**COMPUTER NETWORKS**

**Data Link Layer: -**

1. Node to node / Hop to Hop delivery.
2. **Flow control** - A receiving node can receive the frames at a faster rate than it can process the frame. Without flow control, the receiver's buffer can overflow, and frames can get lost. To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link.

* Stop and Wait Protocol
* Go Back N ARQ (Automatic Repeat Request)
* Selective Repeat

1. **Error Control -** Errors can be introduced by signal attenuation and noise. Data Link Layer protocol provides a mechanism to detect one or more errors.

* CRC (Cyclic Redundancy Check)
* CheckSum
* Parity Bit
* Hamming Code
* Bit Stuffing

1. **Access Control –** To handle collision when two computers want to send messages using the same channel and there is only one communication cable.

* CSMA/CA, CSMA/CD (Carrier Sense Multiple Access, CA – Collision Avoidance, CD – Collision Detection)
* Pure ALOHA, Slotted ALOHA
* Token ring
* Token bus

1. **Physical Address** – The address used to identify a device or computer within a network. Each device is connected to the internet using ethernet cable which connects to the NIC. Each NIC has a unique address which is the MAC address (12-digit, 48-bit, hexadecimal number).
2. **Framing –** Data link layer converts incoming packets from network layer into frames, which is a fixed format and reliable.

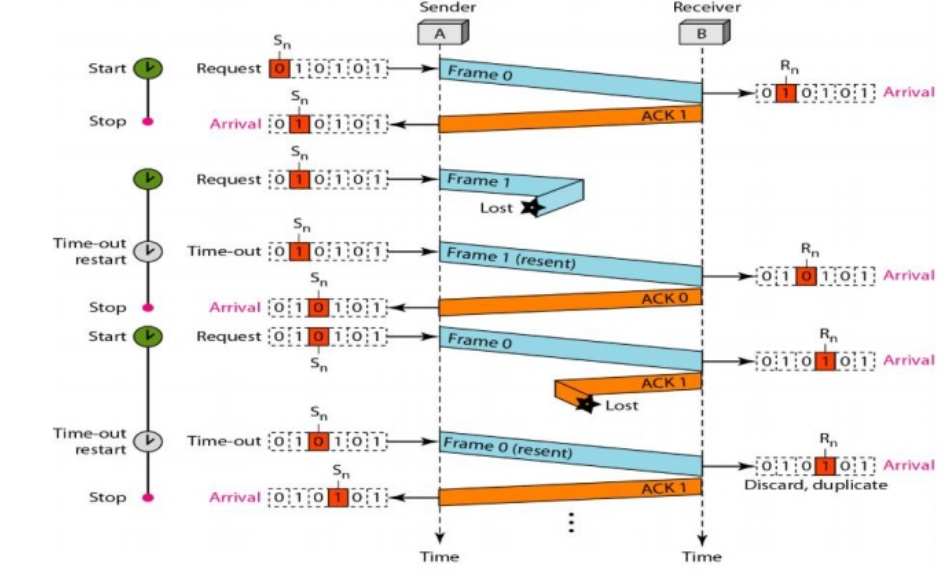
**Stop and Wait Protocol: -**

1. Used in Connection-oriented communication.
2. It offers error and flows control.
3. It is used in Data Link and Transport Layers.
4. Stop and Wait for ARQ mainly implements the Sliding Window Protocol concept with Window Size 1.

**Useful Terms:**

* **Propagation Delay**: Amount of time taken by a packet to make a physical journey from one router to another router.

Propagation Delay = (Distance between routers) / (Velocity of propagation)

* **RoundTripTime (RTT)** = 2\* Propagation Delay
* **TimeOut (TO)** = 2\* RTT
* **Time To Live (TTL)** = 2\* TimeOut. (The amount of time or “hops” that a packet is set to exist inside a network before being discarded. Maximum TTL is 255 seconds)

**Working of Stop and Wait for ARQ:**

1. Sender A sends a data frame or packet with sequence number 0.
2. Receiver B, after receiving the data frame, sends an acknowledgement with sequence number 1 (the sequence number of the next expected data frame or packet).

There is only a one-bit sequence number that implies that both sender and receiver have a buffer for one frame or packet only. Therefore, we need only 2-bit sequence, i.e., 0 & 1.

