CHAPTER 1:

INTRODUCTION

The old model of a single computer serving all of the organization’s computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called **computer networks**.

The **‘‘computer network’’** means a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information. The connection need not be via a copper wire; fiber optics, microwaves, infrared, and communication satellites can also be used. Networks come in many sizes,

Shapes and forms, as we will see later. They are usually connected together to make larger networks, with the **Internet** being the most well-known example of a network of networks.

There is considerable confusion in the literature between a computer network and a **distributed system**. The key distinction is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called **middleware**, is responsible for implementing this model. A well-known example of a distributed system is the **World Wide Web**. It runs on top of the Internet and presents a model in which everything looks like a document (Web page).

In a computer network, this coherence, model, and software are absent. Users are exposed to the actual machines, without any attempt by the system to make the machines look and act in a coherent way.

(Note: Its better to get more and more info on Distributed System once we complete with Computer Networks)

* 1. USES OF COMPUTER NETWORKS

1.1.1 Business / Enterprise Applications

* Resource Sharing (i.e. Physical)
* Sharing Information between two different departments of same organization using networks like **Virtual Private Networks (VPN)**
* Forming **Client-Server model** for **Web Applications**
* As **communication-medium** among employees e.g. Email communication, IP telephony or Voice over IP (VoIP).
* Also **Desktop Sharing** gives adding video to audio.
* For e-commerce

1.1.2 Home Applications

Metcalfe’s Law:

Bob Metcalfe, the inventor of Ethernet, hypothesized that the value of a network is proportional to the square of the number of users because this is roughly the number of different connections that may be made

* Connectivity to remote computers using Internet to get information
* Forming **Peer-to-peer communication model** to share information beside Client-Server model.

Note: How to make git repo as peer-to-peer model by providing list of nearby people.

In this form, individuals who form a loose group can communicate with others in the group. Every person can, in principle, communicate with one or more other people; there is no fixed division into clients and servers

Many peer-to-peer systems, such BitTorrent, do not have any

central database of content. Instead, each user maintains his own database locally and provides a list of other nearby people who are members of the system. A new user can then go to any existing member to see what he has and get the names of other members to inspect for more content and more names. This lookup process can be repeated indefinitely to build up a large local database of what is out there.

Legal applications for peer-to-peer communication also exist. These include fans sharing public domain music, families sharing photos and movies, and users downloading public software packages. In fact, one of the most popular Internet applications of all, email, is inherently peer-to-peer. This form of communication is likely to grow considerably in the future.

* Instant Messaging (This derived from the UNIX ‘talk’ program).
* Multi-person messaging services like Twitter
* **Person-to person communications** and accessing information are **Social Network** applications e.g. wiki, Facebook and Wikipedia
* **Electronic Flea Markets (e-flea)**  (Follows per-to-peer model)
* **IPTV**
* Multi-person real time simulation games (i.e. virtual world gaming)
* **Ubiquitous computing**
* Forming network using RFID

1.1.3 Mobile Users

* Network brings **the Connectivity** among the people or mobile computer devices from anywhere on their go using different network technologies like Cellular and wireless Hotspot i.e. WiFi (802.11).
* Text messaging or Texting using cellular network
* Usage of Smart Phones through Cellular network
* Consumer Electronic devices can also use Cellular and HotSpot networks to stay connected to remote computers
* Mobile maps and navigation using GPS. Also for “geo-tagging” is possible by GPS
* Using mobiles for m-commerce. (Using NFC technology)
* Implementing **Sensor Networks** made up of nodes.
* Wireless Parking meters
* As **Wearable Computers** like Smart watches…

1.1.4 Social Issues

Network neutrality:

This argument for communications that are not differentiated by their content or source or who is providing the content is known as **network neutrality** (Wu, 2003). It is probably safe to say that this debate will go on for a while

Profiling Users:

The private sector does its bit too by profiling users. For example,

small files called cookies that Web browsers store on users’ computers allow

companies to track users’ activities in cyberspace and may also allow credit card

numbers, social security numbers, and other confidential information to leak all

over the Internet (Berghel, 2001). Companies that provide Web-based services

may maintain large amounts of personal information about their users that allows

them to study user activities directly. For example, Google can read your email

and show you advertisements based on your interests if you use its email service,

Gmail.

Note: what is botnet?

* 1. NETWORK HARDWARE

In Network design technology two dimensions are stand out as important: **Transmission Technology and Scale**.

There are 2 types of transmission technology that are **Broadcast** Link and **Point-to-Point** links.

