

CHAPTER – 6

NETWORKING IN JAVA

NETWORKING BASICS:

Java Networking is a concept of connecting two or more computing devices together so that we can share resources. Java socket programming provides facility to share data between different computing devices.

Advantage of Java Networking

- Sharing resources
- Centralize software management






NETWORKING TERMINOLOGY

1. IP Address:

IP address is a unique number assigned to a node of a network example: 192.168.0.1. It is composed of octets that range from 0 to 255. It is a logical address that can be changed.

2. Protocol:

A protocol is a set of rules basically that is followed for communication. For example:

-  TCP
-  FTP
-  Telnet
-  SMTP
-  POP etc.

3. Port Number:

The port number is used to uniquely identify different applications. It acts as a communication endpoint between applications. The port number is associated with the IP address for communication between two applications.

4. MAC Address:

MAC (Media Access Control) Address is a unique identifier of NIC (Network Interface Controller). A network node can have multiple NIC but each with unique MAC.

5. Connection-Oriented And Connection-Less Protocol:

In connection-oriented protocol, acknowledgement is sent by the receiver. So it is reliable but slow. The example of connection-oriented protocol is TCP. But, in connection-less protocol, acknowledgement is not sent by the receiver. So it is not reliable but fast. The example of connection-less protocol is UDP.

6. Socket:

A socket is an endpoint between two way communications.

SOCKETS:

Socket is an object which establish a communication link between two ports. A *socket* is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

The working mechanism of socket is shown in the figure below:

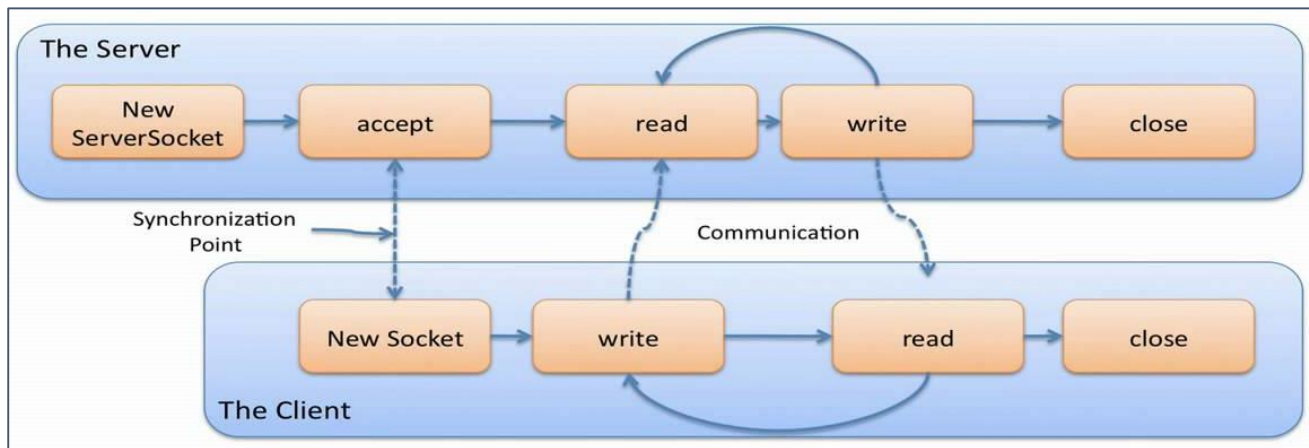


Fig: Overview of Java Socket

PORTS:

Client connect to server through object known as ports. A port server as a channel through which several client can exchange data with the same server or with different servers. Ports are usually specified by numbers. Some ports are dedicated to special server or task. For example: many computers reserve port number 13 for day/time server which allow client to obtain date and time. Port number 80 is reserved for a web server.

Most computers also have 100's or even 1000 of free ports available for use by network application.

PROXY SERVER:

It is an intermediate sever computer that offers a computer network services to allow client to connect with different server. A client connect to the proxy server and then request various services such as a file, connection, web page or other resources available on different server.

The proxy provides the resource either by connecting to a specified server or by serving it from a cache. If the resources have been cached before then the proxy server return them to the client computer. However, if it is not cached it will connect to the relevant servers and request the resources on the behalf of the client computers. Then, it cached resources from the remote servers which in term response to the client.

