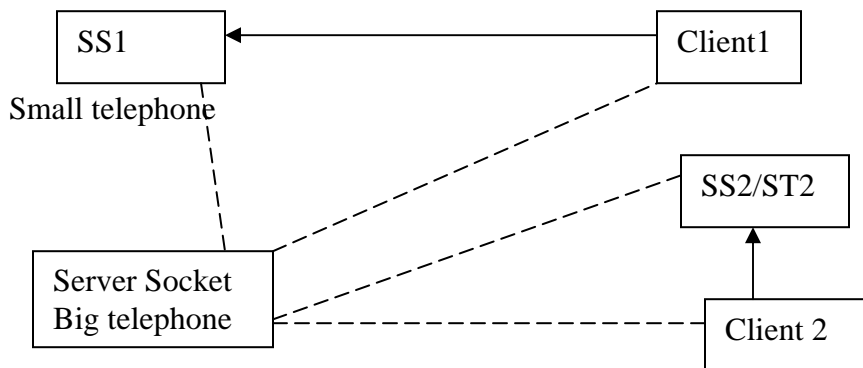
`InetAddress.getAllByName("localhost")` gives all the IP addresses assigned to a particular m/c.

Server Socket: Contains the IP address and port number (196.12.100.100 & 80). Port number should be an integer.

Why port number?

Whenever we write a server program, we have to choose a particular port number which is not currently used by some other application.

If multiple s/w's are running on the same m/c (IP address), it is very difficult to know which s/w is required for us. To differentiate multiple softwares we need a port number.



Whenever a client connected to the server socket, a new telephone will be created and the client is directly connected to a new telephone for communication links.

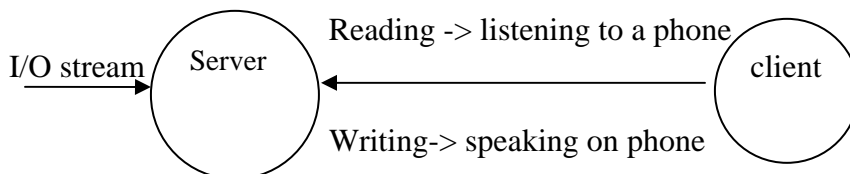
On server side we need to create server socket.

ex, server2.java, client2.java

```
ServerSocket ss = new ServerSocket (16000 , 1);
```

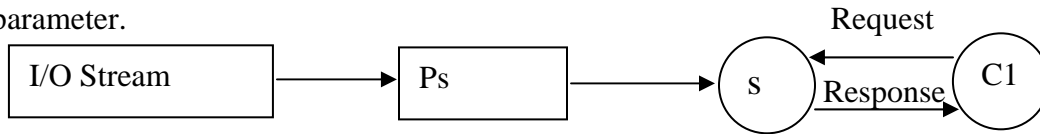
Port number without IP address will give all the IP addresses that are configured to the local m/c.

```
Socket s= ss.accept();
```



Using this new socket we can read and write the data.

Socket class provides the methods `getInput` and `getOutput` streams associated with a socket .we can use these streams inorder to read and write the data through this connection. In order to read and write the data `PrintStream` requires another stream as parameter.



In TCP/IP programming a client has to establish the connection with the server in order to communicate with the server. Once the connection is established the client can read and write the data through the connection.

In order to establish the connection with the server one should know the IP address and port number.

There is no need to explicitly mention the port number; it automatically connects the port number which is currently free. But in the case of server, we have to explicitly mention the port no, because to know which server is currently running.

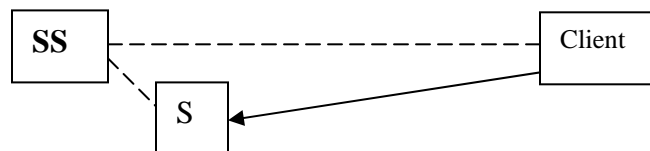
Implement the following protocol in a network server and client.

