

UNIT II JAVA NETWORKING FUNDAMENTALS

- Overview of Java Networking

 - TCP

 - UDP

 - InetAddress and Ports

- Socket Programming

- Working with URLs

- Internet Protocols simulation

 - HTTP

 - SMTP

 - POP

 - FTP

- Remote Method Invocation

- Multithreading Concepts [Already discussed in unit 1]

Presented by,
B.Vijayalakshmi
Computer Centre
MIT Campus
Anna University

Overview of Java Networking

- TCP
- UDP
- InetAddress
- Ports

Introduction to Java Networking

- A **network** is a collection of devices that share a common **communication protocol** and a common communication medium (such as network cables, dial-up connections, and wireless links)
- **Network programming** refers to **writing programs that execute across multiple devices** (computers), in which the devices are all connected to each other using a network.
- Networking in Java is mainly used for sharing the resources and also for centralized software management
- Java program communicates over the network at the application layer.
- **java.net package** is useful for all the Java networking classes and interfaces.

Networking Terminologies

- The widely used Java networking terminologies used are as follows:
 - Protocol
 - Connection-oriented and connection-less protocol
 - MAC Address
 - IP Address
 - Port Number
 - Sockets

Protocol

- A protocol is a **set of rules** followed for communication.
- The java.net package provides support for the two common network protocols –
 - **TCP** (Transmission Control Protocol)
 - This allows for **reliable communication** between two applications.
 - **UDP**(User Datagram Protocol)
 - a **connection-less protocol** that allows for packets of data to be transmitted between applications.
- Some of the different kinds protocol s available are,
 - **HTTP** (Hyper Text Transfer Protocol) ? Enables interaction with the Internet
 - **FTP** (File Transfer Protocol) ? Enables transfer of files between computers.
 - **SMTP** (Simple Mail Transfer Protocol) ? Enables email facility
 - **NNTP** (Network News Transfer Protocol) ? acts as a bulletin board for sharing news

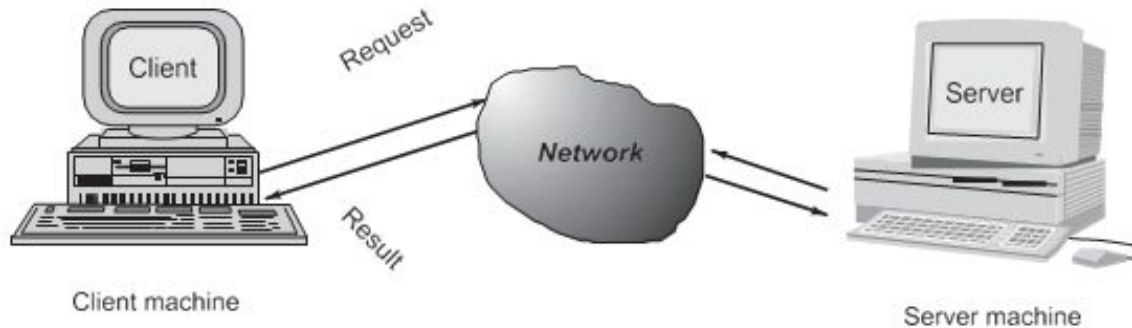
Connection-oriented and connection-less protocol

- In the **connection-oriented protocol, acknowledgment is sent by the receiver.**
- So it is reliable but slow.
- The example of a connection-oriented protocol is TCP.
- TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.
- But, in **the connection-less protocol, acknowledgment is not sent by the receiver.**
- So it is not reliable but fast.
- The example of a connection-less protocol is UDP.

MAC Address

- A MAC address is basically a **hardware identification number** which uniquely identifies each device on a network.
- For example, an Ethernet card may have a MAC address of 00:0d:83:b1:c0:8e.

Client/Server Communication



- At a basic level, network-based systems consist of a server, client, and a media for communication as shown above.
- A computer running a program that **makes a request** for services is called **client machine**.
- A computer running a program that **offers requested services** from one or more clients is called **server machine**. [Multithreading is used to serve multiple client users at the same time]
- The media for communication can be wired or wireless network.
- The client must know two information,
 1. IP Address of the server
 2. Port number

IP Address

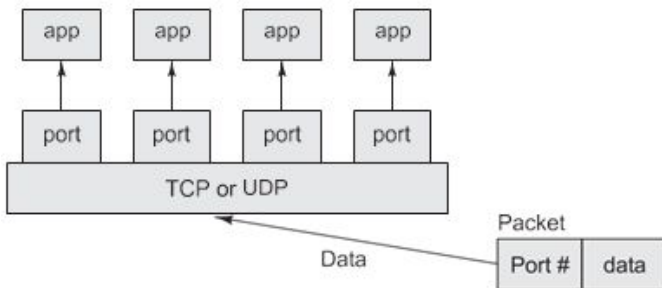
- **Every computer on the Internet is identified by a unique, 4-byte IP address.**
- This is typically written in dotted quad format like 128.250.25.158 where each byte is an unsigned value between 0 and 255.
- This representation is clearly not user-friendly because it does not tell us anything about the content and then it is difficult to remember.
- Hence, IP addresses are mapped using DNS (Domain Name System) to names like `www.google.com` or `www.annauniv.edu`, which are easier to remember.