The advantages of proxy servers are:

- a. **Improve Performance:** It can improve performance for group of users because it saves the results of all request for certain amount of time.
- b. **Filter Request:** It can filter the content also so, it can be used as a filter or firewall.

- c. **Increased Browsing Speed:** It can increase browsing speed due to the concept of caching.
- d. **Security:** It acts as an intermediate between user's computer and the internet to prevent from attack and unexpected access. It can also hide IP Address of the client computer.

INTERNET ADDRESS URL:

The URL allow uniquely identifying or address information on the internet. Every browser uses them to identify information on the web. Within java network class library, the URL class provides a simple, concise API to access information across the internet using URL. All URL share the same basic format although some variation is allows.

A URL specification is based on four components:

- ☑ The first is protocol to use, separate from rest of the locator by a colon. Common protocols are http, ftp, gopher, etc.
- ☑ The second component is the host name or IP Address of the host to use. This is delimited on the left by double slashed (://) and on the right by a (/) or a colon (:).
- ☑ The third component is the port number and is an optional parameter. It default to port 80, the predefine http port.
- ☑ The fourth part actual file path.

Example:

```
import java.net.*;
class URLLDemo{
    public static void main(String [] args) throws MalformedURLException{
        URL hp = new URL ("http://localhost/project");
        System.out.println("Protocol: "+ hp.getProtocol());
        System.out.println("Port: "+ hp.getPort());
        System.out.println("Host: "+ hp.getHost());
        System.out.println("File: "+ hp.getFile());
    }
}
```

Implementing TCP/IP based Server and Client:

TCP/IP sockets are used to implement reliable bidirectional persistent, point to point, stream based connection between hosts on the internet. There are two kinds of TCP sockets in java: one is for server and the other is for the client.

The following steps occurs when establishing TCP connection between two computer using sockets:

1. The server initialize ServerSocket object, denoting in which port number communication is to start.
2. The server invokes the accept () method of the ServerSocket class this method waits until a client connects to the server on the given port.

3. After the server is waiting a client instantiate a socket object specifying the server name and port number to connect to.
4. The constructor of the socket class attempt to connect the client to the specified server and the port number, if communication is established the client now has socket object capable of communicating with the server.
5. On the server side the accept () method returns a reference to a new socket on the server that is connected to the client socket.

After the connections are established communication can occur using input/output stream. Each socket has both an output stream and input stream. The client output stream is connected to the server input stream and the client input stream is connected to the server output stream.

TCP is a two way communication protocol so data can be send across stream at the same time.

DATAGRAM (DATAGRAM PACKET, DATAGRAM SERVER AND DATAGRAM CLIENT):

Datagram packet class represents a datagram packet. They are used to implement a connectionless packet delivery service. Each message is route from one machine to another based on solely on information contained within the packet. Multiple packet sent from one machine to another might route differently and might arrival in any order and the packet delivery is not granted.

Datagrams are bundle of information passed between machines once the datagram have been released to its intended target, there is no assurance that it will arrive or even that someone will be there to catch it. Likewise, when the datagram is received there is no assurance that it has not been damage in transmit or that whoever sent it still there to receive a response.

Java implements a datagram on top of the UDP protocol by using two classes as; the Datagram packet object is the data container and the datagram socket is the mechanism used to send and receive datagram packet.

URL CONNECTION:

URL Connection is a general purpose class for accessing the attribute of a remote resources. Once we make a connection to a remote server, we can use URL Connection to inspect the properties of the remote object before actually transporting it locally. These attributes are exposed by the http protocol specification and also make sense for URL object that are using the http protocol.

To access the actual bits or content information of URL, we create a URL Connection object from URL and using it open connection method is invoke.

Example:

```
import java.net.*;
class URLEDemo1{
    public static void main(String [] args) throws MalformedURLException{
        URL hp = new URL("https://www.facebook.com");
        URLConnection hpCon = hp.openConnection();
```

```
        int len = hpCon.getContentLength();  
        System.out.println("Content Length: "+len);  
    }  
}
```