A client after establishing the connection after the server has to pass a message as shown below. `Hello server \r\n`

The server after receiving this has to pas Good Morning/ Afternoon/ evening depending upon the time.

```

ServerSocket ss = new ServerSocket ();
while(true)
{
    Socket s = ss.accept();
    // code to service the client
}
  
```



If this client takes 10 minutes to complete the task, until unless the first client completes the task no other client should not be able to request the server i.e multiple clients cannot get the services from the server at the same time.

How to solve this:

```

ss = new ServerSocket();
while(true)
  
```

```

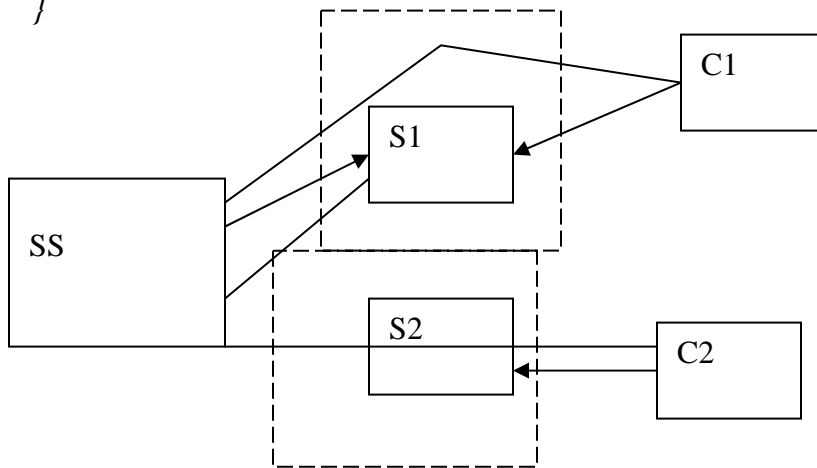
{
Socket s= ss.accept();
create a thread services the request
}

```

```

class Servicecode extends Thread
{
run();
// here is the code to service the client
}
}

```



In this case by using threads we can able to serve multiple clients concurrently.

Problem with this approach: -

Creation of thread is an expensive operation because it takes lots of time to create a thread.

Ex: If one minute is required to create a thread and any client is trying to call then it will get an error message. In this case the o/s is responsible to manage the pending calls. If another client requests a call rather than this pending call then the o/s tells the server that you have one pending call and that call will be received and processed first.(Find the length of the queue).

```

ss = new Serversocket(16000,1);
    here 1 is the length of the queue.

```

o/s have some limit to queuing the pending calls. Default is 5 or 10 depending upon the o/s.

The TCP/IP s/w is responsible for managing the pending connection requests that are arrived from the clients. A java programmer while creating the server socket can mention

the length of the queue. Pending requests are accepted up to pre-defined amount of time only. Maximum length of the queue will depend upon the o/s and configuration of your m/c.

server3.java

Every client/thread has it's own socket. i.e for every thread socket will be passes as a parameter and your thread will take care about the client.

Creation of threads are costly, so the threads are pre-created and these will be used in the applications.

Telnet: Telnet is a application designed to access systems from remote.

This can also be used as a diagnostic tool to find out whether a client can reach a server or not.

We can find out whether the program is running on particular port number or not.

MENU ->CONNECT->REMOTE SYSTEM

Host Name	<input type="text" value="localhost"/>
Port	<input type="text" value="80"/>
Term type	<input type="text"/>
<input type="button" value="connect"/>	<input type="button" value="cancel"/>

This utility allows us to run the protocols from here and these can be tested.

Type :

GET/HTTP/1.0 <enter key>

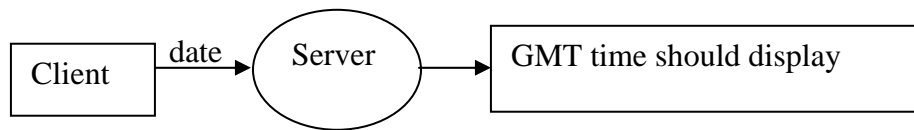
Empty Space <enter key>

PRACTICAL PROBLEMS DURING THE SERVER PROGRAMS:

```
ps.println("some long text here "); // server
is.read();                          // client
```

In this case you can read partial amount of data because the underlying protocol may not read the whole data in a single shot. This cannot work in almost all the cases. It may read the whole data in some cases.

