CST8221 – Java Application Programming
Hybrid Activity #12
Network Basics

Terminology

The term network usually means a set of computers and peripherals (printers, modems, plotters, scanners, and so on) that are connected together by some medium and can send data and receive data from each other, more or less in real time. The connection can be direct (cable, fiber optic and so on) or indirect (through a modem). Usually, if the devices on a network are in a single location such as a building or a group of rooms, they are called a local area network, or LAN. LANs usually have all the devices on the network connected by a single type of network cable. If the devices are scattered widely, such as in different buildings or different cities, they are usually set up into several LANs that are joined together into a larger structure called a wide area network, or WAN. A WAN is composed of two or more LANs. Each LAN has its own network cable connecting all the devices in that LAN. The LANs are joined together by another connection method, often high-speed wired or wireless lines or very fast, dedicated network cables called backbones. Depending on their scale, scope and purpose other classifications include metropolitan area network (MAN), personal area network (PAN), virtual private network (VPN), campus area network (CAN), storage area network (SAN), and many others.

Each machine on a network is called a *node*. Nodes that are fully functional computers are also called *hosts*. Every network node has an address. There are two types of addresses: hardware addresses and Internet addresses. Nodes can have names, and a particular name normally refers to exactly one address. However, names can change and one address can have more than one name.

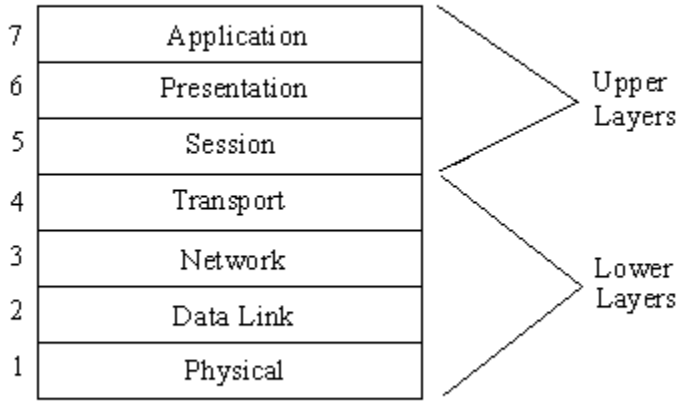
The Nature of Things

All modern networks are packet-switched networks. This means that data traveling on the network is broken into pieces called packets, and each packet is handled separately. Therefore, each packet must contain information about its source and destination.

The different devices on the network communicate with each other through a predefined set of rules called *protocol*. A protocol is a precise set of rules defining how computers communicate. There are many different kinds of protocols defining different aspects of network communication.

There are different *layers* of communication on a network. Each layer represents a different level of abstraction between the physical hardware (wires) and the information being transmitted. In theory, each layer only talks to the layers immediately above and immediately below it.

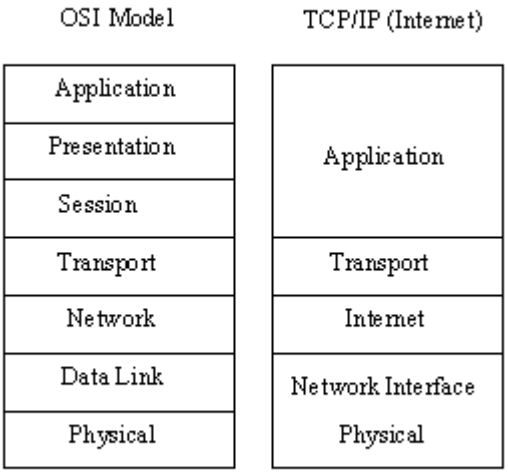
Of course, some crossover of functionality is to be expected, and several different approaches to the same division of layers for a network protocol were proposed. One that became adopted as a standard is the Open Systems Interconnection Reference Model. The OSI Reference Model (OSI-RM) uses seven layers, as shown in below



The TCP/IP architecture is similar but involves only five layers, because it combines some of the OSI functionality in two layers into one. The adoption of TCP/IP didn't conflict with the OSI standards because the two developed concurrently. In some ways, TCP/IP contributed to OSI, and vice-versa. Several important differences do exist, though, which arise from the basic requirements of TCP/IP, which are:

- A common set of applications
- Dynamic routing
- Connectionless protocols at the networking level
- Universal connectivity
- Packet-switching

