Online Course Ware

COMPUTER COMMUNICATION AND NETWORKING

EC605C

Prepared by: Dr. Surajit Basak, Dr. Kaushik Roy

Contacts: 3L Credits: 3

Syllabus :

Module I

Overview of Data Communication and Networking: [2L]

Introduction; network criteria, physical structure (type of connection, topology), categories of network (LAN, MAN,WAN); Internet: brief history, Protocols and standards; Reference models: OSI reference model, TCP/IP reference model, their comparative study.

Physical Level: [4L]

Transmission media (guided & unguided); Circuit switching: time division & space division switch, TDM bus

Module II

Data link Layer: [5L]

Types of errors, framing (character and bit stuffing), error detection & correction methods; Flow control; Protocols: Stop & wait ARQ, Go-Back- N ARQ, Selective repeat ARQ, HDLC

Medium Access sub layer: [4L]

Point to Point Protocol, Token Ring; Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA Traditional Ethernet, fast Ethernet (in brief);

Module III

Network layer: [6L]

Internetworking & devices: Repeaters, Hubs, Bridges(Basic Idea), Switches, Router, Gateway; Addressing : IP addressing, subnetting; Routing : techniques, static vs. dynamic routing , Source and Hop-by-Hop routing (Dijkastra, Bellman Ford Algorithm),Unicast Routing Protocols: RIP, OSPF, BGP; Other Protocols: ARP, IP, ICMP, IPV6

Transport layer: [3L]

Process to Process delivery; UDP; TCP; Congestion Control: Open Loop, Closed Loop choke packets (Concept); Leaky bucket algorithm, Token bucket algorithm,

Module IV

Application Layer [6L]

Introduction to DNS, SMTP, SNMP, FTP, HTTP & WWW; Security: Cryptography (Public, Private Key based), Digital Signature, Firewalls.

Modern topics: ATM, DSL technology, Architecture & Operation in brief Wireless LAN: IEEE 802.11(WSN), Introduction to blue-tooth, Zigbee

Text Books:

1. B. A. Forouzan – ―Data Communications and Networking (3rd Ed.) ― – TMH
2. A. S. Tanenbaum – ―Computer Networks (4th Ed.)‖ – Pearson Education/PHI
3. W. Stallings – ―Data and Computer Communications (5th Ed.)‖ – PHI/ Pearson Education
4. Zheng & Akhtar, Network for Computer Scientists & Engineers, OUP
5. Black, Data & Computer Communication, PHI
6. Shay, Understanding Data Communication & Network, Vikas

Reference Books:

1. Kurose and Rose – ― Computer Networking -A top down approach featuring the internet‖ – Pearson Education
2. Leon, Garica, Widjaja – ―Communication Networks‖ – TMH
3. Walrand – ―Communication Networks‖ – TMH.

Course Objective:

1. An understanding of how devices like Hub, Switch, Router and Bridge are used in network.
2. An understanding of how securely data can be transmitted from one place to remotely place using various protocols.

Course Outcome:

After the course, student will be able to

1. Analyze various protocols used in data communication
2. Design networking structure in data communication.
3. Transmit data securely from one place to another.

CO – PO Mapping :

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1	H	H			M	L						
2		M		H	H							
3	M	H	M		M							

Problem analysis: Analyse performance of a large network system, checking no of packets transmitted and received

Design/development of solutions: Conducting experiments in network setup

Individual and team work: Setup network among different departments and provides security

Modern tool usage: Share knowledge regarding up gradation of computer network.

MODULE 1

INTRODUCTION:

Data communications refers to the transmission of this digital data between two or more computers and a computer network or data network is a telecommunications network that allows computers to exchange data. The physical connection between networked computing devices is established using either cable media or wireless media [1].

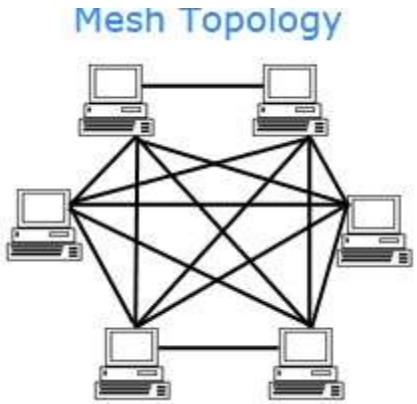
A network consists of two or more computers that are linked in order to share resources (such as printers and CDs), exchange files, or allow electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams. Networking refers to the total process of creating and using computer networks, with respect to hardware, protocols and software, including wired and wireless technology. It involves the application of theories from different technological fields, like IT, computer science and computer/electrical engineering [2].

Physical Structure:

There are four basic types of topologies available.

1. **Mesh Topology –**

   In the mesh topology every device has a dedicated point-to-point link to every other device. In a mesh topology, each physical link carries information only between the two devices that it connects. If it is a duplex connection, you only need half the physical links, since each link travels both ways.



Advantages of a Mesh topology

1. Avoid traffic since each link can carry its own data and none are being shared

2. I f one link breaks, the rest of the network is still functional

3. Privacy since only the dedicated device receives the message.

4. Easy to detect a problem in the network by discovering which device is having problems and examining the link that connects to it.

Disadvantages of a Mesh topology

1. A lot of cables are needed

2. Too many cables too much cost

3. Too many cables not enough physical space

2. Star Topology–

Each device is connected to a hub through a dedicated point-to-point link. The devices are not directly linked to each other. If one device wants to send data to another, it sends it first to the hub, which then forwards the data to the other connected device.

Advantages of a Star topology

1. Less expensive than mesh

2. Easy to install, easy to configure

3. If one link fails the network can still function

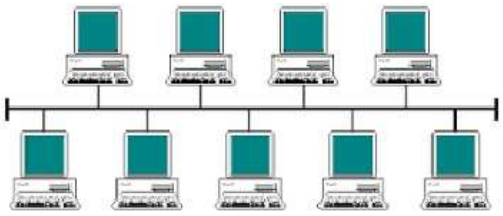
3. Bus Topology–

Multipoint connection. One long cable acts as a backbone; other devices are connected through a drop line and a tap in the link.

Drop line – a connection running between a device and a main cable.

Tap – a physical device that punctures the cable and connects to it.

The longer the cable and the more taps it has the weaker the signal becomes. Taps should be a short distance from each other.



Advantages of a Bus topology

1. Easy to install

2. Minimal Cable

Disadvantages of a Bus topology

1. Difficult reconnection

2. Difficult to find the problem

3. Difficult to add new devices

4. Break stops all transmission of data

4. Ring Topology –

The devices in a ring topology has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction from device to device until the destination is reach. Each device has a repeater that passes the data received that is intended for another device along.

Advantages of a Ring topology

1. Easy to install

2. Easy to reconfigure

3. Easy to detect a problem

Disadvantages of a Ring topology

Break means the whole system is dead

Local Area Network

A **L***ocal-***A***rea* **N***etwork (***LAN***)* is a computer network that spans a relatively small area. Most often, a LAN is confined to a single room, building or group of buildings; however, one LAN can be connected to other LANs over any distance via telephone lines and radio waves.

Metropolitan Area Network

A Metropolitan Area Network, or **MAN**, consists of a computer network across an entire city, college campus or small region. A MAN is larger than a LAN, which is typically limited to a single building or site. Depending on the configuration, this type of network can cover an area from several miles to tens of miles. A MAN is often used to connect several LANs together to form a bigger network. When this type of network is specifically designed for a college campus, it is sometimes referred to as a campus area network, or CAN [4].

Wide Area Network

A **wide area network**, or **WAN**, occupies a very large area, such as an entire country or the entire world. A WAN can contain multiple smaller networks, such as LANs or MANs. The Internet is the best-known example of a public WAN. [4]

Internet Introduction

Internet is a network of networks that connects millions of Computers to form a network and communicating with each other for sharing resources. We can get information, access data, communicate with others, shop, play games and many more by connecting your Computer through Internet. The World Wide Web or web is a platform with various websites that helps us to access global information over the Internet. A web-browser application is mainly important that helps you to access all the information‘s from various websites through Internet. So Internet is everything and needs many things that can help you gather information from worldwide.

Brief History of Internet

Internet has been the main source in different fields like Education, Science, research and others to develop, design and do many other things. It is a tangible entity that relies on physical infrastructure to connect a network other networks. The Internet concept was first coined through networking by J C R Liklider in 1962 which initially termed as Galactic network which was used to interconnect a set of computers for accessing data and programs. Since then DARPA concept came into existence and finally ARPANET where the first host Computer was connected. The ARPANET was used earlier as a networking technology for **first electronic mail** messaging service. Finally ARPANET became the Internet which works on multiple independent networks with its network architecture design. The packet switching method also introduced between networks that makes data communication from one place to another easy.

On October 24, 1995 the FNC passed a resolution proposing the term as Internet with various members of Internet and intellectual property communities. The resolution passed states that Internet is a global information system which is logically linked by global unique space address based on IP or Internet Protocol and support communications using TCP or Transmission Control Protocol. It also states that it provides the users accessible to all the services privately or publicly for communications. Since then with large evolution and technology the networking concept has changed. There came peer to peer, client/server model and more that allows connecting personal computers to a network. Physical cables, telephone wires, networking devices, LAN, WAN and various others also came into existence to the new generation that distribute the networks globally.

There are various connection types like wired, wireless, 2G, 3G and 4G that leads to increase the network capacity overall.

Network Protocols

Network protocols are formal standards and policies comprised of rules, procedures and formats that define communication between two or more devices over a network. Network protocols govern the end-to-end processes of timely, secure and managed data or network communication.

The *Internet Protocol* (IP) is the primary network layer protocol in the TCP/IP protocol suite, and handles the movement of datagrams across a network. The main purpose of IP is to provide a connectionless, best-effort delivery service for datagrams through an internetwork, and to provide fragmentation and reassembly of datagrams to support data links with different *maximum transmission unit* (MTU) sizes. The service provided by IP is *unreliable* in the sense that delivery is not guaranteed. Error and flow control are left to higher level protocols such as the *Transmission Control Protocol* (TCP). Although it is an Internet protocol, IP can be used on any kind of network.

IP has a maximum packet size of 64 kilobytes (65,535 bytes). Because this is larger than most networks can handle, IP can break a datagram down into smaller datagrams when necessary. When the first fragment of a datagram that has been fragmented in this manner arrives at its destination, a *reassembly timer* is started. Unless all of the datagram fragments are received before the timer expires, all of the received fragments are discarded. A sequence number in the packet's IP header enables fragments to be reassembled in the correct order. The IP datagram header is defined in terms of 32-bit *words* (four bytes), and comprises a maximum of six words (24 bytes) in total.

IP datagram header fields:

- Version number - the IP version number.
- Header length - the total length of the IP header in bytes.

- Type of service - usually ignored (most implementations treat all datagrams as having the same priority).
- Packet length - the total length of the datagram in bytes.
- Identification - identifies a datagram as part of a particular message.
- Flags - if the DF (Don't Fragment) flag is set, the datagram cannot be fragmented. If the MF (More Fragments) flag is set, further datagrams that are part of the current message are still to come.
- Fragment offset - the offset of the current datagram from the beginning of the message.
- Time to live (TTL) - the time in seconds that a datagram can persist on the network before it is discarded.
- Transport protocol - identifies the transport protocol used (for example, TCP = 6).
- Header checksum - a checksum for the IP Datagram header itself.
- Source address - the 32-bit source IP address.
- Destination address - the 32-bit destination IP address.
- Options - an optional field comprising several variable length codes.
- Padding - used to ensure the header is a discrete number of bytes.

Once the datagram, complete with its IP header, has been constructed, the first "hop" en route to the destination is determined. This could be the destination computer itself, if it is on the same local network as the source computer, or the default gateway router if the destination computer is on a different network. If a specific route is to be used, routing information is added to the header using the appropriate option, and the datagram is handed down to the link layer.

OSI Model

The **Open Systems Interconnection model** (**OSI model**) is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard protocols [7]. There are seven layers:

1. Physical layer
2. Data link layer
3. Network layer
4. Transport layer
5. Session layer
6. Presentation layer
7. Application layer

Physical (Layer 1)

OSI Model, Layer 1 conveys the bit stream - electrical impulse, light or radio signal — through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects. Fast Ethernet, RS232, and ATM are protocols with physical layer components [8].

Data Link (Layer 2)

At OSI Model, Layer 2, data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub layer controls how

a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking [8].

Network (Layer 3)

Layer 3 provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing [8].

Transport (Layer 4)

OSI Model, Layer 4, provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer [8].

Session (Layer 5)

This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination [8].

Presentation (Layer 6)

This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer [8].

Application (Layer 7)

OSI Model, Layer 7, supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services. Telnet

and FTP are applications that exist entirely in the application level. Tiered application architectures are part of this layer [8].

TCP/IP Protocol

TCP/IP, the protocol stack that is used in communication over the Internet and most other computer networks, has a five-layer architecture. From the bottom up, these layers and their primary responsibilities are:

1. **Physical layer**: responsible for transmitting bits as a physical signal over some medium, e.g. sending electrical signals over a cable, or transmitting an electromagnetic wave over a wireless link.

2. **Link layer** (also called the data link layer or network access layer): responsible for transferring data between devices that are on the same network. This includes (local) addressing, arbitrating access to a shared medium, and checking for (and sometimes correcting) errors that occurred during physical transmission.

3. **Network layer** (also called Internet layer): responsible for transferring data between different networks. This includes (global) addressing and routing.

4. **Transport layer**: responsible for end-to-end communication. The two most common protocols at this layer are UDP, which provides a basic datagram service with no reliability guarantees, and TCP, which provides flow control, connection establishment, and reliable data delivery.

5. **Application layer**: defines that protocols that are used by processes on the end hosts. For example: HTTP, SMTP etc.

Transmission media is a pathway that carries the information from sender to receiver. We use different types of cables or waves to transmit data. Data is **transmitted** normally through electrical

or electromagnetic signals. A **transmission medium** is a material substance (solid, liquid, gas, or plasma) that can propagate energy waves.

Types of transmissions

A transmission may be simplex, half-duplex, or full-duplex.

In simplex transmission, signals are transmitted in only one direction; one station is a transmitter and the other is the receiver. In the half-duplex operation, both stations may transmit, but only one at a time. In full duplex operation, both stations may transmit simultaneously. In the latter case, the medium is carrying signals in both directions at same time. There are two types of transmission media: guided and unguided.

Guided Media:

- Unshielded Twisted Pair (UTP)
- Shielded Twisted Pair (STP)
- Coaxial Cable
- Optical Fiber
- hub

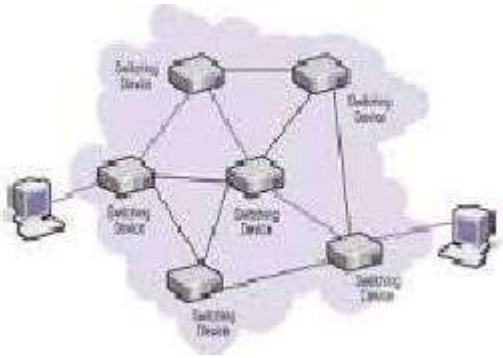
Unguided Media:

Transmission media then looking at analysis of using them unguided transmission media is data signals that flow through the air. They are not guided or bound to a channel to follow. Following are unguided media used for data communication:

- Radio Transmission
- Microwave

Circuit switching

Circuit switching is a method of implementing a telecommunications network in which two network nodes establish a dedicated communications channel (circuit) through the network before the nodes may communicate. The circuit guarantees the full bandwidth of the channel and remains connected for the duration of the communication session. The circuit functions as if the nodes were physically connected as with an electrical circuit [10].



Space Division Switching

The paths in a circuit are separated from each other, spatially in space division switching. Though initially designed for analog networks, it is being used for both analog and digital switching. A Crosspoint switch is mostly referred to as a space division switch because it moves a bit stream from one circuit or bus to another.

The switching system where any channel of one of its incoming PCM highway is connected to any channel of an outgoing PCM highway, where both of them are spatially separated is called the **Space Division Switching**. The Crosspoint matrix connects the incoming and outgoing PCM highways, where different channels of an incoming PCM frame may need to be switched by different Crosspoints in order to reach different destinations [11].

Though the space division switching was developed for the analog environment, it has been carried over to digital communication as well. This requires separate physical path for each signal connection, and uses metallic or semiconductor gates.

Advantages of Space Division Switching

Following is the advantage of Space Division Switching −

- It is instantaneous.

Disadvantages of Space Division Switching

- Number of Crosspoints required to make space-division switching are acceptable in terms of blocking.

Time Division Switching

Time division switching comes under digital switching techniques, where the Pulse Code Modulated signals are mostly present at the input and the output ports. A digital Switching system is one, where the inputs of any PCM highway can be connected to the outputs of any PCM highway, to establish a call.

