CHAPTER

1

ALL ABOUT NETWORKS

We shall begin this chapter with an understanding of networking concepts and proceed further to know why languages, like C/C++, are not opted for a network and why Java is a language suited for a network, especially Internet.

Interconnection of computers is called *network*. A network may contain two or more computers connected with or without a cable. For example, you can connect a computer in Delhi to one placed in Hyderabad with the help of a satellite and data can be sent or received via satellite in the form of electromagnetic waves. So, a network may or may not have a cable.

The primary purpose of a network is *resource sharing,* which means a computer can share its resources, like data, memory, software, hardware, CPU, etc. with other computers on the network. In the network, one computer may ask the other computer for some service. The computer, which requests for some service is called *client* and the other computer, which provides that service to the client is called *server*. A network can have several clients and servers and such a network architecture is also called a *client/ server architecture*.

Figure 1.1 shows a client /server architecture. Here, you can see that the request goes from the client to the server and the server gives back the response.

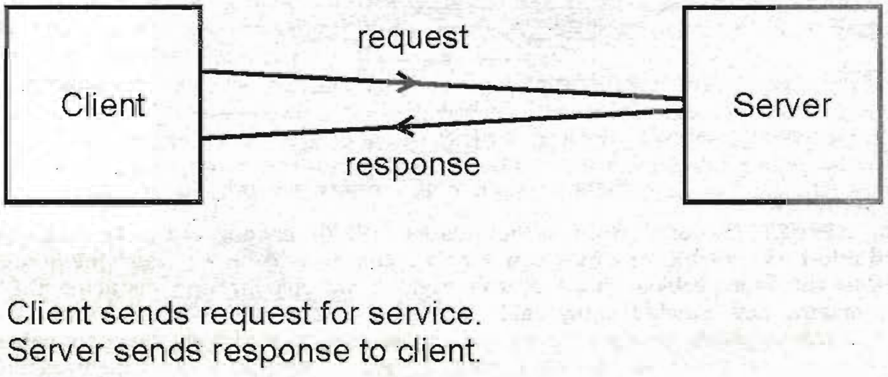
request
Client
Server
response
Client sends request for service.
Server sends response to client.

Figure 1.1 A network with two computers

Important Interview Question

What is the advantage of a network?

The main advantage of a network is that it makes resource sharing possible among the connected systems, thus helping in better utilization of resources.

Now, let us see how we can connect one computer to another. Following are the requirements for setting up a network of computers:

- **Hardware**: Computer systems, Network Interface Cards (NIC), cables, modems, hubs, satellites, etc.
- **Software**: Any software, which supports networking features, like Unix, Linux, Windows NT, etc. so that data can be sent or received.
- **Protocol**: *Protocol* could be defined as a standard procedure for regulating data transmission between computers. It specifies a way to identify the destination and source computers and establish a connection between them to receive or send data. For example, TCP/IP or UDP protocols help to divide data into small packets of several bytes, insert the packets into frames, and then send those frames to the destination computer over the network. Protocol is the underlying feature of the software used on the network.

What Comprises the Internet?

Internet could be defined as the network of all the computers existing on the earth. Not only individual computers but also a network of computers can be connected into the Internet. For example, a company may have 100 computers, all connected by the cable. Now, this local network can be directly connected to the Internet, so that it becomes a part of the Internet. So, Internet is also sometimes defined as a global network of all networks.

Going back to the history of Internet in 1969, DoD (Department of Defense), which is a US military defense department, got 4 computers which they connected with a cable. These 4 computers were connected by a cable and Unix was installed on them. This is the first network on the earth, which is also called ARPANET (Advanced Research Project Agency Network). Figure 1.2 shows how computers were connected in an ARPANET.

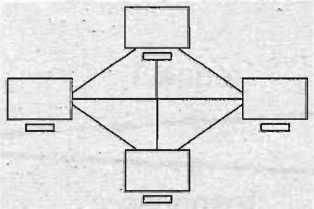

Figure 1.2 ARPANET, the first network

Later on, NSFNET (National Science Foundation (NSF)) encouraged individual users to get connected into this network. Any user can simply connect his computer into this network by using a cable. But the first problem faced by the users is purchasing and installing the cables. For example, imagine how much lengthy cable is needed if a person in India wants to connect his computer to this network. So they thought of an alternative. If already available cables are used, this problem will be resolved. By that time, telephone networks are widely available. Hence, telephone companies like AT&T, Hyundai, British Telecom, etc. came forward and offered their networks for use. Now, the problem of connecting personal computers with a network is solved. People who want to connect to the network can simply sit in their homes and connect their computers to the telephone jack. Thus, more and more computers have started becoming a part of this network and the network has grown rapidly. Today, this network has trillions of computers connected with each other. Thus, it became the biggest network on the earth. This network is called Internet.