**Characteristics of Stop and Wait ARQ:**

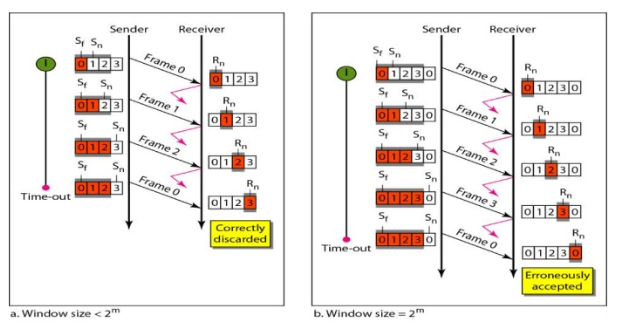
1. It uses a link between sender and receiver as a half-duplex link.
2. **Throughput = 1 Data packet/frame per RTT**
3. If the **Bandwidth\*Delay** product is very high, then stop and wait for protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
4. It is an example of “Closed Loop OR connection-oriented” protocols
5. It is a special category of SWP where its window size is 1
6. Irrespective of the number of packets sender is having stop and wait for protocol requires only 2 sequence numbers 0 and 1.

**Constraints: -**

1. Stop and Wait ARQ has very less efficiency, it can be improved by increasing the window size. For better efficiency, Go back N and Selective Repeat Protocols are used.
2. The Stop and Wait ARQ solves the main three problems but may cause big performance issues as the sender always waits for acknowledgement even if it has the next packet ready to send.

**Go Back N ARQ: -**

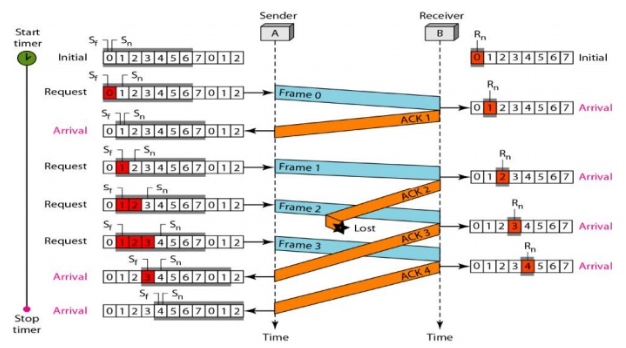
1. Uses protocol pipelining. Sender can send multiple frames before receiving the Ack for the 1st frame.
2. If Ack of a frame is not received within an agreed upon time period then all frames in the current window are transmitted.
3. Size of sending window determines sequence no. of outbound frames.
4. Bits Required in GBN**(m)** = **ceil(log2 (N + 1))**, where N = Sender’s window size.

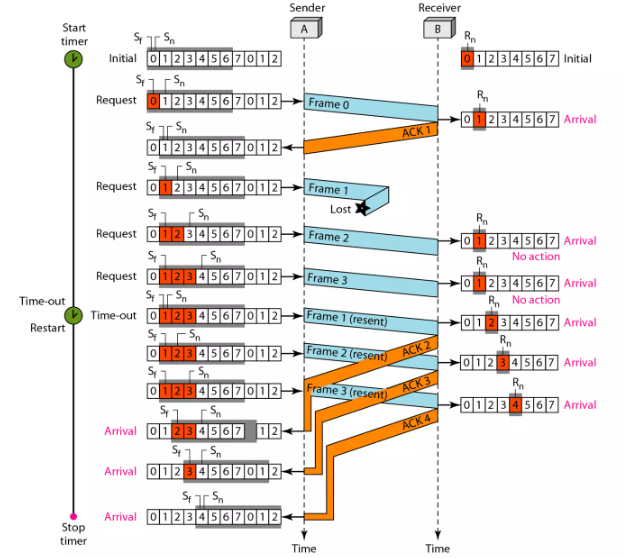
The extra 1 is required in order to avoid the problem of duplicate packets.

1. **Sender Window Size (WS):** It is N itself. If we say the protocol is GB(10), then Ws = 10. N should be always greater than 1 in order to implement pipelining. For N = 1, it reduces to Stop and Wait protocol.
2. **Receiver Window Size (WR):** WR is always 1 in GBN.
3. **Acknowledgements** There are 2 kinds of acknowledgements namely:

* **Cumulative Ack:** One acknowledgement is used for many packets. The main advantage is traffic is less. A disadvantage is less reliability as if one ack is the loss that would mean that all the packets sent are lost.
* **Independent Ack:** If every packet is going to get acknowledgement independently. Reliability is high here but a disadvantage is that traffic is also high since for every packet we are receiving independent ack.