The incoming and outgoing signals when received and re-transmitted in a different time slot, is called **Time Division Switching.** The digitized speech information is sliced into a sequence of time intervals or slots. Additional voice circuit slots, corresponding to other users are inserted into this bit stream of data. Hence, the data is sent in time frames [11].

The main difference between space division multiplexing and time division multiplexing is sharing of Crosspoints. Crosspoints are not shared in space division switching, whereas they can be shared in time division multiplexing, for shorter periods. This helps in reassigning the Crosspoints and its associated circuitry for other connections as well.

Time division switches use time division multiplexing, in switching. The two popular methods of TDM are TSI (Time and Slot Interchange) and TDM bus. The data sent at the transmitter reaches the receiver in the same order, in an ordinary time division multiplexing whereas, in TSI mechanism, the data sent is changed according to the ordering of slots based on the desired connections. It consists of RAM with several memory locations such as input, output locations and control unit.

Both of the techniques are used in digital transmission. The TDM bus utilizes multiplexing to place all the signals on a common transmission path. The bus must have higher data rate than individual I/O lines. The main advantage of time division multiplexing is that, there is no need of Crosspoints. However, processing each connection creates delay as each time slot must be stored by RAM, then retrieved and then passed on.

Question :

1. What are the seven layers of OSI Model ?

2. Explain the functions performed in Network Layer.

3. Differentiate between Circuit Switching and Packet Switching.

Reference

1. ecomputernotes.com › Computer Networking › Comm. Networks

2. https://searchnetworking.techtarget.com/definition/networking

3. courses.aiu.edu/COMPUTERS/7/Lesson7.pdf

4. https://study.com/.../types-of-networks-lan-wan-wlan-man-san-pan-epnvpn.htmlhttps://www.informationq.com/about-the-interne

5. http://www.technologyuk.net/telecommunications/internet/internet-layer-protocols.shtml

6. https://www.informationq.com/about-the-internet/

7. *https://en.wikipedia.org/wiki/OSI_model*

8. https://www.webopedia.com/quick_ref/OSI_Layers.asp

9. https://en.wikipedia.org/wiki/Transmission_medium

10. https://en.wikipedia.org/wiki/Circuit_switching

11.https://www.tutorialspoint.com/telecommunication_switching_systems_and_networks/telecommunication_switching_systems_and_networks_time_division_switching.htm

MODULE 2
DATA LINK LAYER

Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal. In a single-bit error, a 0 is changed to a 1 or a 1 to a 0. In a burst error, multiple bits are changed. For example, a 11100 s burst of impulse noise on a transmission with a data rate of 1200 bps might change all or some of the12 bits of information.

Single-Bit Error

The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.

Burst Error

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at I kbps, a noise of 11100 s can affect 10 bits; if we are sending data at I Mbps, the same noise can affect 10,000 bits.

FRAMING

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination. The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing. The data link layer, on the other hand, needs to

pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag. Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a flag? The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.

In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers, we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame.

Bit stuffing is the process of adding one extra 0 whenever five consecutive 18 follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Detection Versus Correction

The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error. In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28

possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

Error correction :

Parity Checks

Perhaps the simplest form of error detection is the use of a single **parity bit**. Suppose that the information to be sent, *D* has *d* bits. In an even parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1's in the *d+1* bits (the original information plus a parity bit) is even. For odd parity schemes, the parity bit value is chosen such that there are an odd number of 1's. Figure illustrates an even parity scheme, with the single parity bit being stored in a separate field.

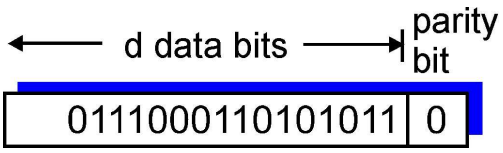


Figure One-bit even parity

Receiver operation is also simple with a single parity bit. The receiver need only count the number of 1's in the received *d+1* bits. If an odd number of 1-valued bits are found with an even parity scheme, the receiver knows that at least one bit error has occurred. More precisely, it knows that some *odd* number of bit errors have occurred.

But what happens if an even number of bit errors occur? You should convince yourself that this would result in an undetected error. If the probability of bit errors is small and errors can be assumed to occur independently from one bit to the next, the probability of multiple bit errors in a packet would be extremely small. In this case, a single parity bit might suffice. However, measurements have shown that rather than occurring independently, errors are often clustered together in "bursts." Under burst error conditions, the probability of undetected errors in a frame protected by single-bit-partity can approach 50 percent [Spragins 1991]. Clearly, a more robust error detection scheme is needed (and, fortunately, is used in practice!). But before examining

error detection schemes that are used in practice, let's cosider a simple generalization of one-bit parity that will provide us with insight into error correction techniques.

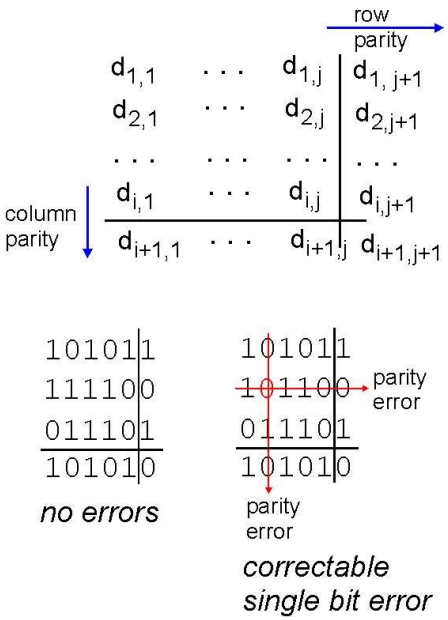


Figure Two-dimensional even parity

Figure shows a two-dimensional generalization of the single-bit parity scheme. Here, the *d* bits in *D* are divided into *i* rows and *j* columns. A parity value is computed for each row and for each column. The resulting *i+j+1* parity bits are the data link frame's error detection bits.

Suppose now that a single bit error occurs in the original *d* bits of information. With this **twodimensional parity** scheme, the parity of both the column and the row containing the flipped bit will be in error. The receiver can thus not only *detect* the fact that a single bit error has occurred, but can use the column and row indices of the column and row with parity errors to actually identify the bit that was corrupted and *correct* that error! Figure shows an example in which the 0-valued bit in position (1,1) is corrupted and switched to a 1 -- an error that is both detectable and correctable at the receiver. Although our discussion has focussed on the original *d* bits of information, a single error in the parity bits themselves is also detectable and correctable. Two dimensional parity can also detect (but not correct!) any combination of two errors in a packet.

Other properties of the two-dimensional parity scheme are explored in the problems at the end of the chapter.

The ability of the receiver to both detect and correct errors is known as **forward error correction (FEC).** These techniques are commonly used in audio storage and playback devices such as audio CD's. In a network setting, FEC techniques can be used by themselves, or in conjunction with the ARQ techniques we examined in Chapter 3. FEC techniques are valuable because they can decrease the number of sender retransmissions required. Perhaps more importantly, they allow for immediate correction of errors at the receiver. This avoids having to wait the round-trip propagation delay needed for the sender to receive a NAK packet and for the retransmitted packet to propagate back to the receiver -- a potentially important advantage for real-time network applications [Rubenstein 1998]. Recent work examining the use of FEC in error control protocols include [Biersack 1992, Nonnenmacher 1998, Byers 1998, Shacham 1990].

5.2.2 Checksumming Methods

In checksumming techniques, the *d* bits of data in Figure 5.2-1 are treated as a sequence of *k*-bit integers. One simple checksumming method is to simply sum these *k*-bit integers and use the resulting sum as the error detection bits. The so-called **Internet checksum** [RFC 1071] is based on this approach -- bytes of data are treated as 16-bit integers and their ones-complement sum forms the Internet checksum. A receiver calculates the checksum it calculates over the received data and checks whether it matches the checksum carried in the received packet. RFC1071 [RFC 1071] discusses the Internet checksum algorithm and its implementation in detail. In the TCP/IP protocols, the Internet checksum is computed over all fields (header and data fields included). In other protocols, e.g., XTP [Strayer 1992], one checksum is computed over the header, with another checksum computed over the entire packet.

McAuley [McAuley 1994] describe improved weighted checksum codes that are suitable for high-speed software implementation and Feldmeier [Feldmeier 1995] presents fast software implementation techniques for not only weighted checksum codes, but CRC (see below) and other codes as well

5.2.3 Cyclic redundancy check

An error detection technique used widely in today's computer networks is based on **cyclic redundancy check (CRC) codes**. CRC codes are also known as **polynomial codes**, since it is

possible to view the bit string to be sent as a polynomial whose coefficients are the 0 and 1 values in the bit string, with operations on the bit string interpreted as polynomial arithmetic.

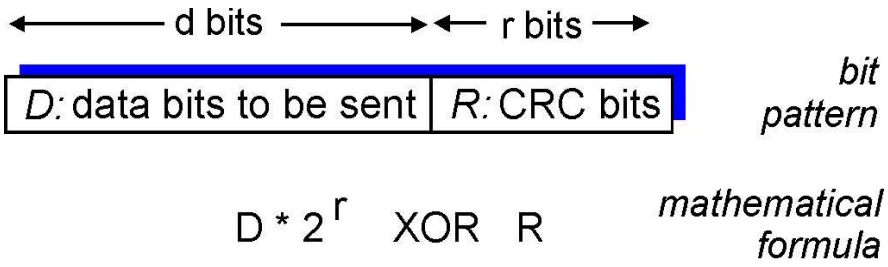


Figure CRC codes

CRC codes operate as follows. Consider the *d-bit* piece of data, *D*, that the sending node wants to send to the receiving node. The sender and receiver must first agree on a *r+1* bit pattern, known as a **generator**, which we will denote as G. We will require that the most significant (leftmost) bit of *G* be a 1. The key idea behind CRC codes is shown in Figure. For a given piece of data, *D*, the sender will choose *r* additional bits, *R*, and append them to *D* such that the resulting *d+r* bit pattern (interpreted as a binary number) is exactly divisible by *G* using modulo 2 arithmetic. The process of error checking with CRC's is thus simple: the receiver divides the *d+r* received bits by *G*. If the remainder is non-zero, the receiver knows that an error has occurred; otherwise the data is accepted as being correct.

All CRC calculations are done in modulo 2 arithmetic without carries in addition or borrows in subtraction. This means that addition and subtraction are identical, and both are equivalent to the bitwise exclusive-or (XOR) of the operands. Thus, for example,

1011XOR0101=1110
1001 XOR 1101 = 0100

Also, we similarly have

1011-0101=1110
1001 - 1101 = 0100

Multiplication and division are the same as in base 2 arithmetic, except that any required addition or subtraction is done without carries or borrows. As in regular binary arithmetic, multiplication

by 2^k left shifts a bit pattern by k places. Thus, given D and R, the quantity $D*2r$ *XOR* R yields the $d+r$ bit pattern shown in Figure 5.2-4. We'll use this algebraic characterization of the $d+r$ bit pattern from Figure 5.2-4 in our discussion below.

Let us now turn to the crucial question of how the sender computes R. Recall that we want to find R such that there is an n such that

$D*2r$XOR $R = nG$

That is, we want to choose R such that G divides into $D*2r$XOR R without remainder. If we exclusive-or (i.e., add modulo 2, without carry) R to both sides of the above equation, we get

$D*2r = nG$ XOR R

This equation tells us that if we divide $D*2^r$ by G, the value of the remainder is precisely R. In other words, we can calculate R as

R = remainder ($D*2r$ / G)

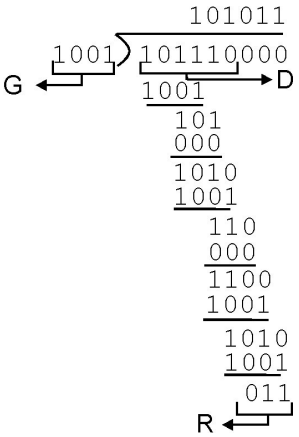


Figure An example CRC calculation

Figure illustrates this calculation for the case of D = 101110, d = 6 and G = 1001, r=3. The nine bits transmitted in this case are 101110 011. You should check these calculations for yourself and also check that indeed $D2^r$ = 101011 * G XOR R.

International standards have been defined for 8-, 12-, 16- and 32-bit generators, *G*. An 8-bit CRC is used to protect the 5-byte header in ATM cells. The CRC-32 32-bit standard, which has been adopted in a number of link-level IEEE protocols, uses a generator of

G_{CRC-32} = 10000010011000001000111010110111

Each of the CRC standards can detect burst errors of less than *r+1* bits and any odd number of bit errors. Furthermore, under appropriate assumptions, a burst of length greater than*r+1* bits is detected with probability 1 - 0.5^r.

https://www.net.t-labs.tu-berlin.de/teaching/computer_networking/05.02.htm

Flow Control

Modern data networks are designed to support a diverse range of hosts and communication mediums. Consider a 933 MHz Pentium-based host transmitting data to a 90 MHz 80486/SX. Obviously, the Pentium will be able to drown the slower processor with data. Likewise, consider two hosts, each using an Ethernet LAN, but with the two Ethernets connected by a 56 Kbps modem link. If one host begins transmitting to the other at Ethernet speeds, the modem link will quickly become overwhelmed. In both cases, flow control is needed to pace the data transfer at an acceptable speed.

Flow Control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data should not be allowed to overwhelm the receiver. Receiver should also be able to inform the transmitter before its limits (this limit may be amount of memory used to store the incoming data or the processing power at the receiver end) are reached and the sender must send fewer frames. Hence, Flow control refers to the set of procedures used to restrict the amount of data the transmitter can send before waiting for acknowledgment.

There are two methods developed for flow control namely Stop-and-wait and Sliding-window. Stop-and-wait is also known as Request/reply sometimes. Request/reply (Stop-and-wait) flow

control requires each data packet to be acknowledged by the remote host before the next packet is sent. This is discussed in detail in the following subsection. Sliding window algorithms, used by TCP, permit multiple data packets to be in simultaneous transit, making more efficient use of network bandwidth.

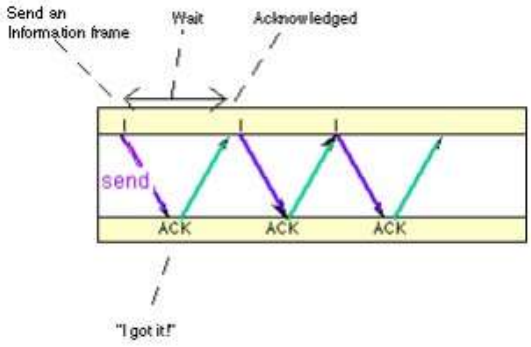
STOP-AND-WAIT

This is the simplest form of flow control where a sender transmits a data frame. After receiving the frame, the receiver indicates its willingness to accept another frame by sending back an ACK frame acknowledging the frame just received. The sender must wait until it receives the ACK frame before sending the next data frame. This is sometimes referred to as ping-pong behavior, request/reply is simple to understand and easy to implement, but not very efficient.

In LAN environment with fast links, this isn't much of a concern, but WAN links will spend most of their time idle, especially if several hops are required. Figure illustrates the operation of the stop-and-wait protocol. The blue arrows show the sequence of data frames being sent across the link from the sender (top to the receiver (bottom). The protocol relies on two-way transmission (full duplex or half duplex) to allow the receiver at the remote node to return frames acknowledging the successful transmission. The acknowledgements are shown in green in the diagram, and flow back to the original sender. A small processing delay may be introduced between reception of the last byte of a Data PDU and generation of the corresponding ACK.

Major drawback of Stop-and-Wait Flow Control is that only one frame can be in transmission at a time, this leads to inefficiency if propagation delay is much longer than the transmission delay.

Some protocols pretty much require stop-and-wait behavior. For example, Internet's Remote Procedure Call (RPC) Protocol is used to implement subroutine calls from a program on one machine to library routines on another machine. Since most programs are single threaded, the sender has little choice but to wait for a reply before continuing the program and possibly sending another request.



Sliding Window

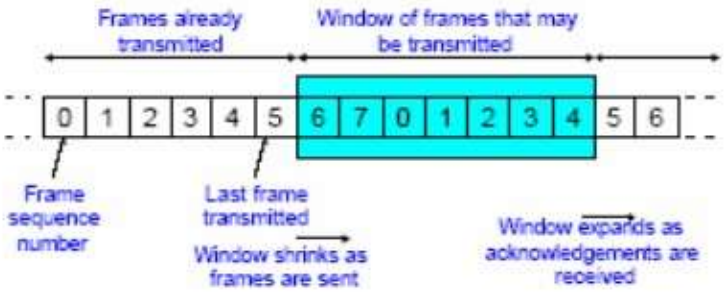
With the use of multiple frames for a single message, the stop-and-wait protocol does not perform well. Only one frame at a time can be in transit. In stop-and-wait flow control, if a > 1, serious inefficiencies result. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time. Efficiency can also be improved by making use of the full-duplex line.