**Point-to-point links** connect individual pairs of machines. To go from the

Source to the destination on a network made up of **point-to-point links**, short messages, called **packets** in certain contexts, may have to first visit one or more intermediate machines. Often multiple routes, of different lengths, are possible, so finding good ones is important in point-to-point networks. Point-to-point transmission with exactly one sender and exactly one receiver is sometimes called **unicasting.**

In contrast, on a **broadcast network**, the **communication channel** is shared by all the machines on the network; packets sent by any machine are received by all the others. An address field within each packet specifies the intended recipient. Upon receiving a packet, a machine checks the address field. If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored. Some **broadcast systems also support transmission to a subset of the machines, which known as multicasting.**

An alternative criterion for classifying networks is by **scale**. Distance is important as a classification metric because different technologies are used at different scales.

Based on scale we have following classifications…

* Personal Area Network (PAN)
* Local Area Network (LAN)
* Metropolitan Area Network (MAN)
* Wide Area Network (WAN)
* Internetwork
  + 1. PAN

Technologies involved in PAN are Blue Tooth, RFID and NFC….

* + 1. LAN

When LAN are used by companies, they are called **Enterprise Networks.**

In LAN we are using device, called an AP (Access Point), wireless router, or base station, relays packets between the wireless computers and also between them and the Internet. Being the AP is like being the popular kid as school because everyone wants to talk to you. However, if other computers are close enough, they can communicate directly with one another in a **peer-to-peer** configuration.

There is a standard for wireless LANs called IEEE 802.11, popularly known

as WiFi. The topology of many wired LANs is built from point-to-point links. IEEE

802.3, popularly called Ethernet, is, by far, the most common type of wired LAN.

It is the job of the protocol to sort out what paths packets should travel to safely reach the intended computer. It is also possible to divide one large physical LAN into two smaller logical LANs i.e. **Virtual LAN or VLAN**.

Both wireless and wired broadcast networks can be divided into static and dynamic designs, depending on how the channel is allocated.

* + 1. MAN

Here, both television signals and Internet being fed into the centralized cable headend for subsequent distribution to people’s homes.

Cable television is not the only MAN, though. Recent developments in highspeed

wireless Internet access have resulted in another MAN, which has been

standardized as IEEE 802.16 and is popularly known as **WiMAX**.

* + 1. WAN

Communication Subnet or Subnet:

It is a network that connects hosts (a computer which runs user program / app). The job of subnet is to carry messages from host to host.

In most WANs, the subnet consists of two distinct components: transmission

lines and switching elements. Transmission lines move bits between machines.

They can be made of copper wire, optical fiber, or even radio links. Most companies

do not have transmission lines lying about, so instead they lease the lines

from a telecommunications company. Switching elements, or just switches, are

specialized computers that connect two or more transmission lines. When data

arrive on an incoming line, the switching element must choose an outgoing line on

which to forward them. These switching computers have been called by various

names in the past; the name router is now most commonly used. The routers will usually connect different kinds of networking technology.

Rather than lease dedicated transmission lines, a company might connect its offices to the Internet This allows connections to be made between the offices as virtual links that use the underlying capacity of the Internet. This arrangement is called a VPN (Virtual Private Network).

The second variation is that the subnet may be run by a different company. The subnet operator is known as **a network service provider** and the offices are its customers. The subnet operator will connect to other customers too, as long as they can pay and it can provide service. Since it would be a disappointing network service if the customers could only send packets to each other, the subnet operator will also connect to other networks that are part of the Internet. Such a subnet operator is called an **ISP (Internet Service Provider)** and the subnet is an **ISP network**. Its customers who connect to the ISP receive Internet service.

How the network makes the decision as to which path to use is called the **routing algorithm**. Many such algorithms exist. How each router makes the decision as to where to send a packet next is called the **forwarding algorithm**.

The cellular telephone network is another example of a WAN that uses wireless technology. This system has already gone through three generations and a fourth one is on the horizon. The first generation was analog and for voice only. The second generation was digital and for voice only. The third generation is digital and is for both voice and data. Each cellular base station covers a distance much larger than a wireless LAN, with a range measured in kilometers rather than tens of meters. The base stations are connected to each other by a backbone network that is usually wired. The data rates of cellular networks are often on the order of 1 Mbps, much smaller than a wireless LAN that can range up to on the order of 100 Mbps.