If B is the bandwidth of the channel, then

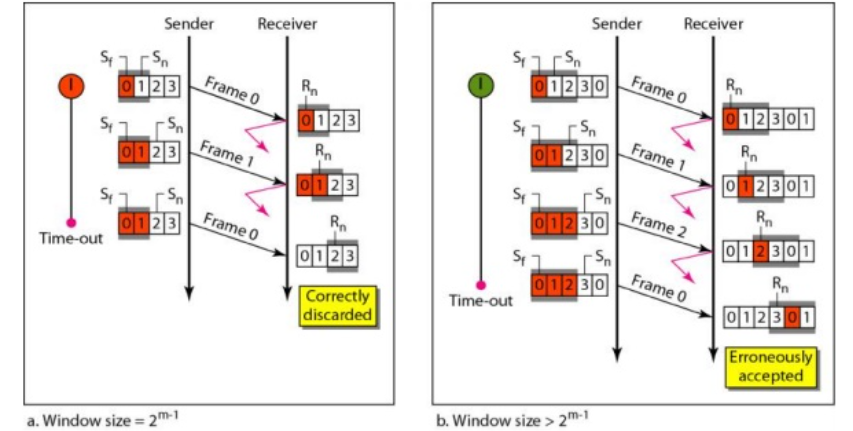
**The Ack 3 is a Cumulative Ack i.e., it is also Ack for 1 and 2. So even if Ack 2 is lost, Ack 3 does the job.**

****

**Since receiver window size is 1, when frame 1 is lost, receiver keeps waiting for frame 1. Sender sends till frame 3 but does not receive Ack for frame 1. So, it resends all frames in the window starting from frame 1 i.e., frames 1, 2 and 3. Thus disadvantage of GBN is that it cannot accept out of order packets.**

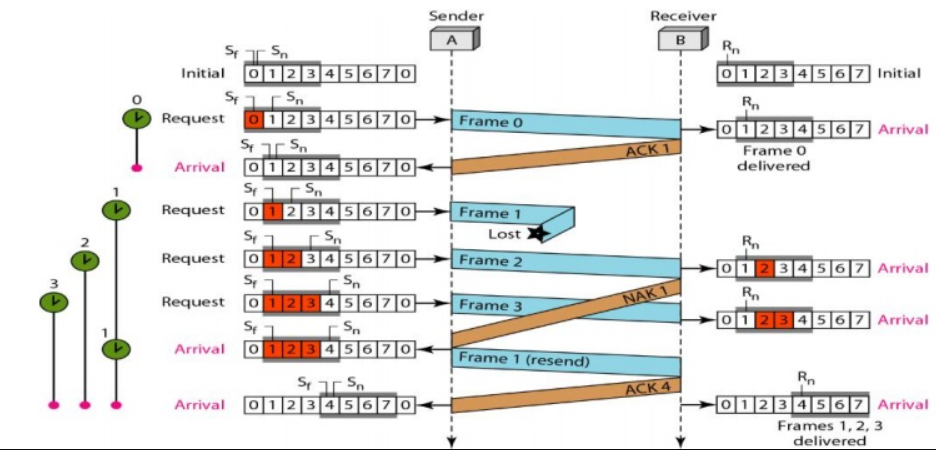
**Selective Repeat: -**

1. Uses two windows: a send window and a receive window.
2. Both sender and receiver window are of size . This is to avoid packets being recognized incorrectly. If the size of the window is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.



**Retransmission requests: -**

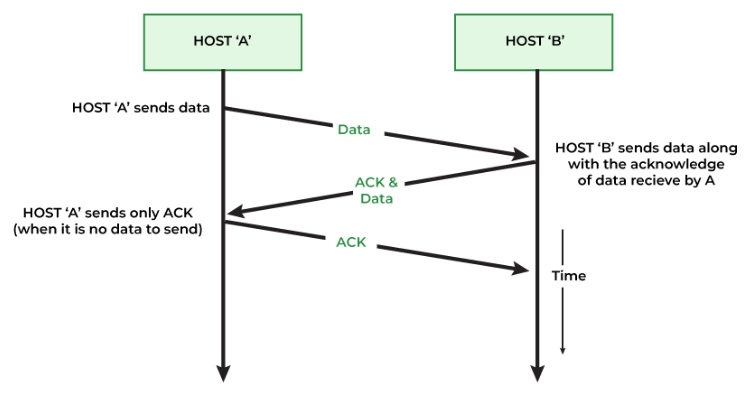
1. **Implicit –** The receiver acknowledges every good packet, packets that are not ACKed before a time-out are assumed lost or in error. Notice that this approach must be used to be sure that every packet is eventually received.
2. **Explicit –** An explicit NAK (selective reject) can request retransmission of just one packet. This approach can expedite the retransmission but is not strictly needed.
3. Sender can transmit new packets as long as their number is with W of all unACKed packets.
4. Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
5. Receiver ACKs all correct packets.
6. Receiver stores correct packets until they can be delivered in order to the higher layer.