A network of computers within an organization is called *Intranet*. An external network connecting an organization's network with that of another organization is called *Extranet*. In other words, we can take Extranet as a network outside an organization connecting it to other organizations' network. Similarly, an organization may use the services of a third party private network to communicate with outside networks. This third party private network is called *VAN* (Value Added Network). Now, let us try to understand the concept of Intranet and Extranet with the help of a figure. In Figure 1.3, you can notice that the network within an organization is called Intranet and that connecting the networks of two organizations is known as an Extranet.

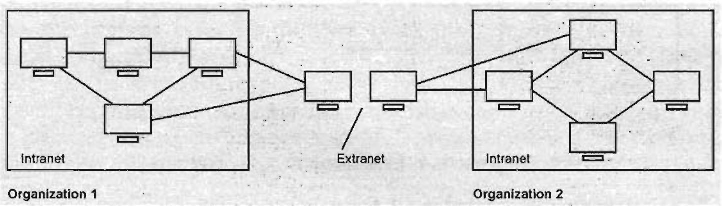

Figure 1.3 Intranet and Extranet

How does Internet Function?

Like any network, Internet also requires certain hardware, software, and protocols to perform its functions. For example, hardware like computer systems, cables, hubs, routers, satellites, etc. are needed to connect different computers with each other. The computer, which is connected to Internet Service Provider's (ISP) computer is called *client* and the ISP's computer is called *server*. Other than ISP servers, many other servers are maintained and used privately by some organizations. Currently, millions of servers are available on Internet. Some servers focus on a single area like maintaining employees' attendance in a company or scheduling the jobs for the employees of different branches. Such servers are called *Application servers*. Some servers provide services purely related to Internet, for example, maintaining the Websites and communicating with clients on Internet. Such servers are called *Web servers*.

A client machine and a server machine on Internet also need software. The software that is installed on an Internet client is called *Web browser*. Popular Web browsers are Internet Explorer, Netscape Navigator, Mozilla Firefox, Mosaic, etc. The software that is installed on an Internet server machine is called *Web server*. There are several Web servers, like IIS (Internet Information Server), Web logic, Web dynamics, Apache, JBoss, etc.

Apart from software, protocols are also used on Internet. The most widely used protocol on Internet is `http`, which stands for hypertext transfer protocol. `http` is responsible for displaying the html pages (or Web pages) on your client machine when you browse Internet. When you download a file from Internet to your client machine, you are using `ftp` (file transfer protocol). When you send mails, SMTP (Simple Mail Transfer Protocol) is used, and while receiving mails POP (Post Office Protocol) is used.

Important Interview Question

Which is the most widely used protocol on Internet?

Hypertext transfer protocol (http) is the most widely used protocol on Internet. The text on Internet is sent or received from one machine to another using this protocol only.

Software Development for Internet

Suppose we are writing a C program with the name `x.c`. This means `x.c` file contains source code. When we compile this program, we get `x.obj` file that contains machine language code, which is just equal to the source code in `x.c` file. For example, this `x.obj` file contains the first statement to include a header file, like `<stdio.h>`. When we run the program, C compiler goes to the C standard library (it is found generally in `tc\lib`) and searches for the header file `<stdio.h>` there. When the header file is found, C compiler copies the entire code (this code will be in machine language format) from the header file into the C program. So, if you write a C program of 10 lines and the `<stdio.h>` file contains 200 lines, then the total size of the file will be 210 lines. This file is also called `x.exe` file. This file is a full-fledged file that contains the entire program in machine language instructions. These instructions are understandable to the microprocessor. So it executes them and gives the results. Figure 1.4 shows how the source code in a C program is converted to machine language instructions before being executed by the compiler.