* + 1. INTERNETWORKS

A collection of interconnected networks is called an internetwork or internet. Subnets, networks, and internetworks are often confused. The term ‘‘subnet’’ makes the most sense in the context of a wide area network, where it refers to the collection of routers and communication lines owned by the network operator.

A network is formed by the combination of a subnet and its hosts. However,

the word ‘‘network’’ is often used in a loose sense as well. A subnet might be described

as a network, as in the case of the ‘‘ISP network’’.

Let us say more about **what constitutes an internetwork**. We know that an internet is formed when distinct networks are interconnected. In our view, connecting a LAN and a WAN or connecting two LANs is the usual way to form an internetwork, but there is little agreement in the industry over terminology in this area. There are two rules of thumb that are useful. First, if different organizations have paid to construct different parts of the network and each maintains its part, we have an internetwork rather than a single network. Second, if the underlying technology is different in different parts (e.g., broadcast versus point-to-point and wired versus wireless), we probably have an internetwork.

To go deeper, we need to talk about how two different networks can be connected.

The general name for a machine that makes a connection between two or

more networks and provides the necessary translation, both in terms of hardware

and software, is a **gateway**. **Gateways are distinguished by the layer at which**

**they operate in the protocol hierarchy.** We will have much more to say about layers

and protocol hierarchies starting in the next section, but for now imagine that

higher layers are more tied to applications, such as the Web, and lower layers are

more tied to transmission links, such as Ethernet. The level in the middle that is ‘**‘just right’’** is often called the network layer, and a router is a gateway that switches packets at the network layer. We can now spot an internet by finding a network that has routers.

* 1. NETWORK SOFTWARE
     1. Protocol Hierarchies

To reduce their design complexity, most networks are organized as a stack of

**layers** or **levels**, each one built upon the one below it. The number of layers, the

name of each layer, the contents of each layer, and the function of each layer differ

from network to network. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

When layer *n* on one machine carries on a conversation with layer *n* on another

machine, the rules and conventions used in this conversation are collectively known as the layer *n* protocol. Basically, a **protocol** is an agreement between the communicating parties on how communication is to proceed.

The entities comprising the corresponding layers on different machines are called **peers**. The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.

In reality, no data are directly transferred from layer *n* on one machine to layer *n* on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the **physical medium** through which actual communication occurs.

Between each pair of adjacent layers is an **interface**. The interface defines which primitive operations and services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers.

A set of layers and protocols is called a **network architecture**. A list of the protocols used by a certain system, one protocol per layer, is called a **protocol stack**.

The lower layers of a protocol hierarchy are frequently implemented in hardware or firmware.

* + 1. Design Issues for the Layers

1. **Reliability** is the design issue of making a network that operates correctly even though it is made up of a collection of components that are themselves unreliable. Think about the bits of a packet traveling through the network. There is a chance that some of these bits will be received damaged (inverted) due to fluke electrical noise, random wireless signals, hardware flaws, software bugs and so on. How is it possible that we find and fix these errors?
2. Error Detection

One mechanism for finding errors in received information uses codes for **error detection**. Information that is incorrectly received can then be retransmitted until it is received correctly. They are used at low layers, to protect packets sent over individual links, and high layers, to check that the right contents were received.

1. Routing

Another reliability issue is finding a working path through a network. Often

there are multiple paths between a source and destination, and in a large network, there may be some links or routers that are broken. Suppose that the network is down in Germany. Packets sent from London to Rome via Germany will not get through, but we could instead send packets from London to Rome via Paris. The network should automatically make this decision. This topic is called routing.

1. **Evolution of Networks:** Over time, networks grow larger and new designs emerge that need to be connected to the existing network.
2. Protocol layering

Since there are many computers on the network, every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message. This mechanism is called **addressing** or **naming**, in the low and high layers,respectively.

1. Internetworking

The differences in the maximum size of a message that the networks can transmit. This leads to mechanisms for disassembling, transmitting, and then

reassembling messages.

1. Scalable
2. Resource Allocation

Networks provide a service to hosts from their underlying resources, such as the capacity of transmission lines. To do this well, they need mechanisms that divide their resources so that one host does not interfere with another too much. Many designs share network bandwidth dynamically, according to the shortterm needs of hosts, rather than by giving each host a fixed fraction of the bandwidth that it may or may not use. This design is called **statistical multiplexing**, meaning sharing based on the statistics of demand. It can be applied at low layers for a single link, or at high layers for a network or even applications that use the network.

An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Feedback from the receiver to the sender is often used. This subject is called **flow control**.