**Since frame 2 is received successfully and frame 1 was not, the receiver sends a negative ack (NAK) just for packet 1. The sender then resends only packet 1. As in GBN, Ack 4 acts as a cumulative acknowledgement for 1, 2, 3 & 4.**

**Piggybacking: -**

1. The three protocols discussed are all unidirectional, i.e., data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction.
2. In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions.
3. There are two ways through which we can achieve full-duplex transmission:

* **Two Separate Channels:** One way to achieve full-duplex transmission is to have two separate channels with one for forwarding data transmission and the other for reverse data transfer (to accept). But this will almost completely waste the bandwidth of the reverse channel.
* **Piggybacking:** A preferable solution would be to use each channel to transmit the frame (front and back) both ways, with both channels having the same capacity. Assume that A and B are users. Then the data frames from A to B are interconnected with the acknowledgment from A to B and can be identified as a data frame or acknowledgment by checking the sort field in the header of the received frame.

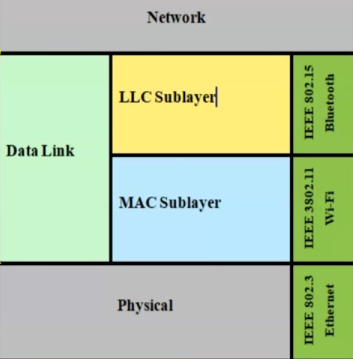


**Advantages: -**

1. The major advantage of piggybacking is the better use of available channel bandwidth. This happens because an acknowledgment frame needs not to be sent separately.
2. Usage cost reduction.
3. Improves latency of data transfer.
4. To avoid the delay and rebroadcast of frame transmission, piggybacking uses a very short-duration timer.

**Disadvantages: -**

1. The disadvantage of piggybacking is the additional complexity.
2. If the data link layer waits long before transmitting the acknowledgment (blocks the ACK for some time), the frame will rebroadcast.

**Sublayers of Data Link layer: -**

The Data Link layer consists of two sublayers: -

• The Logical Link Control (LLC) sublayer.

* The Media Access Control (MAC) sublayer

**Logical Link Control Sublayer: -**

1. Handles communication between upper and lower layers.
2. Takes the network protocol data and adds control information to help deliver the packet to the destination. **(Flow control)**
3. Performs Error checking
4. Performs multiplexing and demultiplexing - This allows multiple network protocols to exist within a single multiple access point network.

**MAC Sublayer: -**

1. Constitutes the lower sublayer of the data link layer.
2. Implemented by hardware, typically in the computer NIC.
3. Primary responsibilities:

* Framing
* Physical addressing or MAC Addressing

**Framing: -**

In the physical layer, data transmission involves synchronized transmission of bits from the source to the destination. The data link layer packs these bits into frames.

Data-link layer takes the packets from the Network Layer and encapsulates them into frames. If the frame size becomes too large, then the packet may be divided into small sized frames. Smaller sized frames make flow control and error control more efficient.

Types of framing: -

1. **Fixed size framing: -**
   1. Size of the frame is fixed and frame length acts as delimiter of the frame.
   2. Does not require additional boundary bits to identify start and end of frame.
2. **Variable sized framing: -**
3. Size of each frame transmitted may be different.
4. Additional mechanisms are kept to mark end of one frame and beginning of next frame.

Framing approaches: -

1. **Bit oriented approach –** Simply views the frame as a collection of bits. Bit oriented protocol – **HDLC (High- Level Data Link Control).**

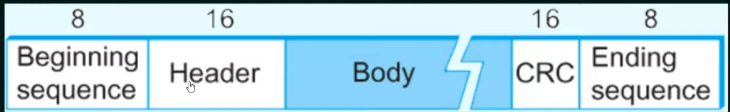
**HDLC: -**

i. The **Synchronous Data Link Control (SDLC)** protocol developed by IBM is an example of a bit-oriented protocol.

ii. SDLC was later standardized by the ISO as the High-Level Data Link

Control (HDLC) protocol.

**HDLC Frame format: -**

****

**Beginning and Ending Sequences** – 01111110. This sequence is also transmitted during any times that the link is idle so that the sender and receiver can keep their clocks synchronized.

**Header -** Address and Control Field.

**Body -** Payload (Variable size)

**CRC -** Cyclic Redundancy check - Error Detection

**Types of HDLC Frames: -**

Determined by the control field.