To keep track of the frames, sender station sends sequentially numbered frames. Since the sequence number to be used occupies a field in the frame, it should be of limited size. If the header of the frame allows k bits, the sequence numbers range from 0 to $2^k - 1$. Sender maintains a list of sequence numbers that it is allowed to send (sender window). The size of the sender‘s window is at most $2^k - 1$. The sender is provided with a buffer equal to the window size. Receiver also maintains a window of size $2^k - 1$.

The receiver acknowledges a frame by sending an ACK frame that includes the sequence number of the next frame expected. This also explicitly announces that it is prepared to receive the next N frames, beginning with the number specified. This scheme can be used to acknowledge multiple frames. It could receive frames 2, 3, 4 but withhold ACK until frame 4 has arrived. By returning an ACK with sequence number 5, it acknowledges frames 2, 3, 4 in one go. The receiver needs a buffer of size 1. Sliding window algorithm is a method of flow control for network data transfers. TCP, the Internet's stream transfer protocol, uses a sliding window algorithm.

A sliding window algorithm places a buffer between the application program and the network data flow. For TCP, the buffer is typically in the operating system kernel, but this is more of an implementation detail than a hard-and-fast requirement.

Data received from the network is stored in the buffer, from where the application can read at its own pace. As the application reads data, buffer space is freed up to accept more input from the network. The window is the amount of data that can be "read ahead" - the size of the buffer, less the amount of valid data stored in it. Window announcements are used to inform the remote host of the current window size. Sender sliding Window: At any instant, the sender is permitted to send frames with sequence numbers in a certain range (the sending window).

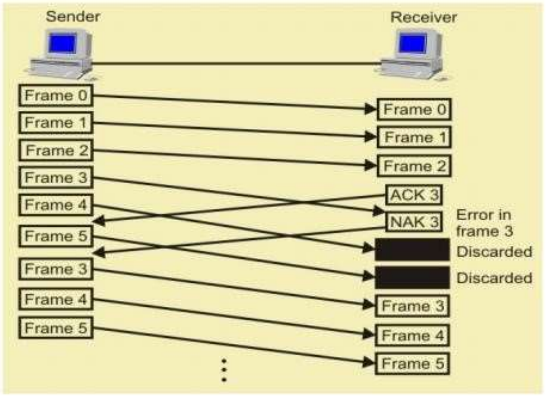


Sender‘s Window

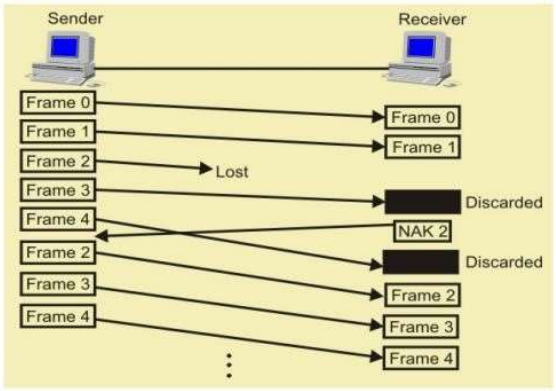
Go-back-N ARQ

The most popular ARQ protocol is the go-back-N ARQ, where the sender sends the frames continuously without waiting for acknowledgement. That is why it is also called as continuous ARQ. As the receiver receives the frames, it keeps on sending ACKs or a NACK, in case a frame is incorrectly received. When the sender receives a NACK, it retransmits the frame in error plus all the succeeding frames. Hence, the name of the protocol is go-back-N ARQ. If a frame is lost, the receiver sends NAK after receiving the next frame. In case there is long delay before sending the NAK, the sender will resend the lost frame after its timer times out. If the ACK frame sent by the receiver is lost, the sender resends the frames after its timer times out. Assuming full-duplex transmission, the receiving end sends piggybacked acknowledgement by using some number in the ACK field of its data frame. Let us assume that a 3-bit sequence number is used and suppose that a station sends frame 0 and gets back an RR1, and then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR1.This might either mean that RR1 is a cumulative ACK or all 8 frames were damaged. This ambiguity can be overcome if the maximum window size is limited to 7, i.e. for a

k-bit sequence number field it is limited to 2^k-1. The number N (=2^k-1) specifies how many frames can be sent without receiving acknowledgement.



Frames in Error in Go-Back-N ARQ



Lost Frames in Go-Back-N ARQ

Medium Access Control (MAC)

Introduction

A network of computers based on multi-access medium requires a protocol for effective sharing of the media. As only one node can send or transmit signal at a time using the broadcast mode, the main problem here is how different nodes get control of the medium to send data, that is

―who goes next?‖. The protocols used for this purpose are known as Medium Access Control (MAC) techniques. The key issues involved here are - Where and how the control is exercised.

Goals of MACs

Medium Access Control techniques are designed with the following goals in mind.

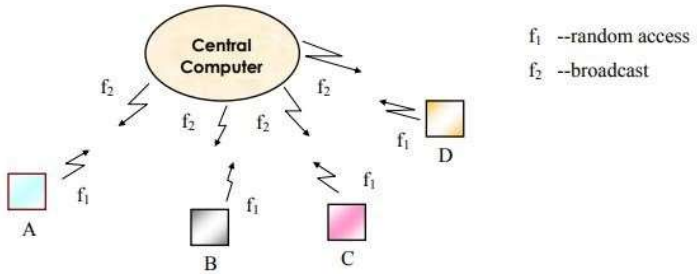
- Initialization: The technique enables network stations, upon power-up, to enter the state required for operation.
- Fairness: The technique should treat each station fairly in terms of the time it is made to wait until it gains entry to the network, access time and the time it is allowed to spend for transmission.
- Priority: In managing access and communications time, the technique should be able to give priority to some stations over other stations to facilitate different type of services needed.
- Limitations to one station: The techniques should allow transmission by one station at a time.
- Receipt: The technique should ensure that message packets are actually received (no lost packets) and delivered only once (no duplicate packets), and are received in the proper order.
- Error Limitation: The method should be capable of encompassing an appropriate error detection scheme.
- Recovery: If two packets collide (are present on the network at the same time), or if notice of a collision appears, the method should be able to recover, i.e. be able to halt all the transmissions and select one station to retransmit.

Token Passing

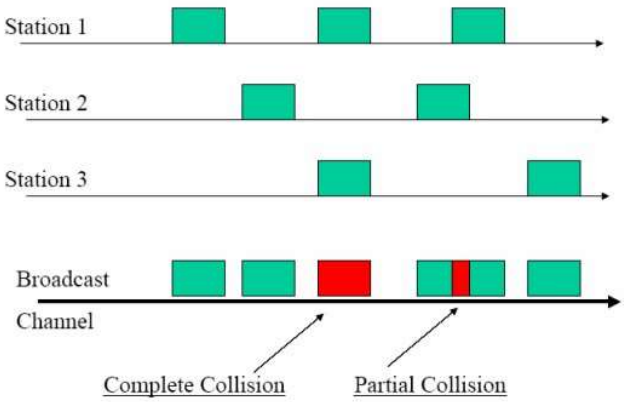
In token passing scheme, all stations are logically connected in the form of a ring and control of the access to the medium is performed using a token. A token is a special bit pattern or a small packet, usually several bits in length, which circulate from node to node. Token passing can be used with both broadcast (token bus) and sequentially connected (token ring) type of networks with some variation in the details as considered in the next lesson. In case of token ring, token is passed from a node to the physically adjacent node. On the other hand, in the token bus, token is passed with the help of the address of the nodes, which form a logical ring. In either case a node currently holding the token has the _right to transmit'. When it has got data to send, it removes the token and transmits the data and then forwards the token to the next logical or physical node in the ring. If a node currently holding the token has no data to send, it simply forwards the token to the next node. The token passing scheme is efficient compared to the polling technique, but it relies on the correct and reliable operation of all the nodes. There exists a number of potential problems, such as lost token, duplicate token, and insertion of a node, removal of a node, which must be tackled for correct and reliable operation of this scheme.

ALOHA

The ALOHA scheme was invented by Abramson in 1970 for a packet radio network connecting remote stations to a central computer and various data terminals at the campus of the university of Hawaii. A simplified situation is shown in Fig.. Users are allowed random access of the central computer through a common radio frequency band f1 and the computer center broadcasts all received signals on a different frequency band f2. This enables the users to monitor packet collisions, if any. The protocol followed by the users is simplest; whenever a node has a packet to sent, it simply does so. The scheme, known as Pure ALOHA, is truly a free-for-all scheme. Of course, frames will suffer collision and colliding frames will be destroyed. By monitoring the signal sent by the central computer, after the maximum round-trip propagation time, an user comes to know whether the packet sent by him has suffered a collision or not.

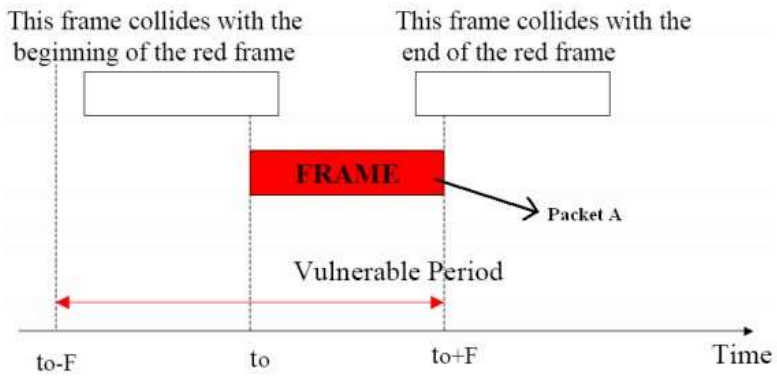


ALOHA scheme for a packet radio system



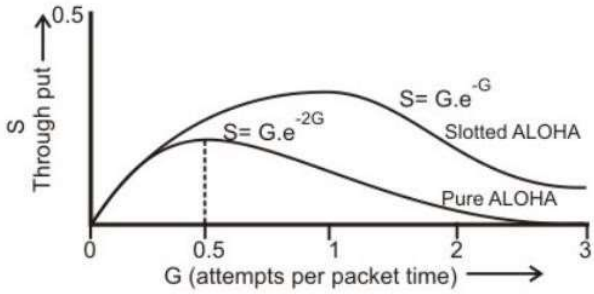
Collision in Pure ALOHA

It may be noted that if all packets have a fixed duration of τ (shown as F in Figure), then a given packet A will suffer collision if another user starts to transmit at any time from τ before to until τ after the start of the packet A as shown in Fig. This gives a vulnerable period of 2τ. Based on this assumption, the channel utilization can be computed. The channel utilization, expressed as throughput S, in terms of the offered load G is given by $S=Ge^{-2G}$.



Vulnerable period in Pure ALOHA

Based on this, the best channel utilization of 18% can be obtained at 50 percent of the offered load as shown in Fig. At smaller offered load, channel capacity is underused and at higher offered load too many collisions occur reducing the throughput. The result is not encouraging, but for such a simple scheme high throughput was also not expected.



Throughput versus offered load for ALOHA protocol

CSMA

The poor efficiency of the ALOHA scheme can be attributed to the fact that a node start transmission without paying any attention to what others are doing. In situations where propagation delay of the signal between two nodes is small compared to the transmission time of a packet, all other nodes will know very quickly when a node starts transmission. This observation is the basis of the carrier-sense multiple-access (CSMA) protocol. In this scheme, a node having

data to transmit first listens to the medium to check whether another transmission is in progress or not. The node starts sending only when the channel is free, that is there is no carrier. That is why the scheme is also known as listen-before talk. There are three variations of this basic scheme as outlined below.

(i) 1-persistent CSMA: In this case, a node having data to send, start sending, if the channel is sensed free. If the medium is busy, the node continues to monitor until the channel is idle. Then it starts sending data.

(ii) Non-persistent CSMA: If the channel is sensed free, the node starts sending the packet. Otherwise, the node waits for a random amount of time and then monitors the channel.

(iii)p-persistent CSMA: If the channel is free, a node starts sending the packet. Otherwise the node continues to monitor until the channel is free and then it sends with probability p.

The efficiency of CSMA scheme depends on the propagation delay, which is represented by a parameter a, as defined below: a = Propagation delay /Packet transmission time.

The throughput of 1-persistent CSMA scheme is shown in Fig. for different values of a. It may be noted that smaller the value of propagation delay, lower is the vulnerable period and higher is the efficiency.

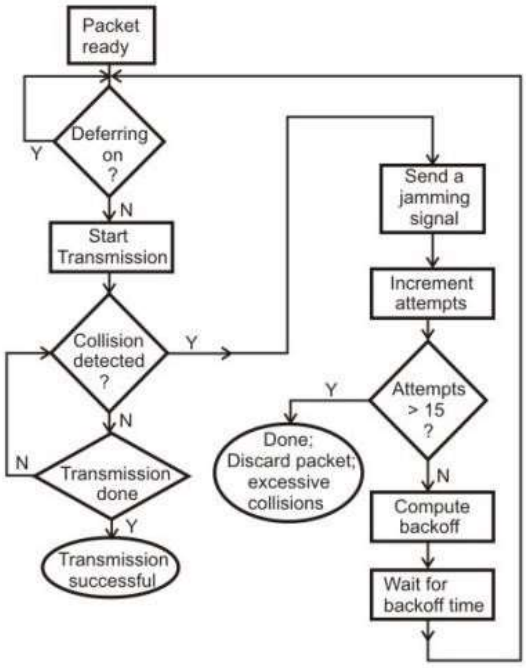
CSMA/CD

CSMA/CD protocol can be considered as a refinement over the CSMA scheme. It has evolved to overcome one glaring inefficiency of CSMA. In CSMA scheme, when two packets collide the channel remains unutilized for the entire duration of transmission time of both the packets. If the propagation time is small (which is usually the case) compared to the packet transmission time, wasted channel capacity can be considerable. This wastage of channel capacity can be reduced if the nodes continue to monitor the channel while transmitting a packet and immediately cease transmission when collision is detected. This refined scheme is known as Carrier Sensed Multiple Access with Collision Detection (CSMA/CD) or Listen-While-Talk.

On top of the CSMA, the following rules are added to convert it into CSMA/CD:

(i) If a collision is detected during transmission of a packet, the node immediately ceases transmission and it transmits jamming signal for a brief duration to ensure that all stations know that collision has occurred.

(ii) After transmitting the jamming signal, the node waits for a random amount of time and then transmission is resumed.



Binary exponential back off algorithm used in CSMA/CD

Protocol	Throughput
ALOHA	$S = Ge^{-2G}$
Slotted ALOHA	$S = Ge^{-G}$
Nonpersistent CSMA	$S = \frac{Ge^{-aG}}{[G(1+2a)+e^{-aG}]}$
Nonpersistent CSMA/CD	$S = \frac{Ge^{-aG}}{[Ge^{-aG}+3aG(1-e^{-aG})+(2-e^{-aG})]}$

Comparison of the Throughputs for the Contention-based MACs

Performance Comparison between CSMA/CD and Token ring: It has been observed that smaller the mean packet length, the higher the maximum mean throughput rate for token passing compared to that of CSMA/CD. The token ring is also least sensitive to workload and propagation effects compared to CSMS/CD protocol. The CSMA/CD has the shortest delay under light load conditions, but is most sensitive to variations to load, particularly when the load is heavy. In CSMA/CD, the delay is not deterministic and a packet may be dropped after fifteen collisions based on binary exponential back off algorithm. As a consequence, CSMA/CD is not suitable for real-time traffic.

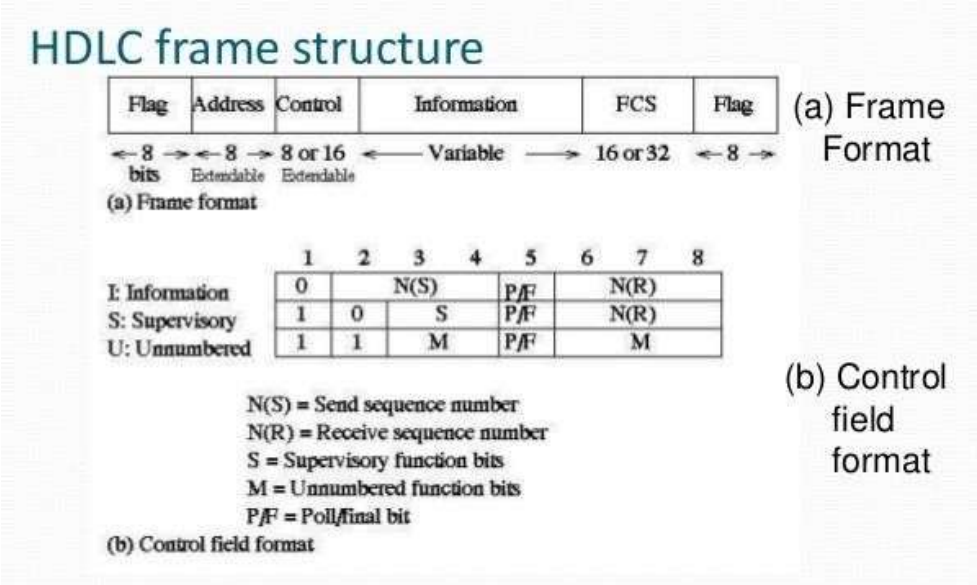
HDLC (High-level Data Link Control)

HDLC (High-level Data Link Control) is a group of protocols or rules for transmitting data between network points (sometimes called nodes). In HDLC, data is organized into a unit (called a *frame*) and sent across a network to a destination that verifies its successful arrival. The HDLC protocol also manages the flow or pacing at which data is sent. HDLC is one of the most commonly-used protocols in what is layer 2 of the industry communication reference model called Open Systems Interconnection (OSI). (Layer 1 is the detailed physical level that involves actually generating and receiving the electronic signals. Layer 3 is the higher level that has knowledge about the network, including access to router tables that indicate where to forward or send data. On sending, programming in layer 3 creates a frame that usually contains source and destination network addresses. HDLC (layer 2) encapsulates the layer 3 frame, adding data link control information to a new, larger frame.