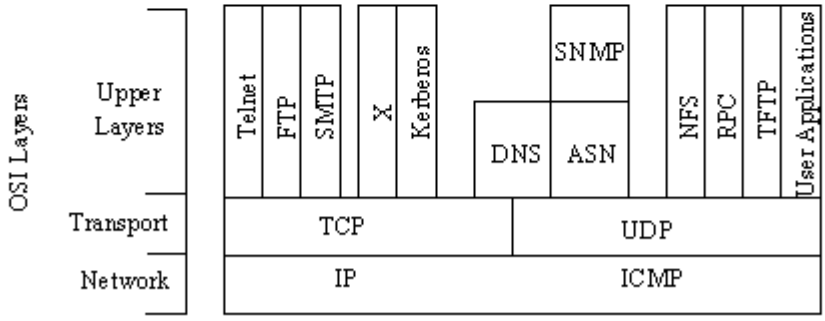
The differences between the OSI architecture and that of TCP/IP relate to the layers above the transport level and those at the network level. OSI has both the session layer and the presentation layer, whereas TCP/IP combines both into an application layer. The requirement for a connectionless protocol also required TCP/IP to combine OSI's physical layer and data link layer into a network level. A schematic view of TCP/IP's layered structure compared with OSI's seven-layer model is shown in below.



You can see that TCP/IP is not involved in the bottom two layers of the OSI model (data link and physical) but begins in the network layer, where the Internet Protocol (IP) resides. In the transport layer, the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are involved. Above this, the utilities and protocols that make up the rest of the TCP/IP suite are built using the TCP or UDP and IP layers for their communications system. The figure below shows the basic elements of some of the TCP/IP family of protocols.

Telnet - RemoteLogin
FTP - File Transfer Protocol
SMTP - Simple Mail Transfer Protocol
X - X Windows System
Kerberos - Security
DNS - Domain Name System
ASN - Abstract Syntax Notation
SNMP - Simple Network Management Protocol

NFS - Network File Server
RPC - Remote Procedure Calls
TFTP - Trivial File Transfer Protocol
TCP - Transmission Control Protocol
User Datagram Protocol
IP - Internet Protocol
ICMP - Internet Control Message Protocol



TCP/IP is dependent on the concept of clients and servers. This has nothing to do with a file server being accessed by a diskless workstation or PC. The term client/server has a simple meaning in TCP/IP: any device that initiates communications is the client, and the device that answers is the server. The server is responding to (serving) the client's requests.

A Quick Overview of TCP/IP Components

To understand the roles of the many components of the TCP/IP protocol family, it is useful to know what you can do over a TCP/IP network. Then, once the applications are understood, the protocols that make it possible are a little easier to comprehend. The following list is not exhaustive but mentions the primary user applications that TCP/IP provides.

Telnet

The Telnet program provides a remote login capability. This lets a user on one machine log onto another machine and act as though he or she were directly in front of the second machine. The connection can be anywhere on the local network or on another network anywhere in the world, as long as the user has permission to log onto the remote system. You can use Telnet when you need to perform actions on a machine across the country. This isn't often done except in a LAN or WAN context, but a few systems accessible through the Internet allow Telnet sessions while users play around with a new application or operating system.

File Transfer Protocol

File Transfer Protocol (FTP) enables a file on one system to be copied to another system. The user doesn't actually log in as a full user to the machine he or she wants to access, as with Telnet, but instead uses the FTP program to enable access. Again, the correct permissions are necessary to provide access to the files. Once the connection to a remote machine has been established, FTP enables you to copy one or more files to your machine. (The term transfer implies that the file is moved from one system to another but the original is not affected. Files are copied.) FTP is a widely used service on the Internet, as well as on many large LANs and WANs.

Simple Mail Transfer Protocol

Simple Mail Transfer Protocol (SMTP) is used for transferring electronic mail. SMTP is completely transparent to the user. Behind the scenes, SMTP connects to remote machines and transfers mail messages much like FTP transfers files. Users are almost never aware of SMTP working, and few system administrators have to bother with it. SMTP is a mostly trouble-free protocol and is in very wide use.

Kerberos

Kerberos is a widely supported security protocol. Kerberos uses a special application called an authentication server to validate passwords and encryption schemes. Kerberos is one of the more secure encryption systems used in communications and is quite common in UNIX.