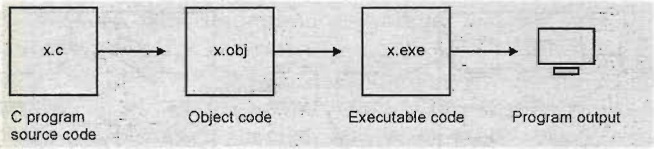

Figure 1.4 Execution of a C program

There are several microprocessors developed by many companies, for example, Intel Corporation. Every microprocessor can recognize a group of instructions called the *Instruction set* of that microprocessor. So the instructions recognized by Pentium processor may not be understandable to the Celeron and similarly the instructions of Celeron may not be understandable by Spark processor.

If we generate `x.exe` file on a computer with Pentium processor, then that `x.exe` file contains machine language instructions understandable to Pentium only. If we try to execute this `x.exe` file on another computer with Spark processor, it may not be able to execute the instructions of `x.exe` file. The reason is Spark processor cannot understand the Pentium processor instructions, which are in `x.exe`.

Similarly, develop `x.exe` file using an operating system like DOS. Then copy this `x.exe` into other operating system like UNIX. Now, try to execute the `x.exe` file on UNIX. You will see that you cannot! The reason for this is that every operating system stores the data and instructions in different formats. For example, the instruction for addition of two values may be stored by DOS as: `add a,b`; whereas the same instruction may be stored by UNIX as: `a add b`. So, if we try to execute `add a, b` instruction in UNIX, it cannot understand it since its format is different from what it is expecting. This is the reason, why we cannot execute the same `x.exe` file in different operating systems.

Note

There is a lot about instruction formats. The example that we have taken is only for illustration purpose.

What does it mean? If we created an `.exe` file in C or C++ on X processor and Y operating system, then we can execute it on X processor and Y operating system only. If the processor is changed or the operating system is changed, then it is not possible to execute the same code. This means C or C++ programs are executable only on the computer system that has X processor and Y operating system, where they have been developed. They cannot be executed on a different configuration. So they are called system dependent programs. Hence, C or C++ languages are also called system dependent or platform dependent languages.

Now, let us come back to Internet. As we know that Internet is a global network of all the computers existing on the earth and anyone can connect his/her computer to this network. So there will be different types of computers with different processors and different operating systems existing on Internet. It is completely heterogeneous. So, if a software is developed using C/C++ like languages, that software can be distributed on the Internet and people can download the `.exe` files, but they cannot run those files because of their system dependency. So C/C++ is not suitable for developing software for Internet. We need a language, which is completely system independent so that it can run on any system on Internet. And that is Java!

How Java is suitable for Internet? In Java, if we write a program, it is stored with an extension `.java`. Let us create a program, say `x.java`. It contains java source code statements. Now, the first step is to compile this `x.java` program. To do that, Java compiler first translates the `x.java` program into `x.class` file. What will be there in `x.class` file? Java people coined a group of instructions to express any operation. These instructions are called *byte code* instructions because the size of each instruction is 1 byte (=8 bits) exactly. There are a fixed number of byte code instructions, around 200. Any java program can be rewritten using these few byte -code instructions. That file is called `x.class`. So, we can say that a `x.class` file contains source code equivalent to byte code.

But the microprocessor cannot understand byte code instructions, neither can it execute them. Then, these byte code instructions are meant for whom? They are for JVM (Java Virtual Machine) . These byte code instructions are understandable by JVM, which is a program written to understand the byte code instructions and convert them into machine code. JVM's role is very crucial in a way that it has to first identify the operating system and processor used in the computer system and then convert the byte code instructions in an understandable format for that particular processor and operating system.

Remember, JVM is not a machine—it is a program! JVM program is freely available on Internet and can be downloaded for use in a few minutes. Figure 1.5 shows how a java source code passes through various stages before being executed by the compiler.

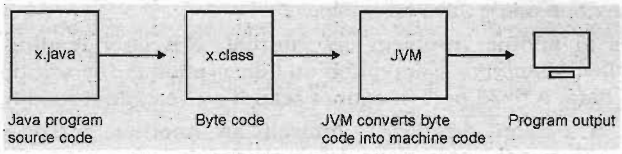

Figure 1.5 Execution of a Java program

So, if we write `x.java` program on a computer using Pentium processor and compile it, the compiler will produce `x:class` containing byte code instructions.