Now an ISO standard, HDLC is based on IBM's SDLC protocol, which is widely used by IBM's large customer base in mainframe computer environments. In HDLC, the protocol that is essentially SDLC is known as Normal Response Mode (NRM). In Normal Response Mode, a primary station (usually at the mainframe computer) sends data to secondary stations that may be local or may be at remote locations on dedicated leased lines in what is called a multidrop or multipoint network. (This is not the network we usually think of; it's a nonpublic closed network. In this arrangement, although communication is usually half-duplex.)

Variations of HDLC are also used for the public networks that use the X.25 communications protocol and for frame relay, a protocol used in both and wide area network, public and private.

In the X.25 version of HDLC, the data frame contains a packet. (An X.25 network is one in which packets of data are moved to their destination along routes determined by network conditions as perceived by routers and reassembled in the right order at the ultimate destination.) The X.25 version of HDLC uses peer-to-peer communication with both ends able to initiate communication on duplex links. This mode of HDLC is known as Link Access Procedure Balanced (LAPB).



FAST ETHERNET:

Fast Ethernet is an extension of the 10 megabit Ethernet standard. It runs on twisted pair or optical fiber cable in a star wired bus topology, similar to the IEEE standard 802.3i called 10BASE-T, itself an evolution of 10BASE5 (802.3) and 10BASE2 (802.3a). Fast Ethernet devices are generally backward compatible with existing 10BASE-T systems, enabling plug-andplay upgrades from 10BASE-T. Most switches and other networking devices with ports capable of Fast Ethernet can perform autonegotiation, sensing a piece of 10BASE-T equipment and setting the port to 10BASE-T half duplex if the 10BASE-T equipment cannot perform auto negotiation itself. The standard specifies the use of CSMA/CD for media access control. A fullduplex mode is also

specified and in practice all modern networks use Ethernet switches and operate in full-duplex mode, even as legacy devices that use half duplex still exist.

A Fast Ethernet adapter can be logically divided into a media access controller (MAC), which deals with the higher-level issues of medium availability, and a physical layer interface (PHY). The MAC is typically linked to the PHY by a four-bit 25 MHz synchronous parallel interface known as a media-independent interface (MII), or by a two-bit 50 MHz variant called reduced media independent interface (RMII). In rare cases the MII may be an external connection but is usually a connection between ICs in a network adapter or even two sections within a single IC. The specs are written based on the assumption that the interface between MAC and PHY will be an MII but they do not require it. Fast Ethernet or Ethernet hubs may use the MII to connect to multiple PHYs for their different interfaces.

The MII fixes the theoretical maximum data bit rate for all versions of Fast Ethernet to 100 Mbit/s. The information rate actually observed on real networks is less than the theoretical maximum, due to the necessary header and trailer (addressing and error-detection bits) on every Ethernet frame, and the required inter packet gap between transmissions.

Reference :

1. www.wikipedia.org

2. www.searchtelecom.techtarget.com

3. www.cisco.com

4. www.techopedia.com

Question :

1. Explain how error is checked by CRC method.

2. Differentiate between stop and wait protocol and sliding window protocol.

MODULE - III

NETWORK LAYER

COMPUTER NETWORK DEVICE

1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about

repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, collision domain of all hosts connected through Hub remains one. Also, they do not have intelligence to find out best path for data packets which leads to inefficiencies and wastage.

Types of Hub

- **Active Hub :-** These are the hubs which have their own power supply and can clean , boost and relay the signal along the network. It serves both as a repeater as well as wiring center. These are used to extend maximum distance between nodes.
- **Passive Hub :-** These are the hubs which collect wiring from nodes and power supply from active hub. These hubs relay signals onto the network without cleaning and boosting them and can‘t be used to extend distance between nodes.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

Types of Bridges

Transparent Bridges :- These are the bridge in which the stations are completely unaware of the bridge‘s existence i.e. whether or not a bridge is added or deleted from the network ,

reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

Source Routing Bridges :- In these bridges, routing operation is performed by source station and the frame specifies which route to follow. The hot can discover frame by sending a special frame called discovery frame, which spreads through the entire network using all possible paths to destination.

NETWORK LAYER

IP ADDRESS :

IPv4 addresses 32 bit binary addresses (divided into 4 octets) used by the Internet Protocol (OSI Layer 3) for delivering packet to a device located in same or remote network. MAC address (Hardware address) is a globally unique address which represents the network card and cannot be changed. IPv4 address refers to a logical address, which is a configurable address used to identify which network this host belongs to and also a network specific host number. In other words, an IPv4 address consists of two parts; a network part and a host part.

This can be compared to your home address. A letter addressed to your home address will be delivered to your house because of this logical address. If you move to another house, your address will change, and letters addressed to you will be sent to your new address. But the person who the letter is being delivered to, that is ―you‖, is still the same.

IPv4 addresses are stored internally as binary numbers but they are represented in decimal numbers because of simplicity.

An example of IPv4 address is 192.168.10.100, which is actually 11000000.10101000.00001010.01100100.

For Each network, one address is used to represent the network and one address is used for broadcast. Network address is an IPv4 address with all host bits are "0". Broadcast address is an IPv4 address with all host bits are "1".

That means, for a network, the first IPv4 address is the network address and the last IPv4 address is the broadcast address.You cannot configure these addresses for your devices. All the usable IPv4 addresses in any IP network are between network address and broadcast address.

We can use the following equation for find the number of usable IPv4 addresses in a network (We have to use two IPv4 addresses in each network to represent the network id and the broadcat id.)

Number of usable IPv4 addresses = (2^n)-2. Where "n" is the number of bits in host part.

Many IPv4 addresses are reserved and we cannot use those IPv4 address. There are five IPv4 address Classes and certain special addresses.

Class A IPv4 addresses

"Class A" IPv4 addresses are for very large networks. The left most bit of the left most octet of a "Class A" network is reserved as "0". The first octet of a "Class A" IPv4 address is used to identify the Network and the three remaining octets are used to identify the host in that particular network (Network.Host.Host.Host).

The 32 bits of a "Class A" IPv4 address can be represented as

0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.

The minimum possible value for the leftmost octet in binaries is 00000000 (decimal equivalent is 0) and the maximum possible value for the leftmost octet is 01111111 (decimal equivalent is 127). Therefore for a "Class A" IPv4 address, leftmost octet must have a value between 0-127 (0.X.X.X to 127.X.X.X).

The network 127.0.0.0 is known as loopback network. The IPv4 address 127.0.0.1 is used by the host computer to send a message back to itself. It is commonly used for troubleshooting and network testing.

Computers not connected directly to the Internet need not have globally-unique IPv4 addresses. They need an IPv4 addresses unique to that network only. 10.0.0.0 network belongs to "Class A" is reserved for private use and can be used inside any organization.

Class B IPv4 addresses

"Class B" IPv4 addresses are used for medium-sized networks. Two left most bits of the left most octet of a "Class B" network is reserved as "10". The first two octets of a "Class B" IPv4 address is used to identify the Network and the remaining two octets are used to identify the host in that particular network (Network.Network.Host.Host).

The 32 bits of a "Class B" IPv4 address can be represented as 10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.

The minimum possible value for the leftmost octet in binaries is 10000000 (decimal equivalent is 128) and the maximum possible value for the leftmost octet is 10111111 (decimal equivalent is 191). Therefore for a "Class B" IPv4 address, leftmost octet must have a value between 128-191 (128.X.X.X to 191.X.X.X).

Network 169.254.0.0 is known as APIPA (Automatic Private IPv4 addresses). APIPA range of IPv4 addresses are used when a client is configured to automatically obtain an IPv4 address from the DHCP server was unable to contact the DHCP server for dynamic IPv4 address.

Networks starting from 172.16.0.0 to 172.31.0.0 are reserved for private use.

Class C IPv4 addresses

"Class C" IPv4 addresses are commonly used for small to mid-size businesses. Three left most bits of the left most octet of a "Class C" network is reserved as "110". The first three octets of a "Class C" IPv4 address is used to identify the Network and the remaining one octet is used to identify the host in that particular network (Network.Network.Networkt.Host).

The 32 bits of a "Class C" IPv4 address can be represented as 110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.

The minimum possible value for the leftmost octet in binaries is 11000000 (decimal equivalent is 192) and the maximum possible value for the leftmost octet is 11011111 (decimal equivalent is 223). Therefore for a "Class C" IPv4 address, leftmost octet must have a value between 192-223 (192.X.X.X to 223.X.X.X).

Networks starting from 192.168.0.0 to 192.168.255.0 are reserved for private use.

Class D IPv4 addresses

Class D IPv4 addresses are known as multicast IPv4 addresses. Multicasting is a technique developed to send packets from one device to many other devices, without any unnecessary packet duplication. In multicasting, one packet is sent from a source and is replicated as needed in the network to reach as many end-users as necessary. You cannot assign these IPv4 addresses to your devices.

Four left most bits of the left most octet of a "Class D" network is reserved as "1110". The other 28 bits are used to identify the group of computers the multicast message is intended for.

The minimum possible value for the left most octet in binaries is 11100000 (decimal equivalent is 224) and the maximum possible value for the leftmost octet is 11101111 (decimal equivalent is 239). Therefore for a "Class D" IPv4 address, leftmost octet must have a value between 224239 (224.X.X.X to 239.X.X.X).

Class E IPv4 addresses

Class E is used for experimental purposes only and you cannot assign these IPv4 addresses to your devices.

Four left most bits of the left most octet of a "Class E" network is reserved as "1111".

The minimum possible value for the left most octet in binaries is 11110000 (decimal equivalent is 240) and the maximum possible value for the leftmost octet is 11111111 (decimal equivalent is 255). Therefore for a "Class E" IPv4 address, leftmost octet must have a value between 240255 (240.X.X.X to 255.X.X.X).

What is Subnet Mask?

An IPv4 address has two components, a "Network" part and a "Host" part. To identify which part of an IPv4 address is the "Network" part and which part of the IPv4 address is "Host" part, we need another identifier, which is known as "Subnet Mask". IPv4 address is a combination of IPv4 address and Subnet mask and the purpose of subnet mask is to identify which part of an IPv4 address is the network part and which part is the host part. Subnet mask is also a 32 bit number where all the bits of the network part are represented as "1" and all the bits of the host part are represented as "0".

An IP address has two components, the network address and the host address. A subnet mask separates the IP address into the network and host addresses (<network><host>). Subnetting further divides the host part of an IP address into a subnet and host address (<network><subnet><host>) if additional subnetwork is needed. Use the Subnet Calculator to retrieve subnetwork information from IP address and Subnet Mask. It is called a subnet mask because it is used to identify network address of an IP address by perfoming a bitwise AND operation on the netmask.

A Subnet mask is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. Subnet Mask is made by setting network bits to all "1"s and setting host bits to all "0"s. Within a given network, two host addresses are reserved for special purpose, and

cannot be assigned to hosts. The "0" address is assigned a network address and "255" is assigned to a broadcast address, and they cannot be assigned to hosts.

Subnetting an IP network is to separate a big network into smaller multiple networks for reorganization and security purposes. All nodes (hosts) in a subnetwork see all packets transmitted by any node in a network. Performance of a network is adversely affected under heavy traffic load due to collisions and retransmissions.

Applying a subnet mask to an IP address separates network address from host address. The network bits are represented by the 1's in the mask, and the host bits are represented by 0's. Performing a bitwise logical AND operation on the IP address with the subnet mask produces the network address. For example, applying the Class C subnet mask to our IP address 216.3.128.12 produces the following network address:

IP: 1101 1000 . 0000 0011 . 1000 0000 . 0000 1100 (216.003.128.012)

Mask: 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000 (255.255.255.000)

STATIC AND DYNAMIC ROUTING :

Routing algorithms in the context of networking can be classified variously. The prior classification is based on the building and modification of a routing table. This can be done in two manners statically or dynamically. More precisely these are known as static and dynamic routing respectively.

In the Static routing, the table is set up and modified manually whereas in the Dynamic routing the table is built automatically with the help of the routing protocols. Dynamic routing is preferred

over static routing because of the major issue in static routing where in case of link/node failure the system cannot recover. The dynamic routing overcomes from the static routing limitations.

Routing is the process of transferring the packets from one network to another network and delivering the packets to the hosts. The traffic is routed to all the networks in the internetwork by the routers. In the routing process a router must know following things:

- Destination device address.
- Neighbor routers for learning about remote networks.
- Possible routes to all remote networks.
- The best route with the shortest path to each remote network.
- How the routing information can be verified and maintained.

ROUTING PROTOCOL :

The Routing Information Protocol (RIP) defines a way for routers, which connect networks using the Internet Protocol (IP), to share information about how to route traffic among networks. RIP is classified by the Internet Engineering Task Force (IETF) as an Interior Gateway Protocol (IGP), one of several protocols for routers moving traffic around within a larger autonomous system network -- e.g., a single enterprise's network that may be comprised of many separate local area networks (LANs) linked through routers.

Each RIP router maintains a routing table, which is a list of all the destinations (networks) it knows how to reach, along with the distance to that destination. RIP uses a distance vector algorithm to decide which path to put a packet on to get to its destination. It stores in its routing table the distance for each network it knows how to reach, along with the address of the "next hop" router -- another router that is on one of the same networks -- through which a packet has to travel to get to that destination. If it receives an update on a route, and the new path is shorter, it will update its table entry with the length and next-hop address of the shorter path; if the new path is longer, it will wait through a "hold-down" period to see if later updates reflect the higher value as well, and only update the table entry if the new, longer path is stable.

Using RIP, each router sends its entire routing table to its closest neighbors every 30 seconds. (The *neighbors* are the other routers to which this router is connected directly -- that is, the other routers on the same network segments this router is on.) The neighbors in turn will pass the information on to their nearest neighbors, and so on, until all RIP hosts within the network have the same knowledge of routing paths, a state known as *convergence*.

If a router crashes or a network connection is severed, the network discovers this because that router stops sending updates to its neighbors, or stops sending and receiving updates along the severed connection. If a given route in the routing table isn't updated across six successive update cycles (that is, for 180 seconds) a RIP router will drop that route, letting the rest of the network know via its own updates about the problem and begin the process of reconverging on a new network topology.

RIP uses a modified hop count as a way to determine network distance. (*Modified* reflects the fact that network engineers can assign paths a higher cost.) By default, if a router's neighbor owns a destination network (i.e., if it can deliver packets directly to the destination network without using any other routers), that route has one hop, described as a cost of 1. RIP allows only 15 hops in a path. If a packet can't reach a destination in 15 hops, the destination is considered unreachable. Paths can be assigned a higher cost (as if they involved extra hops) if the enterprise wants to limit or discourage their use. For example, a satellite backup link might be assigned a cost of 10 to force traffic follow other routes when available.

RIP has been supplanted mainly due to its simplicity and its inability to scale to very large and complex networks. Other routing protocols push less information of their own onto the network, while RIP pushes its whole routing table every 30 seconds. As a result, other protocols can converge more quickly, use more sophisticated routing algorithms, include latency, packet loss, actual monetary cost and other link characteristics, as well as hop count with arbitrary weighting.

INTERIOR GATEWAY ROUTING PROTOCOL :

Interior Gateway Routing Protocol (**IGRP**) is a distance vector interior gateway protocol (IGP) developed by Cisco. It is used by routers to exchange routing data within an autonomous system.

IGRP is a proprietary protocol. IGRP was created in part to overcome the limitations of RIP (maximum hop count of only 15, and a single routing metric) when used within large networks. IGRP supports multiple metrics for each route, including bandwidth, delay, load, and reliability; to compare two routes these metrics are combined together into a single metric, using a formula which can be adjusted through the use of pre-set constants. By default, the IGRP composite metric is a sum of the segment delays and the lowest segment bandwidth. The maximum configurable hop count of IGRP-routed packets is 255 (default 100), and routing updates are broadcast every 90 seconds (by default).[1] IGRP uses protocol number 9 for communication.[2]

IGRP is considered a classful routing protocol. Because the protocol has no field for a subnet mask, the router assumes that all subnetwork addresses within the same Class A, Class B, or Class C network have the same subnet mask as the subnet mask configured for the interfaces in question. This contrasts with classless routing protocols that can use variable length subnet masks. Classful protocols have become less popular as they are wasteful of IP address space.