To read the whole details in a single shot our protocol is designed in such a way that it should be designed by taking care about each and every minued details.



```
ps.print("Let us say here is the data \r\n");
```

Client program should be written such that it should read the data until \r\n is reached.
client program:

```
do
{
a=is.read();
}while(a!=0);
```

but the method call is expensive.

Remedy:

```
while(true)
{
a=is.read(b,0,100);
if((b[a-1])=='\n' &&(b[a-2]=='r'))
break;
}
```

Most of the programmers tend to write network programs making assumptions like the data return by the method `ps.print()` will be sent to the client in a single shot. So, when we design a protocol the protocol should take care about each and every minued details and the server and the client should be written according to the protocol.

HTTP - to transfer the data by Webservers

FTP (File Transfer Protocol) - To download files

SMTP (Simple Mail Transfer Protocol)-To send the mails from a client to the server

POP3 (Post Office Protocol) - to receive mails

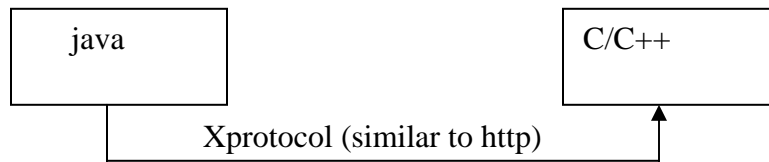
IRC (Internet Relay chat)-for chatting purpose

NNTP (Network News Transfer Protocol)-To display the news

Internet protocol designers have reserved some well known port numbers, in order to run different standard protocols.

Ex :

Default port no for running HTTP server is 80.
For Weblogic port no is 7001.



Web Browsers: Browsers can understand protocols and gives response. By default HTTP protocol is used

1. port no : 80 IIS server
2. 7001 For WebLogic

One of the services provided by WebLogic is HTTP.

When we use the port no, for every port no this program opens some socket at that particular no's.

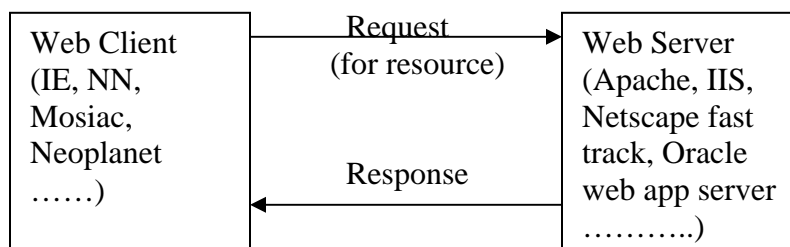
HTTP://localhost:7001 at the Internet Explorer address bar , it displays the WebLogic server.

How these client & server uses Http Protocol: -

Client-> sends the requests in a specified format, to the server.

Server-> sends back the response in a specified format tot the client.

Http server is nothing but a web server. Http client is nothing but a web client.



Http protocol is available in two versions:

1. Http-1.0 and 2.Http-1.1

Http1.1 is little bit of extension to the Http1.0

Java webserver is abonded by JavaSoft. So, never say that we have used the Java webserver to develop the network programs.

Jigsaw webserver is an open specification. If you want the source code of this webserver simply go to w3c.org/jigsaw and download this.

Request will be sent in following format:-

```
Initial Request line
Header 1
Header 2
:
:
:
Header n
Blank line ( carriage return line feed)
```

Ex: - GET/index.html/HTTP/1.0

Response will be sent in the following format:

```
Initial Response line (status line) Example: HTTP/1.0 200 OK
Header 1
:
:
:
Header n
Blank line
Message body
```

eg:- 1) type `http://localhost:8080/` <enter key>
in the address bar of an internet explorer

Headers are used to convey additional messages regarding the web clients.

We will see the current browser that is used by the user/client

Type the same URL in the Netscape Navigator address bar, then you will observe the output.