Port Number

- In general, **each computer only has one Internet address.**
- However, **computers often need to communicate and provide more than one type of service** or to talk to multiple hosts/computers at a time.
- For example, there may be multiple ftp sessions, web connections, and chat programs all running at the same time.
- **To distinguish these services, a concept of ports, a logical access point, represented by a 16-bit integer number is used.**
- That means, each service offered by a computer is uniquely identified by a port number.
- Each Internet packet contains both the destination host address and the port number on that host to which the message/request has to be delivered.
- The host computer dispatches the packets it receives to programs by looking at the port numbers specified within the packets.

Port Number Cont'd

TCP/UDP mapping of incoming packets to appropriate port/process



- IP address can be thought of as a house address when a letter is sent via post mail and port number as the name of a specific individual to whom the letter has to be delivered.
- Some ports have been reserved to support common/well known services:
 - ftp 21/tcp
 - telnet 23/tcp
 - smtp25/tcp
 - login513/tcp
 - http 80/tcp,udp
 - https 443/tcp,udp
- User-level process/services generally use port number value ≥ 1024 .

InetAddress

- **Java InetAddress class represents an IP address**
- The `java.net.InetAddress` class **provides methods to get the IP of any host name** *for* `www.google.com`, `www.mitindia.edu`, etc.
- An instance of `InetAddress` represents the IP address with its corresponding host name.
- **Inet4Address and Inet6Address** are subclasses of the `InetAddress` class.
- `Inet4Address` and `Inet6Address` represent IPv4 and IPv6 addresses, respectively.
- However, when writing network applications, we don't have to concern about IPv4 or IPv6 as Java hides all the details.
- The `InetAddress` can refer to either `Inet4Address` or `Inet6Address`, so most of the time using `InetAddress` is enough.

Commonly used methods of InetAddress class

Method	Description
public static InetAddress getByName(String host) throws UnknownHostException	it returns the instance of InetAddress containing LocalHost IP and name.
public static InetAddress getLocalHost() throws UnknownHostException	it returns the instance of InetAddress containing local host name and address.
public String getHostName()	it returns the host name of the IP address.
public String getHostAddress()	it returns the IP address in string format.

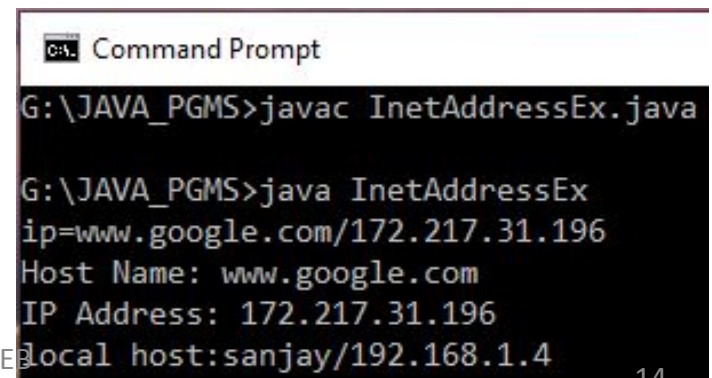
```

import java.io.*;
import java.net.*;
public class InetAddressEx
{
    public static void main(String[] args)
    {
        try{
            InetAddress ip=InetAddress.getByName("www.google.com");
            System.out.println("ip="+ip);
            System.out.println("Host Name: "+ip.getHostName());
            System.out.println("IP Address: "+ip.getHostAddress());

            InetAddress ip2=InetAddress.getLocalHost();
            System.out.println("local host:"+ip2);

        }catch(Exception e){System.out.println(e);}
    }
}

```



```

C:\> Command Prompt

G:\JAVA_PGMS>javac InetAddressEx.java

G:\JAVA_PGMS>java InetAddressEx
ip=www.google.com/172.217.31.196
Host Name: www.google.com
IP Address: 172.217.31.196
local host:sanjay/192.168.1.4

```

Socket

- A **socket** is one endpoint of a **two-way communication link** between two programs running on the network.
- An endpoint is a combination of an IP address and a port number.
- **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.