OPEN SHORTEST PATH FIRST :

Open Shortest Path First (OSPF) was designed as an interior gateway protocol (IGP), for use in an autonomous system such as a local area network (LAN). It implements Dijkstra's algorithm, also

known as the shortest path first (SPF) algorithm. As a link-state routing protocol it was based on the link-state algorithm developed for the ARPANET in 1980 and the IS-IS routing protocol. OSPF was first standardised in 1989 as RFC 1131, which is now known as OSPF version 1. The development work for OSPF prior to its codification as open standard was undertaken largely by the Digital Equipment Corporation, which developed its own proprietary DECnet protocols.

Routing protocols like OSPF calculate the *shortest* route to a destination through the network based on an algorithm. The first routing protocol that was widely implemented, the Routing Information Protocol (RIP), calculated the shortest route based on hops, that is the number of routers that an IP packet had to traverse to reach the destination host. RIP successfully implemented dynamic routing, where routing tables change if the network topology changes. But RIP did not adapt its routing according to changing network conditions, such as data-transfer rate. Demand grew for a dynamic routing protocol that could calculate the *fastest* route to a destination. OSPF was developed so that the shortest path through a network was calculated based on the *cost* of the route, taking into account bandwidth, delay and load. Therefore OSPF undertakes route cost calculation on the basis of link-cost parameters, which can be weighted by the administrator. OSPF was quickly adopted because it became known for reliably calculating routes through large and complex local area networks.

As a link state routing protocol, OSPF maintains link state databases, which are really network topology maps, on every router on which it is implemented. The *state* of a given route in the network is the cost, and OSPF algorithm allows every router to calculate the cost of the routes to any given reachable destination. Unless the administrator has made a configuration, the link cost of a path connected to a router is determined by the bit rate (1 Gbit/s, 10 Gbit/s, etc) of the interface. A router interface with OSPF will then advertise its link cost to neighbouring routers through multicast, known as the *hello procedure*. All routers with OSPF implementation keep sending hello packets, and thus changes in the cost of their links become known to neighbouring routers. The information about the cost of a link, that is the speed of a point to point connection between two routers, is then cascaded through the network because OSPF routers advertise the information they receive from one neighbouring router to all other neighbouring routers. This process of flooding link state information through the network is known as *synchronisation*. Based on this

information, all routers with OSPF implementation continuously update their link state databases with information about the network topology and adjust their routing tables.

An OSPF network can be structured, or subdivided, into routing *areas* to simplify administration and optimize traffic and resource utilization. Areas are identified by 32-bit numbers, expressed either simply in decimal, or often in the same dot-decimal notation used for IPv4 addresses. By convention, area 0 (zero), or 0.0.0.0, represents the core or *backbone* area of an OSPF network.

While the identifications of other areas may be chosen at will; administrators often select the IP address of a main router in an area as the area identifier. Each additional area must have a connection to the OSPF backbone area. Such connections are maintained by an interconnecting router, known as an area border router (ABR). An ABR maintains separate link-state databases for each area it serves and maintains summarized routes for all areas in the network.

OSPF detects changes in the topology, such as link failures, and converges on a new loop-free routing structure within seconds. OSPF has become a popular dynamic routing protocol. Other commonly used dynamic routing protocols are the RIP and the Border Gateway Protocol (BGP). Today router ssupport at least one interior gateway protocol to advertise their routing tables within a local area network. Frequently implemented interior gateway protocols besides OSPF are RIP, IS-IS, and the proprietary Interior Gateway Routing Protocol (IGRP) by Cisco.

Router relationships

OSPF supports complex networks with multiple routers, including backup routers, to balance traffic load on multiple links to other subnets. Neighboring routers in the same broadcast domain or at each end of a point-to-point link communicate with each other via the OSPF protocol. Routers form *adjacencies* when they have detected each other. This detection is initiated when a router identifies itself in a *Hello* protocol packet. Upon acknowledgment, this establishes a *two-way state* and the most basic relationship. The routers in an Ethernet or Frame Relay network select a *Designated Router* (DR) and a *Backup Designated Router* (BDR) which act as a hub to reduce traffic between routers. OSPF uses both unicast and multicast transmission modes to send "Hello" packets and link state updates.

As a link state routing protocol, OSPF establishes and maintains neighbor relationships for exchanging routing updates with other routers. The neighbor relationship table is called an *adjacency database*. Two OSPF routers are neighbors if they are members of the same subnet and share the same area ID, subnet mask, timers and authentication. In essence, OSPF neighborship is a relationship between two routers that allow them to see and understand each other but nothing more. OSPF neighbors do not exchange any routing information – the only packets they exchange are Hello packets. OSPF adjacencies are formed between selected neighbors and allow them to exchange routing information. Two routers must first be neighbors and only then, can they become adjacent. Two routers become adjacent if at least one of them is Designated Router or Backup Designated Router (on multiaccess type networks), or they are interconnected by a point-to-point or point-to-multipoint network type. For forming a neighbor relationship between, the interfaces used to form the relationship must be in the same OSPF area. While an interface may be configured to belong to multiple areas, this is generally not practiced.
When configured in a second area, an interface must be configured as a secondary interface.

Adjacency state machine

Each OSPF router within a network communicates with other neighboring routers on each connecting interface to establish the states of all adjacencies. Every such communication sequence is a separate *conversation* identified by the pair of router IDs of the communicating neighbors. RFC 2328 specifies the protocol for initiating these conversations (*Hello Protocol*) and for establishing full adjacencies (*Database Description Packets*, *Link State Request Packets*). During its course, each router conversation transitions through a maximum of eight conditions defined by a state machine:

1. Down: The state *down* represents the initial state of a conversation when no information has been exchanged and retained between routers with the Hello Protocol.

2. Attempt: The *Attempt* state is similar to the *Down* state, except that a router is in the process of efforts to establish a conversation with another router, but is only used on NBMA networks.

3. Init: The *Init* state indicates that a HELLO packet has been received from a neighbor, but the router has not established a two-way conversation.

4. 2-Way: The *2-Way* state indicates the establishment of a bidirectional conversation between two routers. This state immediately precedes the establishment of adjacency. This is the lowest state of a router that may be considered as a Designated Router.

5. ExStart: The *ExStart* state is the first step of adjacency of two routers.

6. Exchange: In the *Exchange* state, a router is sending its link state database information to the adjacent neighbor. At this state, a router is able to exchange all OSPF routing protocol packets.

7. Loading: In the *Loading* state, a router requests the most recent Link-state advertisements (LSAs) from its neighbor discovered in the previous state.

8. Full: The *Full* state concludes the conversation when the routers are fully adjacent, and the state appears in all router- and network-LSAs. The link state databases of the neighbors are fully synchronized.

OSPF message

Unlike other routing protocols, OSPF does not carry data via a transport protocol, such as the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP). Instead, OSPF forms IP datagrams directly, packaging them using protocol number 89 for the IP Protocol field.
OSPF defines five different message types, for various types of communication:

Hello

Hello messages are used as a form of greeting, to allow a router to discover other adjacent routers on its local links and networks. The messages establish relationships between neighboring devices (called adjacencies) and communicate key parameters about how OSPF is to be used in the autonomous system or area. During normal operation, routers send hello messages to their neighbors at regular intervals (the *hello interval*); if a router

stops receiving hello messages from a neighbor, after a set period (the *dead interval*) the router will assume the neighbor has gone down.

Database Description (*DBD*)

Database description messages contain descriptions of the topology of the autonomous system or area. They convey the contents of the link-state database (LSDB) for the area from one router to another. Communicating a large LSDB may require several messages to be sent by having the sending device designated as a master device and sending messages in sequence, with the slave (recipient of the LSDB information) responding with acknowledgements.

Link State Request (*LSR*)

Link state request messages are used by one router to request updated information about a portion of the LSDB from another router. The message specifies the link(s) for which the requesting device wants more current information.

Link State Update (*LSU*)

Link state update messages contain updated information about the state of certain links on the LSDB. They are sent in response to a Link State Request message, and also broadcast or multicast by routers on a regular basis. Their contents are used to update the information in the LSDBs of routers that receive them.

Link State Acknowledgment (*LSAck*)

Link state acknowledgement messages provide reliability to the link-state exchange process, by explicitly acknowledging receipt of a Link State Update message.

ROUTING ALGORITHMS

• Routing algorithms that solve a routing problem are based on shortest-path algorithms.

• Two common shortest-path algorithms are Dijkstra‘s Algorithm and the Bellman-Ford Algorithm.

• Routing algorithms fall into two general categories.

Link-State Algorithms

• The network topology and all link costs are known. • Example: Dijkstra‘s Algorithm

. • More complex of the two types.

• Nodes perform independent computations.

• Used in Open Shortest Path First (OSPF) protocol, a protocol intended to replace RIP.

Dijkstra‘s shortest path algorithm

Given a graph and a source vertex in the graph, find shortest paths from source to all vertices in the given graph.

Dijkstra‘s algorithm is very similar to Prim‘s algorithm for minimum spanning tree. Like Prim‘s MST, we generate a *SPT (shortest path tree)* with given source as root. We maintain two sets, one set contains vertices included in shortest path tree, other set includes vertices not yet included in shortest path tree. At every step of the algorithm, we find a vertex which is in the other set (set of not yet included) and has a minimum distance from the source.

Below are the detailed steps used in Dijkstra‘s algorithm to find the shortest path from a single source vertex to all other vertices in the given graph.

Algorithm

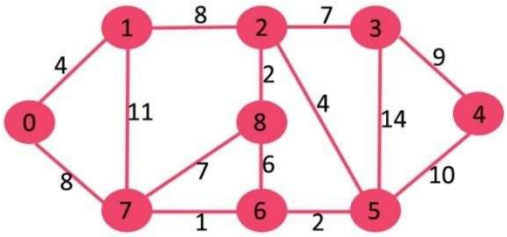
1) Create a set *sptSet* (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.

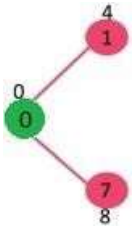
2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.

a) Pick a vertex u which is not there in *sptSet* and has minimum distance value.

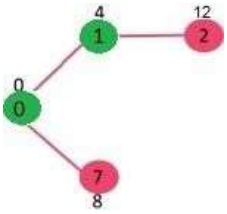
b) Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.

Let us understand with the following example:

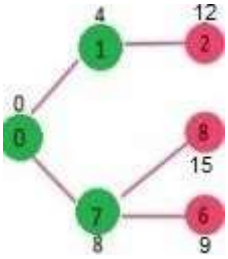




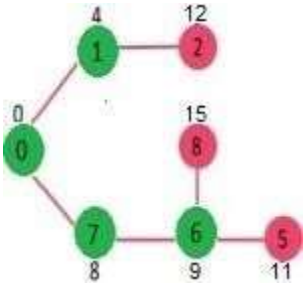
Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). The vertex 1 is picked and added to sptSet. So sptSet now becomes {0, 1}. Update the distance values of adjacent vertices of 1. The distance value of vertex 2 becomes 12.



Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 7 is picked. So sptSet now becomes {0, 1, 7}. Update the distance values of adjacent vertices of 7. The distance value of vertex 6 and 8 becomes finite (15 and 9 respectively).



Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 6 is picked. So sptSet now becomes {0, 1, 7, 6}. Update the distance values of adjacent vertices of 6. The distance value of vertex 5 and 8 are updated.



BELLMAN FORD ALGORITHM

Bellman Ford algorithm helps us find the shortest path from a vertex to all other vertices of a weighted graph.

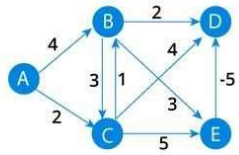
How Bellman Ford's algorithm works

Bellman Ford algorithm works by overestimating the length of the path from the starting vertex to all other vertices. Then it iteratively relaxes those estimates by finding new paths that are shorter than the previously overestimated paths.

By doing this repeatedly for all vertices, we are able to guarantee that the end result is optimized.

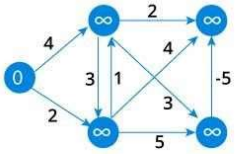
1

Start with a weighted graph



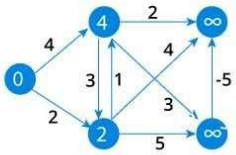
2

Choose a starting vertex and assign infinity path values to all other vertices



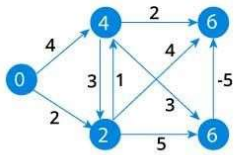
3

Visit each edge and relax the path distances if they are inaccurate



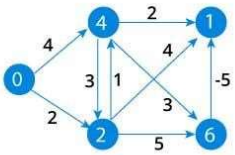
4

We need to do this V times because in the worst case, a vertex's path length might need to be readjusted V times



5

Notice how the vertex at the top right corner had its path length adjusted



6

After all the vertices have their path lengths, we check if a negative cycle is present.

A	B	C	D	E
0	∞	∞	∞	∞
0	4	2	∞	∞
0	3	2	6	6
0	3	2	1	6
0	3	2	1	6

ADDRESS RESOLUTION PROTOCOL

Address Resolution Protocol (ARP) is a protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. For example, in IP Version 4, the most common level of IP in use today, an address is 32 bits long. In an Ethernet local area network, however, addresses for attached devices are 48 bits long. (The physical machine address is also known as a Media Access Control or MAC address.) A table, usually called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions.

IPv6 (Internet Protocol Version 6)

IPv6 (Internet Protocol version 6) is a set of specifications from the Internet Engineering Task Force (IETF) that's essentially an upgrade of IP version 4 (IPv4). The basics of IPv6 are similar to those of IPv4 -- devices can use IPv6 as source and destination addresses to pass packets over a network, and tools like ping work for network testing as they do in IPv4, with some slight variations.

The most obvious improvement in IPv6 over IPv4 is that IP addresses are lengthened from 32 bits to 128 bits. This extension anticipates considerable future growth of the Internet and provides relief for what was perceived as an impending shortage of network addresses. IPv6 also supports auto-configuration to help correct most of the shortcomings in version 4, and it has integrated security and mobility features.

IPv6 features include:

- Supports source and destination addresses that are 128 bits (16 bytes) long.
- Requires IPSec support.
- Uses Flow Label field to identify packet flow for QoS handling by router.
- Allows the host to send fragments packets but not routers.
- Doesn't include a checksum in the header.
- Uses a link-local scope all-nodes multicast address.
- Does not require manual configuration or DHCP.
- Uses host address (AAAA) resource records in DNS to map host names to IPv6 addresses.
- Uses pointer (PTR) resource records in the IP6.ARPA DNS domain to map IPv6 addresses to host names.

- Supports a 1280-byte packet size (without fragmentation).
- Moves optional data to IPv6 extension headers.
- Uses Multicast Neighbor Solicitation messages to resolve IP addresses to link-layer addresses.
- Uses Multicast Listener Discovery (MLD) messages to manage membership in local subnet groups.
- Uses ICMPv6 Router Solicitation and Router Advertisement messages to determine the IP address of the best default gateway.

BORDER GATEWAY PROTOCOL

BGP (Border Gateway Protocol) is protocol that manages how packets are routed across the internet through the exchange of routing and reachability information between edge routers. BGP directs packets between autonomous systems (AS) -- networks managed by a single enterprise or service provider. Traffic that is routed within a single network AS is referred to as internal BGP, or iBGP. More often, BGP is used to connect one AS to other autonomous systems, and it is then referred to as an external BGP or eBGP.

BGP offers network stability that guarantees routers can quickly adapt to send packets through another reconnection if one internet path goes down. BGP makes routing decisions based on paths, rules or network policies configured by a network administrator. Each BGP router maintains a standard routing table used to direct packets in transit. This table is used in conjunction with a separate routing table, known as the routing information base (RIB), which is a data table stored on a server on the BGP router. The RIB contains route information both from directly connected external peers, as well as internal peers, and continually updates the routing table as changes occurs. BGP is based on TCP/IP and uses client-server topology to communicate routing information, with the client-server initiating a BGP session by sending a request to the server.

Reference :

1. www.wikipedia.org

2. www.searchtelecom.techtarget.com

TRANSPORT LAYER

UDP (User Datagram Protocol)

UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss-tolerating connections between applications on the internet.

Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. But there are important differences between the two.

Where UDP enables process-to-process communication, TCP supports host-to-host communication. TCP sends individual packets and is considered a reliable transport medium; UDP sends messages, called datagrams, and is considered a best-effort mode of communications.

In addition, where TCP provides error and flow control, no such mechanisms are supported in UDP. UDP is considered a connectionless protocol because it doesn't require a virtual circuit to be established before any data transfer occurs.

UDP provides two services not provided by the IP layer. It provides port numbers to help distinguish different user requests and, optionally, a checksum capability to verify that the data arrived intact.