2) type `http://localhost:8080/postform.html` <enter key>

If the resource (postform.html) is available in that particular server, it will receive the request and displays the response according to the request received.

Http clients use different methods to interact with the http servers. These http methods are GET, POST, HEAD, DELETE, PUT, OPTIONS and TRACE etc.

These headers will give additional information about the requests.

1. ACCEPT header: -

ACCEPT: image/gif, image/x-xbitmap, */*

Here ACCEPT is the name of the header and the remaining is the value of the header.

Image/gif in that image is the main file and gif is sub file. Name of the header and the value of header must be separated by a colon.

This header shows what are the contents that can display will be displayed as part of the header files.

2. ACCEPT language:

Accept language: en-us

Web server may encode the data according to the US language.

ACCEPT - ENCODING: gzip, deflate

This says that I can accept to encoding data and decoding in lthis particular format.

3. USER-Agent:-

User-Agent: Mozilla/4.0/IE or NN

Used to give an information about which browser is currently used by the client.

Why Mozilla will be displayed before IE or NN:

Mozaic was developed by Mark underson, and IE and NN are developed by taking the source code from Mozaic. That's why they acknowledged this with their browsers.

4. HOST : Localhost or Localhost:8080:

Domain name of the server to which the request has to be sent

5. CONNECTION : Keep - Alive: -

Web server requesting the client do not cut the connection. But the web servers are designed in such a way that, once if the connection is established between the client and the server automatically the connection will be cut off. That's why these are called connection less or stateless web servers.

POST Method to submit lthe data:

Referer:- http://postform.html <enter key>

Content-type and Content-length ->whatever the data that you have send through post method is passed as a content.

Content-length will display the length of the content. Content will be sent as a part of the content body.

When we develop web applications, the form data can be submitted using GET or POST methods. The data is submitted using the request type POST as shown below

POST / abc.jsp Http/1.0

Header 1
:
:
Header n
blank line --> (\r\n)

Form data

- . (Huge amount of data can be sent) ---> body
- . Security is encountered
- . Data is sent as a standard input.

GET method: -

The data submitted using GET method will be appended as part of the URL. The web clients use http GET method to pass the data as shown below:

GET/xyz.asp? formdata Http/1.0
Header 1
:
:
Header n
blank line ---> (\r\n)

- . Limited data can be sent (1024 bytes)
- . No security
- . Data is appended at the end of URL

How the web server sends a response back to a clients:-

Telnet:-

Goto Terminal ---> start logging (port no 80)

1. GET/HTTP/1.0 and press enter

observe the output

status code : http/1.0 200 ok

200 is the status code for success of the request.

Server: Microsoft - IIS/4.0 --> web server used.

Content-Type : text/html

Type of the content that is used html or text or bmp or gif or some other file.

Content-Length : 4350 ---> length of the content blank line.

Content-Body:

2. GET Bag2.gif/Http/1.0 press enter

Content-Type : image/gif

3. Status Code 404 -> object not found and Request is failed.

How to request for a file on the server: -

- 1) make a connection
- 2) send a request.

In case of Http/1.0

```
ps.println("GET/HTTP/1.0 \r\n\r\n");
```

with HTTP/1.1 protocol every request should have at least one header host pointing to the domain name.

Now the request line should be:

```
"GET/HTTP/1.1 \r\n Host:inetsolv \r\n\r\n"
```

Web Application Development:

1. We will be choosing a web server.
2. Web server is configured by the web administrator
3. When we install IIS web server, by default IIS automatically picks up the documents from *inetpub/wwwroot* folder.

In UNIX, we will call this folder as Document Root.

How to configure a WebLogic:

```
c:\weblogic\myserver\weblogic.html  
http://localhost:7001/weblogic.html
```

The document root for WebLogic set is: *c:\weblogic\myserver\public-html*

How to configure IIS4.0:

When we use URL like *http://xyz.com/* we will get a resource depending upon the web server configuration.

In most of the web servers, web administrators configure them not to display the list of files in a directory.