**I-Frame: Information Frame –** 1st bit is 0.

**S-Frame: Supervisory Frame –** 1st two bits is 10**.**

**U-Frame: Un-numbered Frame. –** 1st two bits is 11.

1. **Byte oriented approach –** Each frame is viewed as a collection of bytes (characters). Byte oriented protocols: -

**BISYNC - Binary Synchronous Communication Protocol.**

**DDCMP - Digital Data Communication Message Protocol.**

**PPP - Point-to-Point Protocol**

**Error Control: -**

There can be two types of error: -

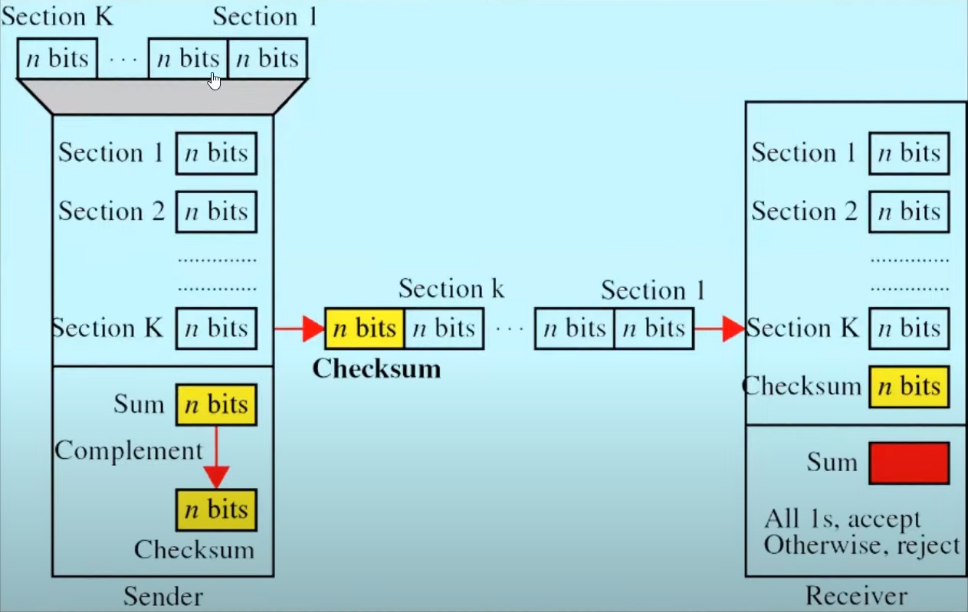
1. Bit Error – Only 1 bit in the data unit has been changed.
2. Burst Error – 2 or more bits in the data have been changed.

Error control can be done in two ways: -

1. **Error detection** − Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter. Techniques for Error detection – **Parity Bit Check, Checksum and Cyclic Redundancy Check (CRC).**
2. **Error correction** − Error correction involves ascertaining the exact number of bits that has been corrupted and the location of the corrupted bits. Techniques for Error correction – **Hamming Code**.

**Checksum: -**

**Sender side: -**

1. Break the original message in to 'k' number of blocks with 'n' bits in each block.
2. Sum all the k' data blocks.
3. Add the carry to the sum, if any.
4. Do 1's complement to the sum = Checksum.

**Receiver side: -**

1. Sum all the k data blocks including the checksum block.
2. If the result is all 1’s, ACCEPT; else REJECT.

**Performance of Checksum: -**

1. The checksum detects all errors involving an odd number of bits.
2. It detects most errors involving an even number of bits.
3. If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect the error (s).

If the data transmitted along with checksum is 10101001 00111001. But the data received at destination is **0**0101001 **1**0111001.

Receiver Site: -

00101001 1st bit of subunit 1 is damaged

10111001 1st bit of subunit 2 is damaged

00011101 checksum

11111111 sum

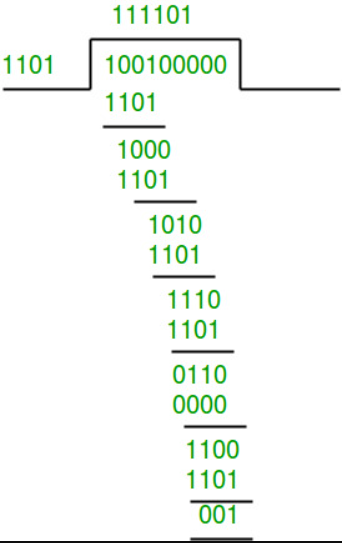
00000000 Ok 1's complement

Although data is corrupted, the error is undetected.

**CRC (Cyclic Redundancy Check): -**

**Sender side: -**

1. Find the length of the divisor 'L'.
2. Append 'L-1' bits to the original message. **((L – 1) no. of 0 bits)**
3. Perform binary division operation.
4. Remainder of the division = CRC.
5. Message to be transmitted = Message + CRC.