Domain Name System

Domain Name System (DNS) enables a computer with a common name to be converted to a special network address. For example, a PC called Blackhole cannot be accessed by another machine on the same network (or any other connected network) unless some method of checking the local machine name and replacing the name with the machine's hardware address is available. DNS provides a conversion from the common local name to the unique physical address of the device's network connection.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) provides status messages and problem reports across a network to an administrator. SNMP uses User Datagram Protocol (UDP) as a transport mechanism. SNMP employs slightly different terms from TCP/IP, working with managers and agents instead of clients and servers (although they mean essentially the same thing). An agent provides information about a device, whereas a manager communicates across a network with agents.

Network File System

Network File System (NFS) is a set of protocols developed by Sun Microsystems to enable multiple machines to access each other's directories transparently. They accomplish this by using a distributed file system scheme. NFS systems are common in large corporate environments, especially those that use UNIX workstations.

Remote Procedure Call

The Remote Procedure Call (RPC) protocol is a set of functions that enable an application to communicate with another machine (the server). It provides for programming functions, return codes, and predefined variables to support distributed computing.

Trivial File Transfer Protocol

Trivial File Transfer Protocol (TFTP) is a very simple, unsophisticated file transfer protocol that lacks security. It uses UDP as a transport. TFTP performs the same task as FTP, but uses a different transport protocol.

Transmission Control Protocol

Transmission Control Protocol (the TCP part of TCP/IP) is a communications protocol that provides reliable transfer of data. It is responsible for assembling data passed from higher-layer applications into standard packets and ensuring that the data is transferred correctly.

User Datagram Protocol

User Datagram Protocol (UDP) is a connectionless-oriented protocol, meaning that it does not provide for the retransmission of datagrams (unlike TCP, which is connection-oriented). UDP is not very reliable (that is why it is often called Unreliable Datagram Protocol), but it does have specialized purposes. If the applications that use UDP have reliability checking built into them, the shortcomings of UDP are overcome.

Internet Protocol

Internet Protocol (IP) is responsible for moving the packets of data assembled by either TCP or UDP across networks. It uses a set of unique addresses for every device on the network to determine routing and destinations.

Internet Control Message Protocol

Internet Control Message Protocol (ICMP) is responsible for checking and generating messages on the status of devices on a network. It can be used to inform other devices of a failure in one particular machine. ICMP and IP usually work together.

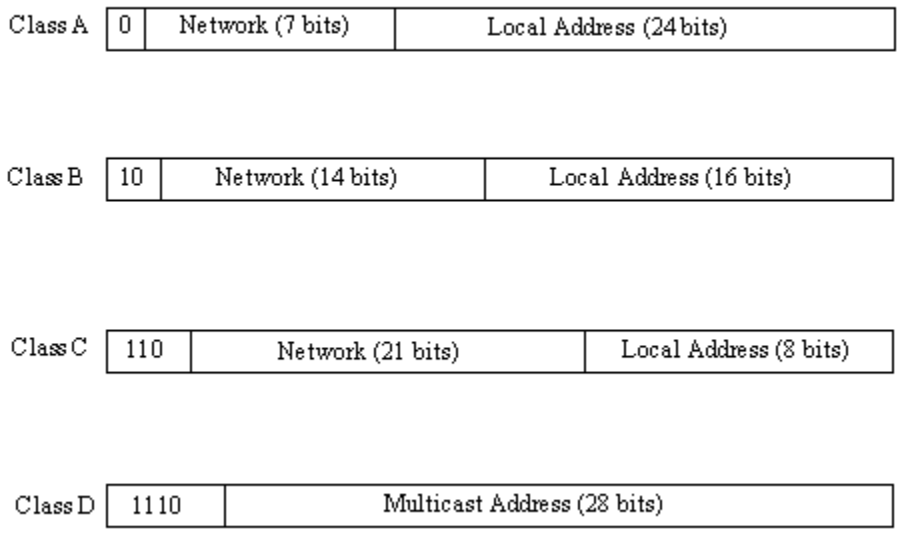
The Internet

The Internet is not a single network but a collection of networks that communicate with each other through gateways. A gateway (sometimes called a router) can be defined as a system that performs relay functions between networks. The different networks connected to each other through gateways are often called subnetworks, because they are a smaller part of the larger overall network. This does not imply that a subnetwork is small or dependent on the larger network. Subnetworks are complete networks, but they are connected through a gateway as a part of a larger internetwork, or in this case the Internet.