These byte code instructions are given to JVM. Now JVM understands that we are using Pentium processor in our system, so it converts byte code into machine code, which is understandable by the Pentium processor. So the processor executes it and displays the results. Similarly, JVM understands the operating system used in the computer and accordingly it forms the instructions, so there will not be any problem for the operating system to understand these instructions. Now, if we send the same `x.class` file to another computer system with a different processor, say Spark processor, then the JVM in that system will convert these byte code instructions to machine code understandable by the Spark processor. So Spark processor can execute those instructions and display the results. In this way, we can execute the `.class` file on any computer system with any processor and any operating system—provided JVM is available.

Remember byte code is system independent. It can be created on any computer using those predefined 200 instructions only. But JVM is system dependent, since it has to interact with the processor and operating system of the computer. This is the reason behind several flavors of JVM available on Internet. If we are using Windows, then we should download only Windows version of JVM and if we are using Linux, then we should use only Linux version of JVM.

So, if we write a `.java` program on X processor and Y operating system, then that program is executable on any other computer system with any processor and operating system. This means java programs are system independent or platform independent. That's why Java is called a system independent programming language.

On Internet, different computer systems with different processors and different operating systems are available, but still Java programs will run on any system and produce the same results on all the systems. If we develop a software using Java, the `.class` files can be downloaded from Internet and can be executed on any other computer system without any problem. Hence, Java is a suitable programming language for software development on Internet.

Important Interview Question

What is the difference between an executable file and a .class file?

.exe file contains machine language instructions for the microprocessor and is system dependent. .class file contains byte code instructions for the JVM and is system independent.

Since Internet is a public network, everyone can connect their computers into Internet and this leads to several security problems for data. Suppose, we are sending an email to a friend. Now, this mail will go through millions of servers on Internet before reaching the destination computer. In between, anybody can open the mail and see the data of the mail. Thus, we can say that data sent over Internet is susceptible to many kinds of security threats. Let us discuss some of the major security problems for data on Internet along with their solutions:

- **Eavesdropping**: This means reading others' data illegally on Internet. For example, opening others' emails and reading them comes into this category. The solution for this problem is: encryption and decryption. Converting the data into an unreadable format is called encryption and getting back the original data from the encrypted format is called decryption. When sending the email, we can encrypt the data and send it. If any third person opens it, he cannot read it because it is in an unreadable format. While encrypting the data, we use a password called key. Only the person who knows that key can decrypt it and read it. This key is known to our friend to whom the email is sent, so that only he can open it and read it. Encryption and decryption can be done using Java technology.
- **Tampering**: This is another problem on Internet. Not only reading others' data but also modifying it is called *tampering*. Encryption and decryption is the solution for this problem also. If we encrypt the data, a third person cannot read it so he cannot modify it also.
- **Impersonation:** A person disguising himself as another person on Internet is called *impersonation*. Many people hide their original identity and act as somebody else to make transactions. The solution for this problem is *digital signature*. Digital signature is a file that contains personal identification information in an encrypted format. Storing the personal information in digital signature and sending this information along with the email assures the receiver of the identity of the sender. Digital signature can be created using Java technology.
- **Virus**: Virus represents a program that can cause harm to the data, software, and hardware of a computer system. Most of the viruses spread with `.exe`, `.doc`, image (`.gif`, `.jpg`), audio, and video files(`.mpg`, `.avi`, `.mp3`). Virus cannot spread with a `.txt` file (example, the file created in Notepad). In Java, we create `.class` files, which are similar to `.txt` files and hence there is no chance of virus affecting these files. Even if somebody intentionally tries to incorporate the virus code into a `.class` file, JVM can verify its presence before running the program and, if found, can abort the `.class` file's execution.

From the above discussion, we can say that Java is the language, which can eliminate the above security problems that often occur on Internet. This is another reason why Java is sought after for developing software meant for Internet.

Important Interview Question

Why Java is suitable for Internet?

Java is suitable for Internet because of two main reasons. 1) It is system independent and hence its programs can run on any type of computer system available on Internet. 2) It eliminates a lot of security problems for data on Internet.

Conclusion

By now, you must have gathered information about networks, the Internet, and how Java is suitable for the Internet.