Here, Divisor = 1101

Message = 100100

Dividend = 100100 + 3 0’s = 100100000

Remainder = 001

Therefore,

CRC = 001

Transmitted message = 100100 append 001

= 100100001

A divisor might be given in the form of a polynomial. E.g., =

Here, the divisor will have no. of bits = highest power + 1

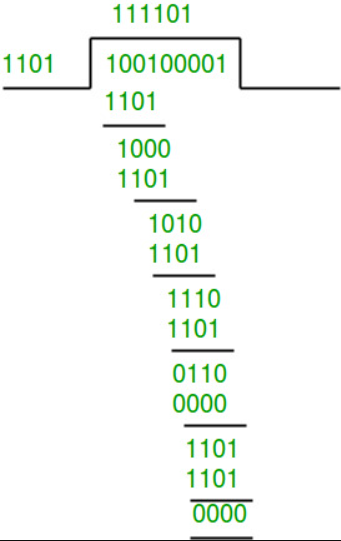
For every power with non-zero coefficient, we set that position bit as 1 and rest as 0.

Therefore, for the above example, set bits will be 7, 5, 2, 1, 0

Divisor= **10100111**

**Receiver side: -**

1. Perform binary division using the same divisor and the transmitted message from sender side.
2. If the remainder is 0, that means there is no error, else error is present.



Since remainder is 0, the transmission is error free.

**Parity bit Check: -**

1. Least expensive method as only 1 bit is added to original message.
2. Count no. of ‘1’ bits in message. If it is even, then add 0, else add 1, to make no. of ‘1’s even.
3. Can detect all single bit and odd no. of errors but cannot detect even no. of errors.

**Hamming distance: -**

1. Given a set of data in form of bits, the hamming distance is the no. of bits in the XOR of any two data.
2. From a given set of data, we can find the min. hamming distance = d. Then for those set of data, we can detect **(d – 1)** bits of error.

Suppose there are four data streams that are sent: 010, 011, 101 and 111.

010 ⊕ 011 = 001, d(010, 011) = 1.

010 ⊕ 101 = 111, d(010, 101) = 3.

010 ⊕ 111 = 101, d(010, 111) = 2.

011 ⊕ 101 = 110, d(011, 101) = 2.

011 ⊕ 111 = 100, d(011, 111) = 1.

101 ⊕ 111 = 010, d(011, 111) = 1.

Hence, the Minimum Hamming Distance, d = 1.

**Bit stuffing: -**

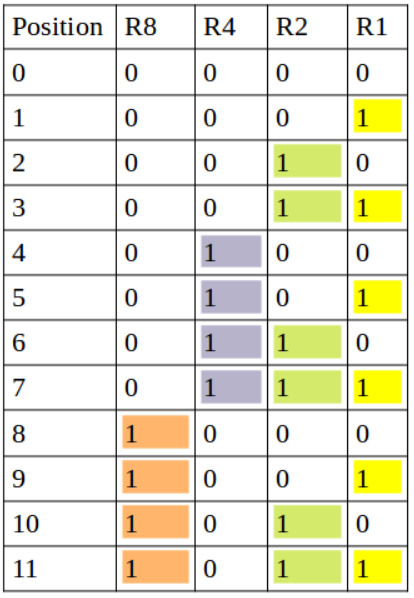
1. In HDLC, the beginning and ending sequence is **01111110**.
2. Suppose we have to send a message – **01111110**0000**01111110**010101111**01111110**. Here the problem is, there is a part of data which resembles beginning/ending sequence, it will cause problems as data might be read as ending sequence.
3. Therefore, when sender receives data from upper layer, it stuffs a 0 when it encounters 5 consecutive 1’s.

Message – **01111110**0000011111**0**10010101111**01111110.**

1. The sender can then un stuff the bit and the beginning and ending sequences are not allowed to appear as part of the data.

**Hamming code: -**

1. Used for error detection and correction.
2. Redundant bits are added to the message. The number of redundant bits can be calculated using the following formula: where, r = redundant bit, m = data bit. Suppose the number of data bits is 7, then the number of redundant bits can be calculated using: = 2^4 ≥ 7 + 4 + 1 Thus, the number of redundant bits= 4 Parity bits.
3. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
4. All the other bit positions are marked as data bits.
5. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form. a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant position (1, 3, 5, 7, 9, 11, etc). b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from the least significant bit (2, 3, 6, 7, 10, 11, etc). c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from the least significant bit (4–7, 12–15, 20–23, etc). d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit bits (8–15, 24–31, 40–47, etc). e. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.
6. Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is odd.
7. Set a parity bit to 0 if the total number of ones in the positions it checks is even.



**R1 -> 1,3,5,7,9,11**

**R2 -> 2,3,6,7,10, 11**

**R3 -> 4,5,6,7**

**R4 -> 8,9,10, 11**

**Example: -**

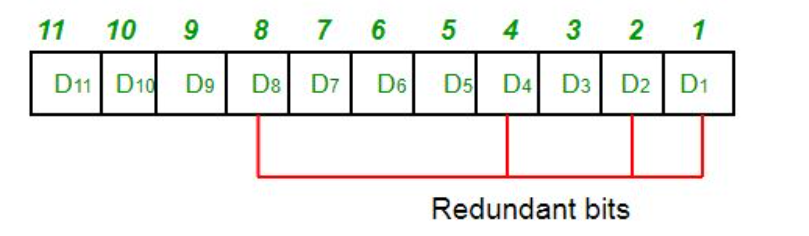
Let’s take an example where data bits(m) = 7.