IP Addresses

TCP/IP uses a 4-byte address (or with IPv6, 16 bytes) to identify a machine on a network and the network to which it is attached. IP addresses identify a machine's connection to the network, not the machine itself—an important distinction. Whenever a machine's location on the network changes, the IP address must be changed too. The IP address is the set of numbers many people see on their workstations or terminals, such as 127.40.8.72, which uniquely identifies the device. IP (or Internet) addresses are assigned by special organizations (for example, the Network Information Center (NIC)), although if a network is not connected to the Internet, that network can determine its own numbering. For all Internet accesses, the IP address must be registered with the NIC.

There are four formats for the IP address, with each used depending on the size of the network. The four formats, called Class A through Class D, are shown in below. The class is identified by the first few bit sequences, shown in the figure as one bit for Class A and up to four bits for Class D. The class can be determined from the first three (high-order) bits. In fact, in most cases, the first two bits are enough, because there are few Class D networks.



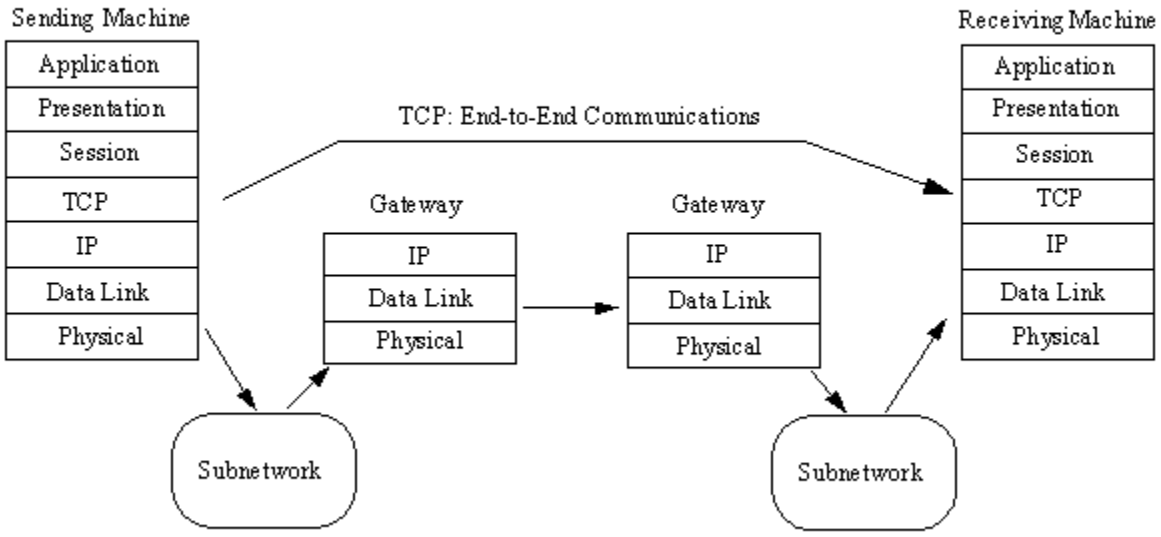
The Transmission Control Protocol provides a considerable number of services to the IP layer and the upper layers. Most importantly, it provides a connection-oriented protocol to the upper layers that enable an application to be sure that a data packet (datagram) sent out over the network was received in its entirety. In this role, TCP acts as a message-validation protocol providing reliable communications. If a data packet is corrupted or lost, TCP usually handles the retransmission, rather than the applications in the higher layers.

TCP manages the flow of datagrams from the higher layers to the IP layer, as well as incoming datagrams from the IP layer up to the higher level protocols. TCP has to ensure that priorities and security are properly respected. TCP must be capable of handling the termination of an application above it that was expecting incoming datagrams, as well as failures in the lower layers. TCP also must maintain a state table of all data streams in and out of the TCP layer. The isolation of all these services in a separate layer enables applications to be designed without regard to flow control or message reliability. Without the TCP layer, each application would have to implement the services themselves, which is a waste of resources.

Because TCP is a connection-oriented protocol responsible for ensuring the transfer of a datagram from the source to destination machine (end-to-end communications), TCP must receive communications messages from the destination machine to acknowledge receipt of the datagram. The term virtual circuit is usually used to refer to the communications between the two end machines, most of which are simple acknowledgment messages (either confirmation of receipt or a failure code) and datagram sequence numbers.