TCP has emerged as the dominant protocol used for the bulk of internet connectivity due to its ability to break large data sets into individual packets, check for and resend lost packets, and reassemble packets in the correct sequence. But these additional services come at a cost in terms of additional data overhead and delays called latency.

In contrast, UDP just sends the packets, which means that it has much lower bandwidth overhead and latency. With UDP, packets may take different paths between sender and receiver and, as a result, some packets may be lost or received out of order.

Applications of UDP

UDP is an ideal protocol for network applications in which perceived latency is critical, such as in gaming and voice and video communications, which can suffer some data loss without adversely affecting perceived quality. In some cases, forward error correction techniques are used to improve audio and video quality in spite of some loss.

UDP can also be used in applications that require lossless data transmission when the application is configured to manage the process of retransmitting lost packets and correctly arranging received packets. This approach can help to improve the data transfer rate of large files compared to TCP.

In the Open Systems Interconnection (OSI) communication model, UDP, like TCP, is in Layer 4, the transport layer. UDP works in conjunction with higher level protocols to help manage data transmission services including Trivial File Transfer Protocol (TFTP), Real Time Streaming Protocol (RTSP), Simple Network Protocol (SNP) and domain name system (DNS) lookups.

User datagram protocol features

The user datagram protocol has attributes that make it advantageous for use with applications that can tolerate lost data.

- It allows packets to be dropped and received in a different order than they were transmitted, making it suitable for real-time applications where latency might be a concern.
- It can be used for transaction-based protocols, such as DNS or Network Time Protocol.
- It can be used where a large number of clients are connected and where real-time error correction isn't necessary, such as gaming, voice or video conferencing, and streaming media.

UDP header composition

The User Datagram Protocol header has four fields, each of which is 2 bytes. They are:

- source port number, which is the number of the sender;
- destination port number, the port the datagram is addressed to;
- length, the length in bytes of the UDP header and any encapsulated data; and
- checksum, which is used in error checking. Its use is required in IPv6 and optional in IPv4.

TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules defining the Internet.

TCP is a connection-oriented protocol, which means a connection is established and maintained until the application programs at each end have finished exchanging messages. It determines how to break application data into packets that networks can deliver, sends packets to and accepts packets from the network layer, manages flow control, and—because it is meant to provide errorfree data transmission—handles retransmission of dropped or garbled packets as well as acknowledgement of all packets that arrive. In the Open Systems Interconnection(OSI) communication model, TCP covers parts of Layer 4, the Transport Layer, and parts of Layer 5, the Session Layer.

For example, when a Web server sends an HTMLfile to a client, it uses the HTTP protocol to do so. The HTTP program layer asks the TCP layer to set up the connection and send the file. The TCP stack divides the file into packets, numbers them and then forwards them individually to the IP layer for delivery. Although each packet in the transmission will have the same source and destination IP addresses, packets may be sent along multiple routes. The TCP program layer in the

client computer waits until all of the packets have arrived, then acknowledges those it receives and asks for the retransmission on any it does not (based on missing packet numbers), then assembles them into a file and delivers the file to the receiving application.

Retransmissions and the need to reorder packets after they arrive can introduce latency in a TCP stream. Highly time-sensitive applications like voice over IP (VoIP) and streaming video generally rely on a transport like User Datagram Protocol (UDP) that reduces latency and jitter (variation in latency) by not worrying about reordering packets or getting missing data retransmitted.

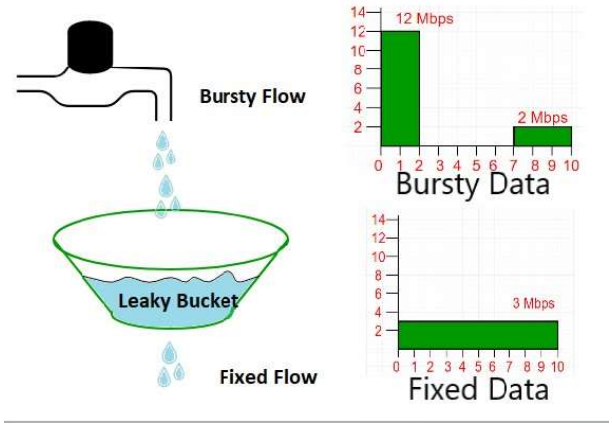
Leaky Bucket Algorithm

The leaky bucket algorithm is a method of temporarily storing a variable number of requests and organizing them into a set-rate output of packets in an asynchronous transfer mode (ATM) network.

The leaky bucket is used to implement traffic policing and traffic shaping in Ethernet and cellular data networks. The algorithm can also be used to control metered-bandwidth Internet connections to prevent going over the allotted bandwidth for a month, thereby avoiding extra charges.

The algorithm works similarly to the way an actual leaky bucket holds water: The leaky bucket takes data and collects it up to a maximum capacity. Data in the bucket is only released from the bucket at a set rate and size of packet. When the bucket runs out of data, the leaking stops. If incoming data would overfill the bucket, then the packet is considered to be non-conformant and is not added to the bucket. Data is added to the bucket as space becomes available for conforming packets.

The leaky bucket algorithm can also detect both gradually increasing and dramatic memory error increases by comparing how the average and peak data rates exceed set acceptable background amounts.



TOKEN BUCKET ALGORITHM :

Token Bucket Concepts

When you apply traffic policing to the input or output traffic at an interface, the rate limits and actions specified in the policer configuration are used to enforce a limit on the average throughput rate at the interface while also allowing bursts of traffic up to a maximum number of bytes based on the overall traffic load. Junos OS policers measure traffic-flow conformance to a policing rate limit by using a *token bucket algorithm*. An algorithm based on a single token bucket allows burst of traffic for short periods, whereas an algorithm based dual token buckets allows more sustained bursts of traffic.

Single Token Bucket Algorithm

A single-rate two-color policer limits traffic throughput at an interface based on how the traffic conforms to rate-limit values specified in the policer configuration. Similarly, a hierarchical policer limits traffic throughput at an interface based on how aggregate and premium traffic subflows conform to aggregate and premium rate-limit values specified in the policer configuration. For both two-color policer types, packets in a conforming traffic flow are categorized as *green*, and packets in a non-conforming traffic flow are categorized as *red*.

The single token bucket algorithm measures traffic-flow conformance to a two-color policer rate limit as follows:

- The token arrival rate represents the single *bandwidth limit* configured for the policer. You can specify the bandwidth limit as an absolute number of bits per second by including the bandwidth-limit *bps* statement. Alternatively, for single-rate two-color policers only, you can use the bandwidth-percent *percentage* statement to specify the bandwidth limit as a percentage of either the physical interface port speed or the configured logical interface shaping rate.

- The token bucket depth represents the single *burst size* configured for the policer. You specify the burst size by including the burst-size-limit *bytes* statement.

- If the bucket is filled to capacity, arriving tokens ―overflow‖ the bucket and are lost.

When the bucket contains insufficient tokens for receiving or transmitting the traffic at the interface, packets might be dropped or else re-marked with a lower forwarding class, a higher packet loss priority (PLP) level, or both.

Conformance Measurement for Two-Color Marking

In two-color-marking policing, a traffic flow whose average arrival or departure rate does not exceed the token arrival rate (bandwidth limit) is considered *conforming traffic*. Packets in a conforming traffic flow (categorized as green traffic) are implicitly marked with a packet loss priority (PLP) level of low and then passed through the interface.

For a traffic flow whose average arrival or departure rate exceeds the token arrival rate, conformance to a two-color policer rate limit depends on the tokens in the bucket. If sufficient tokens remain in the bucket, the flow is considered conforming traffic. If the bucket does not contain sufficient tokens, the flow is considered *non-conforming traffic*. Packets in a nonconforming traffic flow (categorized as red traffic) are handled according to policing actions. Depending on the configuration of the two-color policer, packets might be implicitly discarded; or the packets might be re-marked with a specified forwarding class, a specified PLP, or both, and then passed through the interface.

The token bucket is initially filled to capacity, and so the policer allows an initial traffic burst (back-to-back traffic at average rates that exceed the token arrival rate) up to the size of the token bucket depth.

During periods of relatively low traffic (traffic that arrives at or departs from the interface at average rates below the token arrival rate), unused tokens accumulate in the bucket, but only up to the configured token bucket depth.

Questions:

1. What are the advantages of IPv6 ?

2. Explain the working principle of leaky bucket algorithm.

3. Explain the working principle of Dijkastra algorithm.

MODULE – IV

APPLICATION LAYER

DOMAIN NAME SYSTEM

What is DNS?

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as

192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

How does DNS work?

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machinefriendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it‘s important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs ― behind the scenes‖ and requires no interaction from the user‘s computer apart from the initial request.

There are 4 DNS servers involved in loading a webpage:

- DNS recursor - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the recursor is then responsible for making additional requests in order to satisfy the client‘s DNS query.

- **Root nameserver** - The root server is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.

- TLD nameserver - The top level domain server (TLD) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is ―com‖).

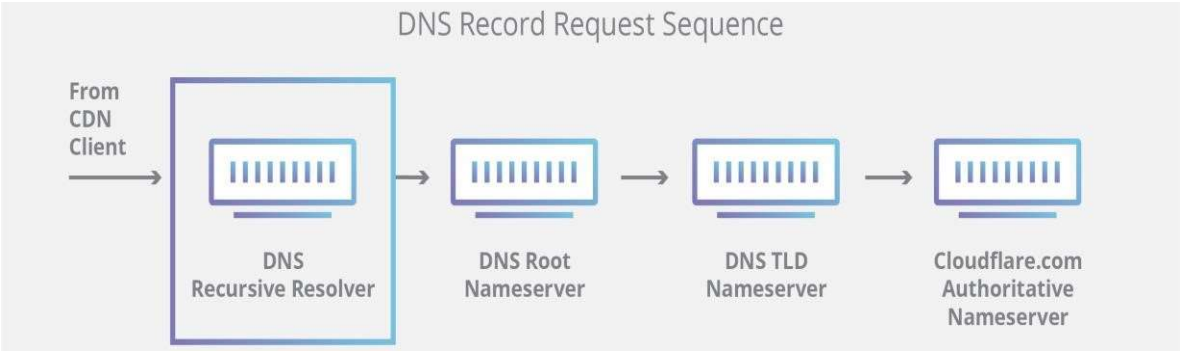
- Authoritative nameserver - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

What's the difference between an authoritative DNS server and a recursive DNS resolver?

Both concepts refer to servers (groups of servers) that are integral to the DNS infrastructure, but each performs a different role and lives in different locations inside the pipeline of a DNS query. One way to think about the difference is the recursive resolver is at the beginning of the DNS query and the authoritative nameserver is at the end.

Recursive DNS resolver

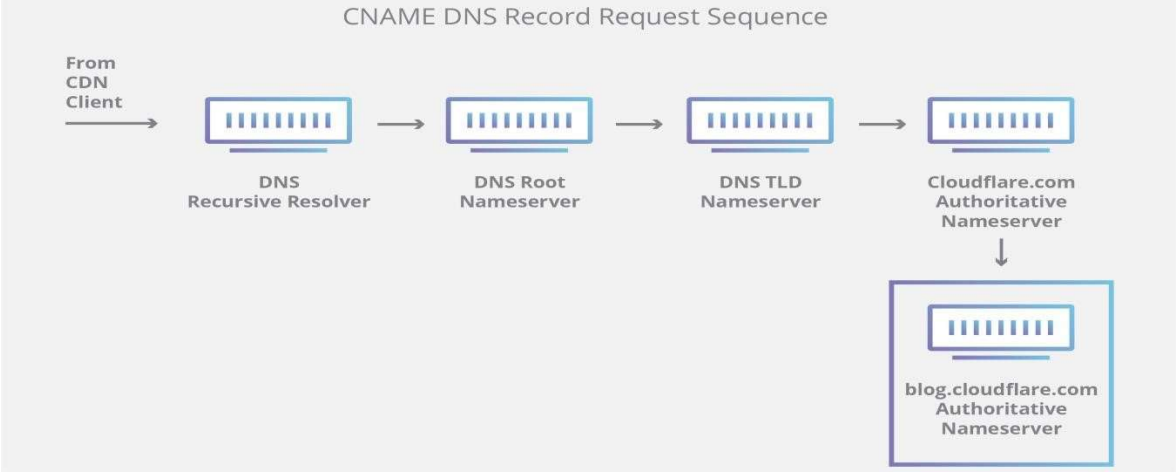
The recursive resolver is the computer that responds to a recursive request from a client and takes the time to track down the DNS record. It does this by making a series of requests until it reaches the authoritative DNS nameserver for the requested record (or times out or returns an error if no record is found). Luckily, recursive DNS resolvers do not always need to make multiple requests in order to track down the records needed to respond to a client; caching is a data persistence process that helps short-circuit the necessary requests by serving the requested resource record earlier in the DNS lookup.



Authoritative DNS server

Put simply, an authoritative DNS server is a server that actually holds, and is responsible for, DNS resource records. This is the server at the bottom of the DNS lookup chain that will respond with the queried resource record, ultimately allowing the web browser making the request to reach the IP address needed to access a website or other web resources. An authoritative nameserver can satisfy queries from its own data without needing to query another source, as it is the final source of truth for certain DNS records.

It‘s worth mentioning that in instances where the query is for a subdomain such as foo.example.com or blog.cloudflare.com, an additional nameserver will be added to the sequence after the authoritative nameserver, which is responsible for storing the subdomain‘s CNAME record.



There is a key difference between many DNS services and the one that Cloudflare provides. Different DNS recursive resolvers such as Google DNS, OpenDNS, and providers like Comcast all maintain data center installations of DNS recursive resolvers. These resolvers allow for quick

and easy queries through optimized clusters of DNS-optimized computer systems, but they are fundamentally different than the nameservers hosted by Cloudflare.

Cloudflare maintains infrastructure-level nameservers that are integral to the functioning of the Internet. One key example is the f-root server network which Cloudflare is partially responsible for hosting. The F-root is one of the root level DNS nameserver infrastructure components responsible for the billions of Internet requests per day. Our Anycast network puts us in a unique position to handle large volumes of DNS traffic without service interruption.

What are the steps in a DNS lookup?

For most situations, DNS is concerned with a domain name being translated into the appropriate IP address. To learn how this process works, it helps to follow the path of a DNS lookup as it travels from a web browser, through the DNS lookup process, and back again. Let's take a look at the steps.

Note: Often DNS lookup information will be cached either locally inside the querying computer or remotely in the DNS infrastructure. There are typically 8 steps in a DNS lookup. When DNS information is cached, steps are skipped from the DNS lookup process which makes it quicker. The example below outlines all 8 steps when nothing is cached.

The 8 steps in a DNS lookup:

1. A user types _example.com‘ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.

2. The resolver then queries a DNS root nameserver (.).

3. The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.

4. The resolver then makes a request to the .com TLD.

5. The TLD server then responds with the IP address of the domain‘s nameserver, example.com.

6. Lastly, the recursive resolver sends a query to the domain‘s nameserver.

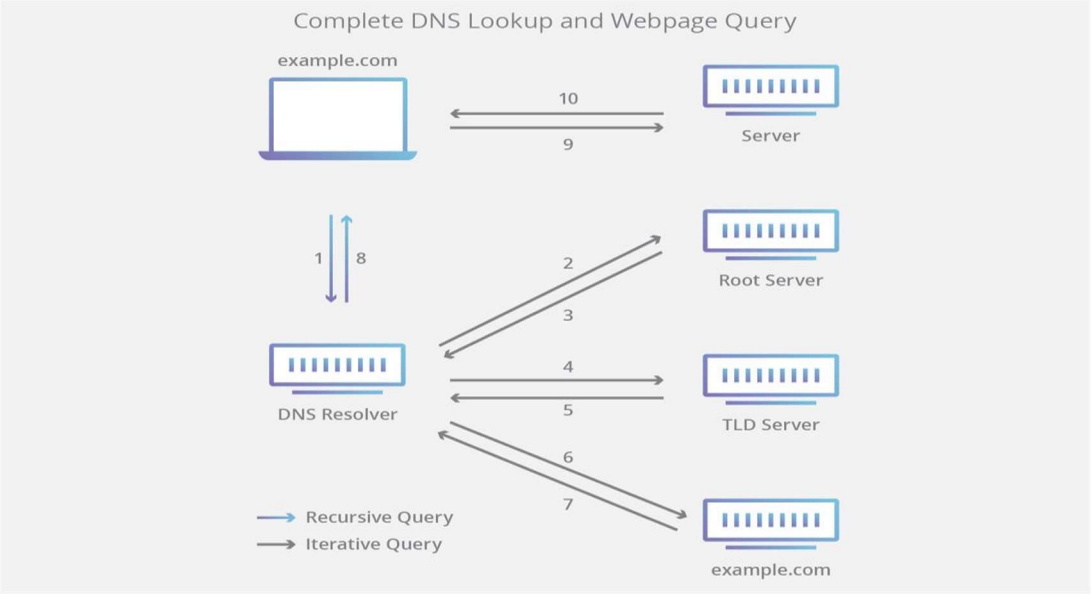
7. The IP address for example.com is then returned to the resolver from the nameserver.