Sometimes the problem is that the network is oversubscribed because too many computers want to send too much traffic, and the network cannot deliver it all. This overloading of the network is called **congestion**.

Most networks must provide service to applications that want this **real-time** delivery at the same time that they provide service to applications that want high throughput. **Quality of service** is the name given to mechanisms that reconcile these competing demands.

1. The last major design issue is to secure the network by defending it against different kinds of threats.
   1. Confidentiality
   2. Authentication
   3. Integrity
      1. Connection-Oriented v/s Connectionless Service

Layers can offer two different types of service to the layers above them: connection-oriented and connectionless.

**Connection-Oriented**

To use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a **tube**: the sender pushes objects (bits) in at one end, and the receiver takes them out at the other end. In most cases the order is preserved so that the bits arrive in the order they were sent. In some cases when a connection is established, the sender, receiver, and subnet conduct a **negotiation** about the parameters to be used, such as maximum message size, quality of service required, and other issues. Typically, one side makes a proposal and the other side can accept it, reject it, or make a counterproposal. A **circuit** is another name for a connection with associated resources, such as a fixed bandwidth.

**Connectionless**

Each message (letter) carries the full destination address, and each one is routed through the intermediate nodes inside the system independent of all the subsequent messages. There are different names for messages in different contexts; a **packet** is a message at the network layer. When the intermediate nodes receive a message in full before sending it on to the next node, this is called **store-and-forward switching**. The alternative, in which the onward transmission of a message at a node starts before it is completely received by the node, is called **cut-through switching**. Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first.

Each kind of service can further be characterized by its reliability. Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable. E.g. file transfer

Reliable connection-oriented service has two minor variations: **message sequences (Message Boundaries) and byte streams** **(No Message Boundries).**

Unreliable (meaning not acknowledged) connectionless service is often called **datagram** service,

**Connection-Oriented Service Connectionless Service**

**Reliable Unreliable Reliable Unreliable**

Message Sequence Digitized Voice traffic Ackldgd Datagram Datagram

e.g. Sequence of pages e.g. VoIP e.g text msg e.g.Junk mail

Byte Stream Request-Reply

e.g. Movie download e.g. DB Query

1.3.4 Service Primitives

A service is formally specified by a set of **primitives** (operations) available to user processes to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operatThe set of primitives available depends on the nature of the service being provided.

The primitives for connection-oriented service are different from those of connectionless service. As a minimal example of the service primitives that might provide a reliable byte stream, consider the primitives listed in Fig. 1-17. They will be familiar to fans of the Berkeley socket interface, as the primitives are a simplified version of that interface.

Primitive Meaning

LISTEN Block Waiting for an incoming connection

CONNECT Establish a connection with a waiting peer

ACCEPT Accept an incoming connection from a peer

RECEIVE Block waiting for incoming message

SEND Send message to a peer

DISCONNECT Terminate the connection

These primitives might be used for a request-reply interaction in a client-server environment. To illustrate how, we sketch a simple protocol that implements the **service using acknowledged datagrams.**

For more detail on this read Computer Networks section.

* + 1. The Relationship of Services to Protocols

A ***service***is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A ***protocol***, in contrast, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled. This is a key concept that any network designer should understand well. To repeat this crucial point, services relate to the interfaces between layers, as illustrated in Fig. 1-19. In contrast, protocols relate to the packets sent between peer entities on different machines. It is very important not to confuse the two concepts.

An analogy with programming languages is worth making. A service is like

an abstract data type or an object in an object-oriented language. It defines operations that can be performed on an object but does not specify how these operations are implemented. In contrast, a protocol relates to the *implementation* of the service and as such is not visible to the user of the service.

* 1. REFERENCE MODELS

We will discuss two important network architectures: the OSI reference model and the TCP/IP reference model. Although the *protocols* associated with the OSI model are not used any more, the *model* itself is actually quite general and still valid, and the features discussed at each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used.

1.4.1 The OSI Reference Model

The model is called the ISO **OSI** (**Open Systems Interconnection**) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the **OSI** **model** for short.

Note that the OSI model itself is not a network architecture because it does not specify the exact services and protocols to be used in each layer. It just tells what each layer should do.

**The Physical Layer**

**The Data Link Layer**

**The Network Layer**

**The Transport Layer**

**The Session Layer**

**The Presentation Layer**

**The Application Layer**

1.4.2 The TCP/IP Reference Model

The Link Layer

1.4.3 The Model used in this book

1.4.4 A Comparison of the OSI and TCP/IP Reference Models

1.5 Example Networks