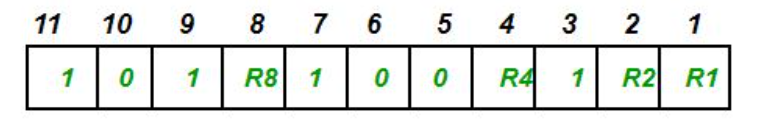
Therefore, the number of redundant bits can be calculated using: = 2^4 ≥ 7 + 4 + 1

The number of data bits = **7**

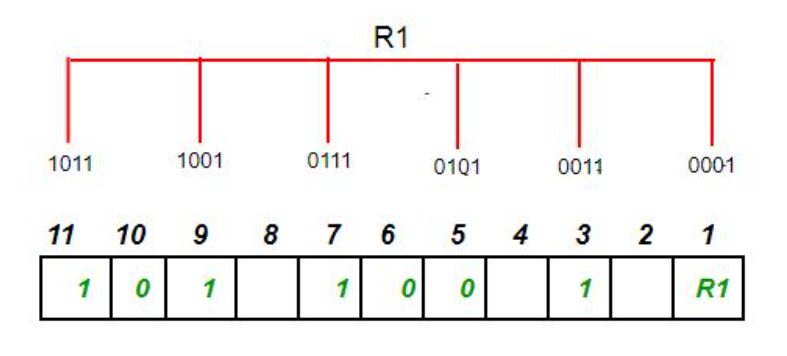
The number of redundant bits = **4**

The total number of bits = **11**

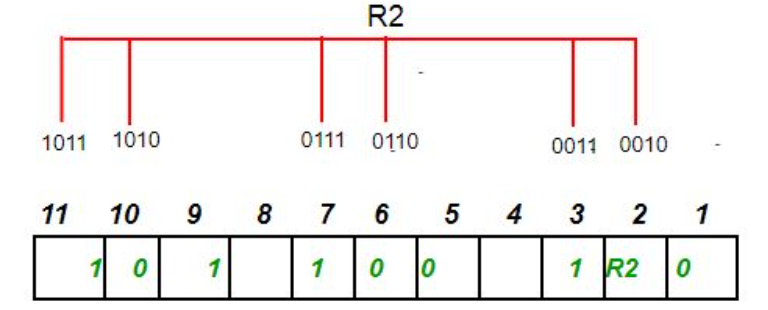
****The redundant bits are placed at positions corresponding to power of 2 – **1, 2, 4, and 8**

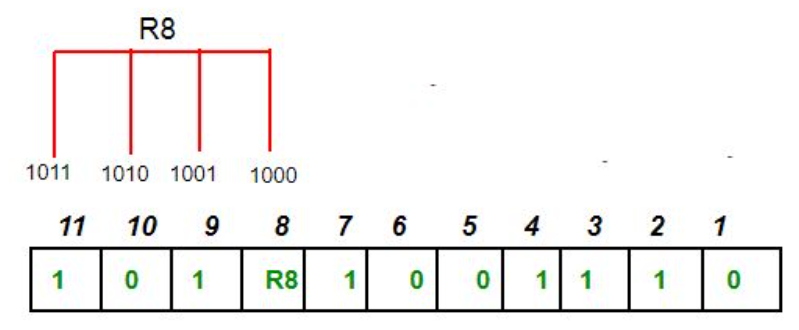
Suppose the data to be transmitted is 1011001, the bits will be placed as follows:

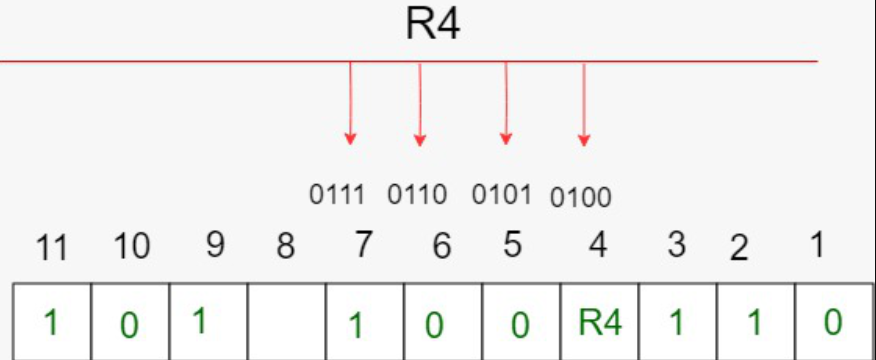
**Determining the Parity bits:**

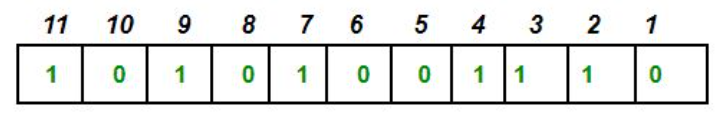
R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position. **R1: bits 1, 3, 5, 7, 9, 11**.