8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

   Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

9. The browser makes a HTTP request to the IP address.

10. The server at that IP returns the webpage to be rendered in the browser (step 10).



What is a DNS resolver?

The DNS resolver is the first stop in the DNS lookup, and it is responsible for dealing with the client that made the initial request. The resolver starts the sequence of queries that ultimately leads to a URL being translated into the necessary IP address.

Note: A typical uncached DNS lookup will involve both recursive and iterative queries.

It's important to differentiate between a recursive DNS query and a recursive DNS resolver. The query refers to the request made to a DNS resolver requiring the resolution of the query. A DNS recursive resolver is the computer that accepts a recursive query and processes the response by making the necessary requests.

What are the types of DNS Queries?

In a typical DNS lookup three types of queries occur. By using a combination of these queries, an optimized process for DNS resolution can result in a reduction of distance traveled. In an ideal situation cached record data will be available, allowing a DNS name server to return a nonrecursive query.

3 types of DNS queries:

1. **Recursive query** - In a recursive query, a DNS client requires that a DNS server (typically a DNS recursive resolver) will respond to the client with either the requested resource record or an error message if the resolver can't find the record.

2. **Iterative query** - in this situation the DNS client will allow a DNS server to return the best answer it can. If the queried DNS server does not have a match for the query name, it will return a referral to a DNS server authoritative for a lower level of the domain namespace. The DNS client will then make a query to the referral address. This process continues with additional DNS servers down the query chain until either an error or timeout occurs.

3. **Non-recursive query** - typically this will occur when a DNS resolver client queries a DNS server for a record that it has access to either because it's authoritative for the record or the record exists inside of its cache. Typically, a DNS server will cache DNS records to prevent additional bandwidth consumption and load on upstream servers.

What is DNS caching? Where does DNS caching occur?

The purpose of caching is to temporarily stored data in a location that results in improvements in performance and reliability for data requests. DNS caching involves storing data closer to the requesting client so that the DNS query can be resolved earlier and additional queries further down the DNS lookup chain can be avoided, thereby improving load times and reducing bandwidth/CPU consumption. DNS data can be cached in a variety of locations, each of which will store DNS records for a set amount of time determined by a time-to-live (TTL).

Browser DNS caching

Modern web browsers are designed by default to cache DNS records for a set amount of time. the purpose here is obvious; the closer the DNS caching occurs to the web browser, the fewer processing steps must be taken in order to check the cache and make the correct requests to an IP address. When a request is made for a DNS record, the browser cache is the first location checked for the requested record.

In chrome, you can see the status of your DNS cache by going to chrome://net-internals/#dns.

Operating system (OS) level DNS caching

The operating system level DNS resolver is the second and last local stop before a DNS query leaves your machine. The process inside your operating system that is designed to handle this query is commonly called a ―stub resolver‖ or DNS client. When a stub resolver gets a request from an application, it first checks its own cache to see if it has the record. If it does not, it then sends a DNS query (with a recursive flag set), outside the local network to a DNS recursive resolver inside the Internet service provider (ISP).

Recursive resolver DNS caching

When the recursive resolver inside the ISP receives a DNS query, like all previous steps, it will also check to see if the requested host-to-IP-address translation is already stored inside its local persistence layer.

The recursive resolver also has additional functionality depending on the types of records it has in its cache:

1. If the resolver does not have the A records, but does have the NS records for the authoritative nameservers, it will query those name servers directly, bypassing several steps in the DNS query. This shortcut prevents lookups from the root and .com nameservers (in our search for example.com) and helps the resolution of the DNS query occur more quickly.

2. If the resolver does not have the NS records, it will send a query to the TLD servers (.com in our case), skipping the root server.

3. In the unlikely event that the resolver does not have records pointing to the TLD servers, it will then query the root servers. This event typically occurs after a DNS cache has been purged.

Simple Mail Transfer Protocol (SMTP)

Email is emerging as the one of the most valuable service in internet today. Most of the internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver‘s side.

SMTP

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection on that port (25). After successfully establishing the TCP connection the client process sends the mail instantly.

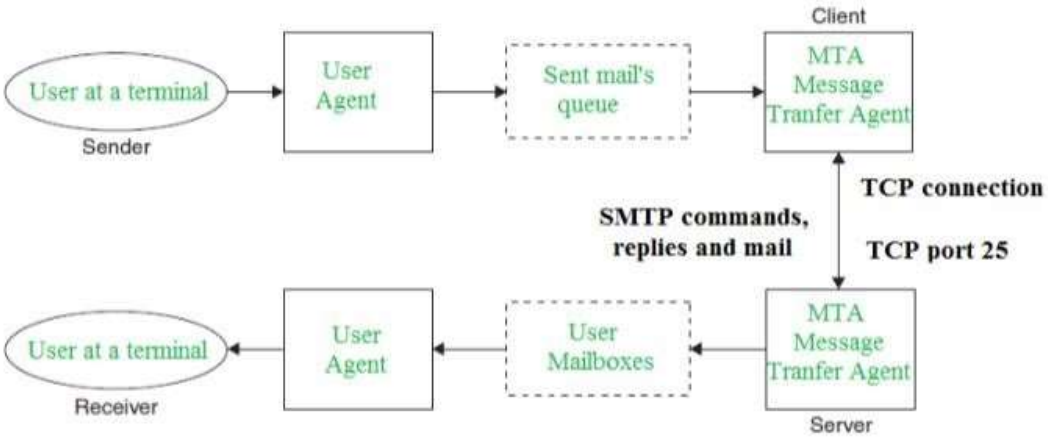
SMTP Protocol

The SMTP model is of two type :

1. End-to- end method
2. Store-and- forward method

The end to end model is used to communicate between different organizations whereas the store and forward method is used within an organization. A SMTP client who wants to send the mail will contact the destination‘s host SMTP directly in order to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver‘s SMTP. The client SMTP is the one which initiates the session let us call it as client- SMTP and the

server SMTP is the one which responds to the session request and let us call it as receiver-SMTP. The client- SMTP will start the session and the receiver-SMTP will respond to the request.



HTTP (Hypertext Transfer Protocol)

HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. As soon as a Web user opens their Web browser, the user is indirectly making use of HTTP. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols (the foundation protocols for the Internet). HTTP concepts include (as the Hypertext part of the name implies) the idea that files can contain references to other files whose selection will elicit additional transfer requests. Any server machine contains, in addition to the Web page files it can serve, an HTTP daemon, a program that is designed to wait for HTTP requests and handle them when they arrive. Your Web browser is an HTTP client, sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing in a Uniform Resource Locator or URL) or clicking on a hypertext link, the browser builds an HTTP request and sends it to the Internet Protocol address (IP address) indicated by the URL. The HTTP daemon in the destination server machine receives the request and sends back the requested file or files associated with the request. (A Web page often consists of more than one file.)

WWW (World Wide Web)

The **World Wide Web** (**WWW**), also called **the Web**, is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet.[1] English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser in 1990 while employed at CERN near Geneva, Switzerland. The browser was released outside CERN in 1991, first to other research institutions starting in January 1991 and to the general public on the Internet in August 1991.

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet. Web pages are primarily text documents formatted and annotated with Hypertext Markup Language (HTML). In addition to

formatted text, web pages may contain images, video, audio, and software components that are rendered in the user's web browser as coherent pages of multimedia content.

Embedded hyperlinks permit users to navigate between web pages. Multiple web pages with a common theme, a common domain name, or both, make up a website. Website content can largely be provided by the publisher, or interactively where users contribute content or the content depends upon the users or their actions. Websites may be mostly informative, primarily for entertainment, or largely for commercial, governmental, or non-governmental organizational purpose.

The terms Internet and World Wide Web are often used without much distinction. However, the two are not the same. The Internet is a global system of interconnected computer networks. In contrast, the World Wide Web is a global collection of documents and other resources, linked by hyperlinks and URIs. Web resources are accessed using HTTP or HTTPS, which are applicationlevel Internet protocols that use the Internet's transport protocols.

Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser, or by following a hyperlink to that page or resource. The web browser then initiates a series of background communication messages to fetch and display the requested page. In the 1990s, using a browser to view web pages—and to move from one web page to another through hyperlinks — came to be known as 'browsing,' 'web surfing' (after channel surfing), or 'navigating the Web'. Early studies of this new behavior investigated user patterns in using web browsers.

PUBLIC-KEY CRYPTOGRAPHY

Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses two mathematically related, but not identical, keys - a public key and a private key. Unlike symmetric

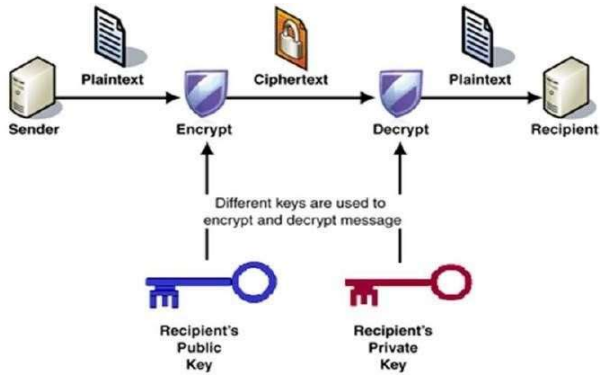
key algorithms that rely on one key to both encrypt and decrypt, each key performs a unique function. The public key is used to encrypt and the private key is used to decrypt.

It is computationally infeasible to compute the private key based on the public key. Because of this, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures, and private keys can be kept secret, and ensuring only the owners of the private keys can decrypt content and create digital signatures.

Since public keys need to be shared but are too big to be easily remembered, they are stored on digital certificates for secure transport and sharing. Since private keys are not shared, they are simply stored in the software or operating system you use, or on hardware (e.g., USB token, hardware security module) containing drivers that allow it to be used with your software or operating system.

Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.

With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems. The process of encryption and decryption is depicted in the following illustration −



The most important properties of public key encryption scheme are −

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.

- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

RSA Cryptosystem

This cryptosystem is one the initial system. It remains most employed cryptosystem even today. The system was invented by three scholars **Ron Rivest, Adi Shamir,** and **Len Adleman** and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below − **Generate the RSA modulus (n)**

- Select two large primes, p and q.
- Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

- **Find Derived Number (e)** ○ Number **e** must be greater than 1 and less than (p − 1)(q − 1). ○ There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.
- **Form the public key** ○ The pair of numbers (n, e) form the RSA public key and is made public. ○ Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.

Generate the private key

- Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
- Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1). ○ This relationship is written mathematically as follows −

$$ed = 1 \bmod (p - 1)(q - 1)$$

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

Example

An example of generating RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factor of 5 and (p − 1)(q − 1) = 6 × 12 = 72, except for 1.
- The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.
- Check that the d calculated is correct by computing −

```
de = 29 × 5 = 145 = 1 mod 72
```

- Hence, public key is (91, 5) and private keys is (91, 29).

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is (n, e).
- The sender then represents the plaintext as a series of numbers less than n.
- To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$$C = P^e \bmod n$$

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.
- Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

$$C = 10^5 \bmod 91$$

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C.
- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

$$\text{Plaintext} = C^d \bmod n$$

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

$$\text{Plaintext} = 82^{29} \bmod 91 = 10$$

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- **Encryption Function** − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.
- **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA

public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

4	81	1

- **Choosing the private key.** The private key x is any number bigger than 1 and smaller than p−1.
- **Computing part of the public key.** The value y is computed from the parameters p, g and the private key x as follows –

$$y = g^x \bmod p$$

- **Obtaining Public key.** The ElGamal public key consists of the three parameters (p, g, y).

  For example, suppose that p = 17 and that g = 6 (It can be confirmed that 6 is a generator of group Z_{17}). The private key x can be any number bigger than 1 and smaller than 71, so we choose x = 5. The value y is then computed as follows −

$$y = 6^5 \bmod 17 = 7$$

 Thus the private key is 62 and the public key is (17, 6, 7)

Digital Signature :

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. The digital equivalent of a handwritten signature or stamped seal, a digital signature offers far more inherent security, and it is intended to solve the problem of tampering and impersonation in digital communications.

Digital signatures can provide the added assurances of evidence of origin, identity and status of an electronic document, transaction or message and can acknowledge informed consent by the signer.

In many countries, including the United States, digital signatures are considered legally binding in the same way as traditional document signatures. The United States Government Publishing Office publishes electronic versions of the budget, public and private laws, and congressional bills with digital signatures.

How digital signatures work

Digital signatures are based on public key cryptography, also known as asymmetric cryptography. Using a public key algorithm, such as RSA, one can generate two keys that are mathematically linked: one private and one public.

Digital signatures work because public key cryptography depends on two mutually authenticating cryptographic keys. The individual who is creating the digital signature uses their own private key to encrypt signature-related data; the only way to decrypt that data is with the signer's public key. This is how digital signatures are authenticated.

Digital signature technology requires all the parties to trust that the individual creating the signature has been able to keep their own private key secret. If someone else has access to the signer's private key, that party could create fraudulent digital signatures in the name of the private key holder.

How to create a digital signature

To create a digital signature, signing software -- such as an email program -- creates a one-way hash of the electronic data to be signed. The private key is then used to encrypt the hash. The encrypted hash -- along with other information, such as the hashing algorithm -- is the digital signature.

The reason for encrypting the hash instead of the entire message or document is that a hash function can convert an arbitrary input into a fixed length value, which is usually much shorter. This saves time as hashing is much faster than signing.

The value of a hash is unique to the hashed data. Any change in the data, even a change in a single character, will result in a different value. This attribute enables others to validate the integrity of the data by using the signer's public key to decrypt the hash.

If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn't changed since it was signed. If the two hashes don't match, the data has either been tampered with in some way -- integrity -- or the signature was created with a private key that doesn't correspond to the public key presented by the signer -- authentication.

A digital signature can be used with any kind of message -- whether it is encrypted or not -- simply so the receiver can be sure of the sender's identity and that the message arrived intact. Digital signatures make it difficult for the signer to deny having signed something -- assuming their private key has not been compromised -- as the digital signature is unique to both the document and the signer and it binds them together. This property is called nonrepudiation.

Digital signatures are not to be confused with digital certificates. A digital certificate, an electronic document that contains the digital signature of the issuing certificate authority, binds together a public key with an identity and can be used to verify that a public key belongs to a particular person or entity.

Most modern email programs support the use of digital signatures and digital certificates, making it easy to sign any outgoing emails and validate digitally signed incoming messages. Digital signatures are also used extensively to provide proof of authenticity, data integrity and nonrepudiation of communications and transactions conducted over the internet.

Digital signature vs. electronic signature

While *digital signature* is a technical term, defining the result of a cryptographic process that can be used to authenticate a sequence of data, the term *electronic signature* -- or *e-signature* -- is a legal term that is defined legislatively.

For example, in the United States, the term was defined in the Electronic Signatures in Global and National Commerce Act, passed in 2000, as meaning "an electronic sound, symbol, or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record."

This means that a digital signature -- which can be expressed digitally in electronic form and associated with the representation of a record -- can be a type of electronic signature. More generally, though, an electronic signature can be as simple as the signer's name being entered on a form on a webpage.

To be considered valid, electronic signature schemes must include three things:

- a way to verify the identity of the entity signing it;

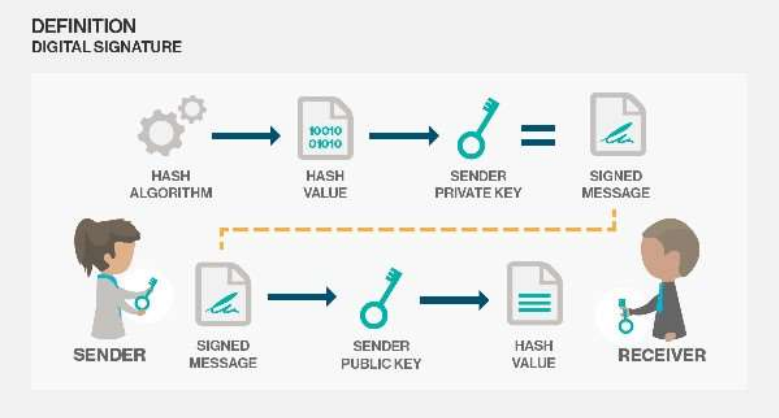
- a way to verify that the signing entity intended to affirm the document being signed; and

  a way to verify that the electronic signature is indeed associated with the signed document.

A digital signature can, on its own, fulfill these requirements to serve as an electronic signature:

- the public key of the digital signature is linked to the signing entity's identification;
- the digital signature can only be affixed by the holder of the public key's associated private key, which implies the entity intends to use it for the signature; and
- the digital signature will only authenticate if the signed data -- document or representation of a document -- is unchanged. If a document is altered after being signed, the digital signature will fail to authenticate.