Ports and Sockets

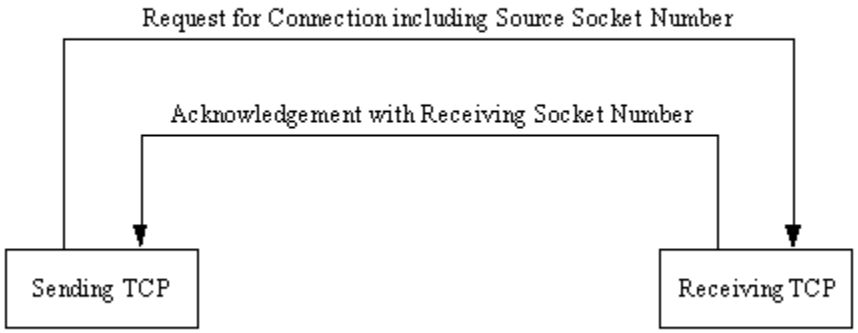
All upper-layer applications that use TCP (or UDP) have a port number that identifies the application. A port number is a 16-bit unsigned integer, thus ranging from 0 to 65535. In theory, port numbers can be assigned on individual machines, or however the administrator desires, but some conventions have been adopted to enable better communications between TCP implementations. This enables the port number to identify the type of service that one TCP system is requesting from another. Port numbers can be changed, although this can cause difficulties. Most systems maintain a file of port numbers and their corresponding service. Typically, port numbers above 1025 are reserved for private use of the local machine, but numbers below 1025 are used for frequently used processes. A list of frequently used port numbers is published by the



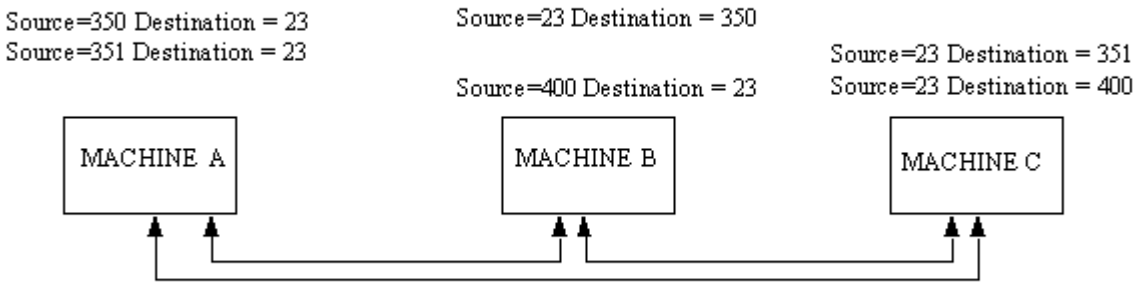
Internet Assigned Numbers Authority. The commonly used port numbers on this list are shown below.

Protocol	Port	Encoding
echo	7	TCP/UDP
daytime	13	TCP/UDP
ftp-data	20	TCP
ftp	21	TCP
telnet	23	TCP
smtp	25	TCP
time	37	TCP/UDP
http	80	TCP

Each communication circuit into and out of the TCP layer is uniquely identified by a combination of two numbers, which together are called a socket. The socket is composed of the IP address of the machine and the port number used by the TCP software. Both the sending and receiving machines have sockets. Because the IP address is unique across the internetwork, and the port numbers are unique to the individual machine, the socket numbers are also unique across the entire internetwork. This enables a process to talk to another process across the network, based entirely on the socket number. This process is shown below. If the sending TCP wants to establish a Telnet session from its port number 350, the socket number would be composed of the source machine's IP address and the port number (350), and the message would have a destination port number of 23 (Telnet's port number). The receiving TCP has a source port of 23 (Telnet) and a destination port of 350 (the sending machine's port).



The sending and receiving machines maintain a port table, which lists all active port numbers. The two machines involved have reversed entries for each session between the two. This is called binding. The source and destination numbers are simply reversed for each connection in the port table. Of course, the IP addresses, and hence the socket numbers, are different. If the sending machine is requesting more than one connection, the source port numbers are different, even though the destination port numbers might be the same. For example, if the sending machine were trying to establish three Telnet sessions simultaneously, the source machine port numbers might be 350, 351, and 352, and the destination port numbers would all be 23.



References

http://en.wikipedia.org/wiki/Computer_network
What is a better place to go?

Code Example

There are not any code examples associated with this hybrid activity.

Exercise

This is an exercise in reading and understanding the basics.

Questions

Q1. How many bytes is the TCP/IP address with IPv6?
Q2. How many port numbers are available on a machine?
Q3. Is TCP a connection-oriented protocol?

Submission

No submission is required for this activity.

Marks

No marks are allocated for this activity, but remember that understanding networking is essential in the modern computing.

And do not forget that:

"*Without networking the world will be a very lonely place*" found on Facebook