While authenticated digital signatures provide cryptographic proof that a document was signed by the stated entity and that the document has not been altered, not all electronic signatures can provide the same guarantees.



FIREWALL :

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

Firewalls have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet.

A firewall can be hardware, software, or both.

Types of firewalls

Proxy Firewall

An early type of firewall device, a proxy firewall serves as the gateway from one network to another for a specific application. Proxy servers can provide additional functionality such as content caching and security by preventing direct connections from outside the network. However, this also may impact throughput capabilities and the applications they can support.

Stateful inspection firewall

Now thought of as a ―traditional‖ firewall, a stateful inspection firewall allows or blocks traffic based on state, port, and protocol. It monitors all activity from the opening of a connection until it is closed. Filtering decisions are made based on both administrator-defined rules as well as context, which refers to using information from previous connections and packets belonging to the same connection.

Unified threat management (UTM) firewall

A UTM device typically combines, in a loosely coupled way, the functions of a stateful inspection firewall with intrusion prevention and antivirus. It may also include additional services and often cloud management. UTMs focus on simplicity and ease of use.

See our UTM devices.

Next-generation firewall (NGFW)

Firewalls have evolved beyond simple packet filtering and stateful inspection. Most companies are deploying next-generation firewalls to block modern threats such as advanced malware and application-layer attacks.

According to Gartner, Inc.‘s definition, a next-generation firewall must include:

- Standard firewall capabilities like stateful inspection
- Integrated intrusion prevention
- Application awareness and control to see and block risky apps
- Upgrade paths to include future information feeds
- Techniques to address evolving security threats

While these capabilities are increasingly becoming the standard for most companies, NGFWs can do more.

Threat-focused NGFW

These firewalls include all the capabilities of a traditional NGFW and also provide advanced threat detection and remediation. With a threat-focused NGFW you can:

- Know which assets are most at risk with complete context awareness
- Quickly react to attacks with intelligent security automation that sets policies and hardens your defenses dynamically
- Better detect evasive or suspicious activity with network and endpoint event correlation
- Greatly decrease the time from detection to cleanup with retrospective security that continuously monitors for suspicious activity and behavior even after initial inspection
- Ease administration and reduce complexity with unified policies that protect across the entire attack continuum

Broadly speaking, a computer firewall is a software program that prevents unauthorized access to or from a private network. Firewalls are tools that can be used to enhance the security of computers

connected to a network, such as LAN or the Internet. They are an integral part of a comprehensive security framework for your network.

A firewall absolutely isolates your computer from the Internet using a "wall of code" that inspects each individual "packet" of data as it arrives at either side of the firewall — inbound to or outbound from your computer — to determine whether it should be allowed to pass or be blocked.

Firewalls have the ability to further enhance security by enabling granular control over what types of system functions and processes have access to networking resources. These firewalls can use various types of signatures and host conditions to allow or deny traffic. Although they sound complex, firewalls are relatively easy to install, setup and operate.

Most people think that a firewall is a of device that is installed on the network, and it controls the traffic that passes through the network segment.

However, you can have a host-based firewalls. This can be executed on the systems themselves, such as with ICF (Internet Connection Firewall). Basically, the work of both the firewalls is the same: to stop intrusion and provide a strong method of access control policy. In simple definition, firewalls are nothing but a system that safeguards your computer; access control policy enforcement points.

What Firewalls Do?

Basically, firewalls need to be able to perform the following tasks:

- Defend resources
- Validate access
- Manage and control network traffic
- Record and report on events
- Act as an intermediary

What is Personal Firewall

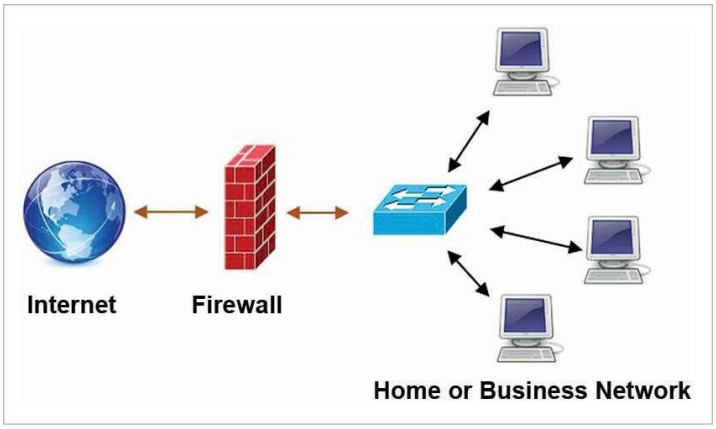
It is important to understand why we need a firewall and how it helps us in the world of secure computing. We need to understand the goals of information security because it helps us to understand how a firewall may address those needs.

Why you need Personal Firewall

In the age of high-speed Internet Access, you electronically connect your computer to a broad network over which, unless you have installed a personal firewall, you have limited control and from which you have limited protection. Until recently, unless you worked for an organization that provided high-speed internet access.

Like anything, the high-speed connection has its own drawbacks. Ironically, the very feature that makes a high-speed connection attractive is also the reason that makes it vulnerable. In a way, connecting to the internet via high-speed connection is like leaving the front door of your house open and unlocked. This is because high-speed Internet connections have the following features:

- A constant IP - Make it easy for an intruder who has discovered your computer on the internet to find you again and again.
- High-Speed Access - Means that the intruder can work much faster when trying to break into your computer.
- Always active connection - means that your computer is vulnerable every time when it is connected to the internet.



Asynchronous Transfer Mode (ATM) :

Asynchronous transfer mode (ATM) is a switching technique used by telecommunication networks that uses asynchronous time-division multiplexing to encode data into small, fixedsized cells. This is different from Ethernet or internet, which use variable packet sizes for data or frames. ATM is the core protocol used over the synchronous optical network (SONET) backbone of the integrated digital services network (ISDN).

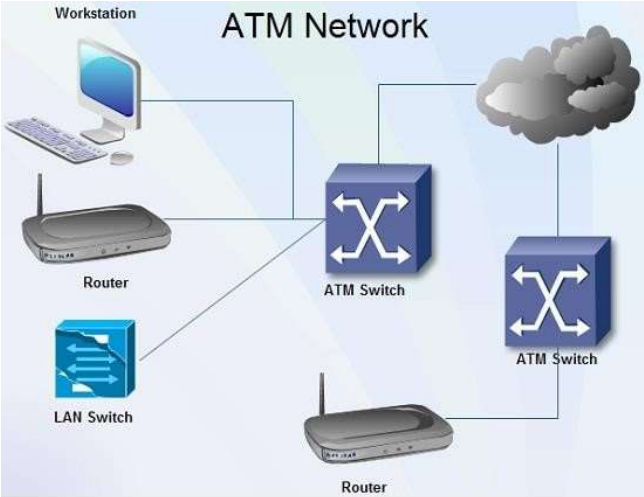
Asynchronous transfer mode was designed with cells in mind. This is because voice data is converted to packets and is forced to share a network with burst data (large packet data) passing through the same medium. So, no matter how small the voice packets are, they always encounter full-sized data packets, and could experience maximum queuing delays. This is why all data packets should be of the same size. The fixed cell structure of ATM means it can be easily switched by hardware without the delays introduced by routed frames and software switching. This is why some people believe that ATM is the key to the internet bandwidth problem. ATM creates fixed routes between two points before data transfer begins, which differs from TCP/IP, where data is divided into packets, each of which takes a different route to get to its destination. This makes it easier to bill data usage. However, an ATM network is less adaptable to a sudden network traffic surge.

The ATM provides data link layer services that run on the OSI's Layer 1 physical links. It functions much like small-packet switched and circuit-switched networks, which makes it ideal for real-rime, low-latency data such as VoIP and video, as well as for high-throughput data traffic like file transfers. A virtual circuit or connection must be established before the two end points can actually exchange data.

ATM services generally have four different bit rate choices:

- Available Bit Rate: Provides a guaranteed minimum capacity but data can be bursted to higher capacities when network traffic is minimal.

- Constant Bit Rate: Specifies a fixed bit rate so that data is sent in a steady stream. This is analogous to a leased line.
- Unspecified Bit Rate: Doesn‘t guarantee any throughput level and is used for applications such as file transfers that can tolerate delays.
- Variable Bit Rate (VBR): Provides a specified throughput, but data is not sent evenly. This makes it a popular choice for voice and videoconferencing.



WLAN (Wireless Local Area Network)

A wireless LAN (or WLAN, for wireless local area network, sometimes referred to as LAWN, for local area wireless network) is one in which a mobile user can connect to a local area network (LAN) through a wireless (radio) connection.

A WLAN, or wireless LAN, is a network that allows devices to connect and communicate wirelessly. Unlike a traditional wired LAN, in which devices communicate over Ethernetcables, devices on a WLAN communicate via Wi-Fi.

While a WLAN may look different than a traditional LAN, it functions the same way. New devices are typically added and configured using DHCP. They can communicate with other devices on the network the same way they would on a wired network. The primary difference is how the data is transmitted. In a LAN, data is transmitted over physical cables in a series of Ethernet packets containing. In a WLAN, data is transmitted over the air using one of Wi-Fi 802.11 protocols.

As wireless devices have grown in popularity, so have WLANs. In fact, most routers sold are now wireless routers. A wireless router serves as a base station, providing wireless connections to any Wi-Fi-enabled devices within range of the router's wireless signal. This includes laptops, tablets, smartphones, and other wireless devices, such as smart appliances and smart home controllers. Wireless routers often connect to a cable modem or other Internetconnected device to provide Internet access to connected devices.

LANs and WLANs can be merged together using a bridge that connects the two networks. Many wireless routers also include Ethernet ports, providing connections for a limited number of wireless devices. In most cases, wireless routers act as a bridge, merging the Ethernet and Wi-Ficonnected devices into the same network. This allows wired and wireless devices to communicate with each other through a single router.

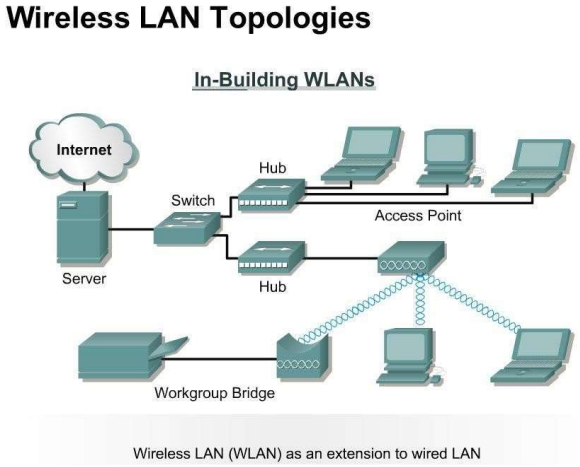
Advantages of WLANs

The most obvious advantage of a WLAN is that devices can connect wirelessly, eliminating the need for cables. This allows homes and businesses to create local networks without wiring the building with Ethernet. It also provides a way for small devices, such as smartphones and tablets, to connect to the network. WLANs are not limited by the number of physical ports on the router and therefore can support dozens or even hundreds of devices. The range of a WLAN can easily be extended by adding one or more repeaters. Finally, a WLAN can be easily upgraded by replacing routers with new versions — a much easier and cheaper solution than upgrading old Ethernet cables.

Disadvantages of WLANs

Wireless networks are naturally less secure than wired networks. Any wireless device can attempt to connect to a WLAN, so it is important to limit access to the network if security is a concern.

This is typically done using wireless authentication such as WEP or WPA, which encrypts the communication. Additionally, wireless networks are more susceptible to interference from other signals or physical barriers, such as concrete walls. Since LANs offer the highest performance and security, they are still used for many corporate and government networks.



Bluetooth

Bluetooth is a telecommunications industry specification that describes how mobile devices, computers and other devices can easily communicate with each other using a shortrange wireless connection.

What is Bluetooth used for?

Early Bluetooth versions allowed users of cellular phones, pagers and personal digital assistants to buy a three-in-one phone that could double as a portable phone at home or in the office, get quickly synchronized with information in a desktop or notebook computer, initiate the sending or receiving of a fax, initiate a printout, and, in general, have all mobile and fixed computer devices be totally coordinated over a short distance.

More recent Bluetooth versions make it possible for a user to place hands-free phone calls through a mobile phone or connect wireless headphones to a smartphone's music playlist, for example. Bluetooth technology can simplify tasks that previously involved copious wires strewn among peripheral devices. For instance, with a Bluetooth-enabled printer, one can connect wirelessly with a desktop, laptop or mobile device and print out documents. It is also possible to sync a wireless keyboard with a tablet-style device, such as an Apple iPad or Kindle Fire, or even a DVD player with a television.

Additionally, mobile operating systems allow users to stream media, such as movies, television shows and music, to compatible TVs, speakers and media players via Bluetooth. With an eye toward the future of Bluetooth, companies such as LG are manufacturing televisions with built-in Bluetooth technology that can display 3D images users view through special "active-shutter" glasses. Though this technology is in its formative stages, it's gotten an enthusiastic reception from gamers.

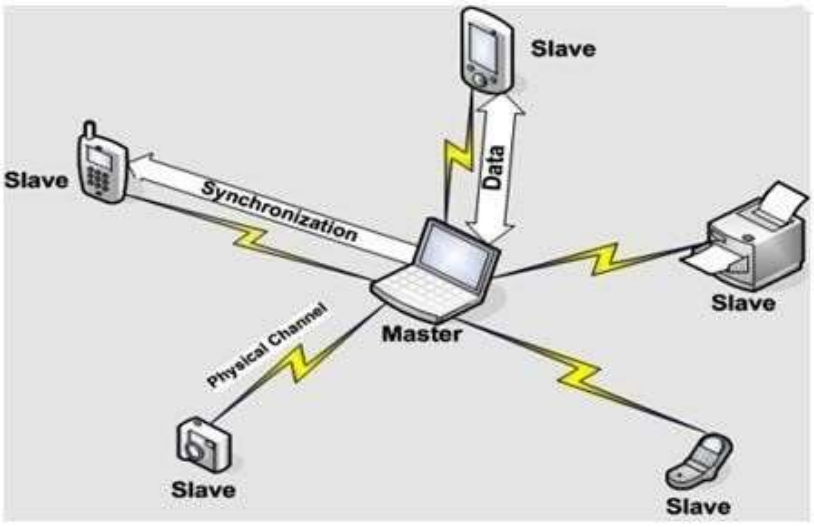
Laptop or desktop computers without built-in Bluetooth can gain those capabilities through an inexpensive USB dongle. The one caveat here is Bluetooth technology typically uses considerable battery power, so it's suggested that it be monitored closely by the user to prevent a device's battery from running down.

How does Bluetooth work?

Bluetooth technology requires that a low-cost transceiver chip be included in each device. The transceiver transmits and receives in a previously unused frequency band of 2.45 GHz that is available globally -- with some variation of bandwidth in different countries. In addition to data, up to three voice channels are available. Each device has a unique 48-bit address from the IEEE 802 standard. Bluetooth connections can be point to point or multipoint.

The maximum Bluetooth range is 10 meters. Data can be exchanged at a rate of 1 megabit per second -- up to 2 Mbps in the second generation of the technology. A frequency hop scheme allows devices to communicate even in areas with a great deal of electromagnetic interference.

Built-in encryption and verification is provided.

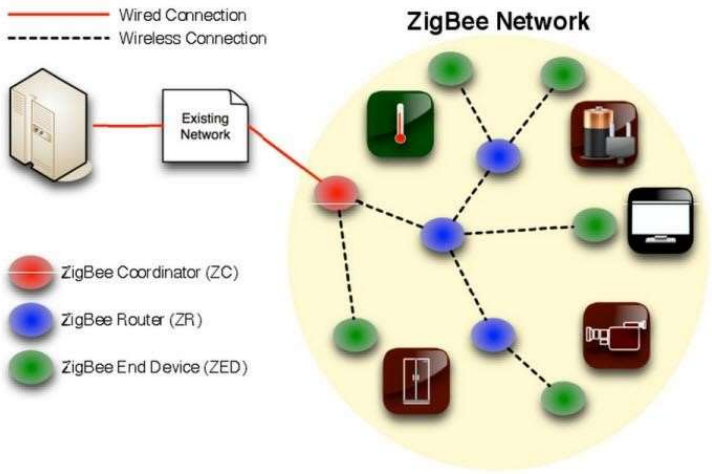


ZIGBEE

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network.

The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires shortrange low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10–100 meters line-of-sight, depending on power output and environmental characteristics. Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. Zigbee is typically used in low data rate applications that require long battery life and secure networking (Zigbee networks are secured by 128 bit symmetric encryption keys.) Zigbee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device.



Reference :

1. www. wikipedia.org

2. www.searchtelecom.techtarget.com

3. www.cloudflare.com

4. www.geeksforgeeks.org

5. www.di-mgt.com.au

Questions:

1. Explain RSA algorithm with example.

2. Explain the working principle of firewall.