To find the redundant bit R1, we check for even parity. Since the total number of 1’s in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit’s value) = 0

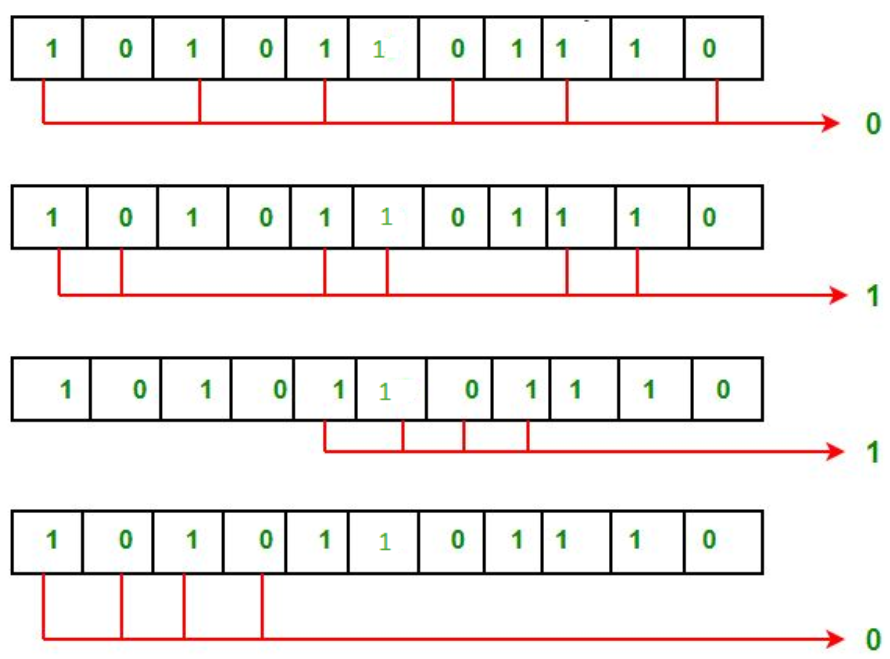
Similarly, we calculate the value of R2, R4, R8 using their respective bits: -

Since the total number of 1’s in all the bit positions corresponding to R2 is odd the value of R2(parity bit’s value) = 1

Since the total number of 1’s in all the bit positions corresponding to R4 is odd the value of R4(parity bit’s value) = 1

Since the total number of 1’s in all the bit positions corresponding to R8 is an even number the value of R8(parity bit’s value) = 0. Thus, the data transferred is:

**Detection and correction: -**

Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number:

The bits give the binary number **0110** whose decimal representation is 6. Thus, bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.

**Features: -**

1. **Error Detection and Correction**: Hamming code is designed to detect and correct single-bit errors that may occur during the transmission of data. This ensures that the recipient receives the same data that was transmitted by the sender.
2. **Redundancy**: Hamming code uses redundant bits to add additional information to the data being transmitted. This redundancy allows the recipient to detect and correct errors that may have occurred during transmission.
3. **Efficiency**: Hamming code is a relatively simple and efficient error-correction technique that does not require a lot of computational resources. This makes it ideal for use in low-power and low-bandwidth communication networks.
4. **Widely Used**: Hamming code is a widely used error-correction technique and is used in a variety of applications, including telecommunications, computer networks, and data storage systems.
5. **Single Error Correction**: Hamming code is capable of correcting a single-bit error, which makes it ideal for use in applications where errors are likely to occur due to external factors such as electromagnetic interference.
6. **Limited Multiple Error Correction**: Hamming code can only correct a limited number of multiple errors. In applications where multiple errors are likely to occur, more advanced error-correction techniques may be required.

**Access Control: -**

When a sender and receiver have a dedicated link to transmit data packets, the data link control is enough to handle the channel. Suppose there is no dedicated path to communicate or transfer the data between two devices. In that case, multiple stations access the channel and simultaneously transmits the data over the channel. It may create collision and cross talk. Hence, the multiple access protocol is required to reduce the collision and avoid crosstalk between the channels. The protocols used for Access Control are known as **Multiple Access Protocols**.

Following are the types of multiple access protocol that is subdivided into the different process as:

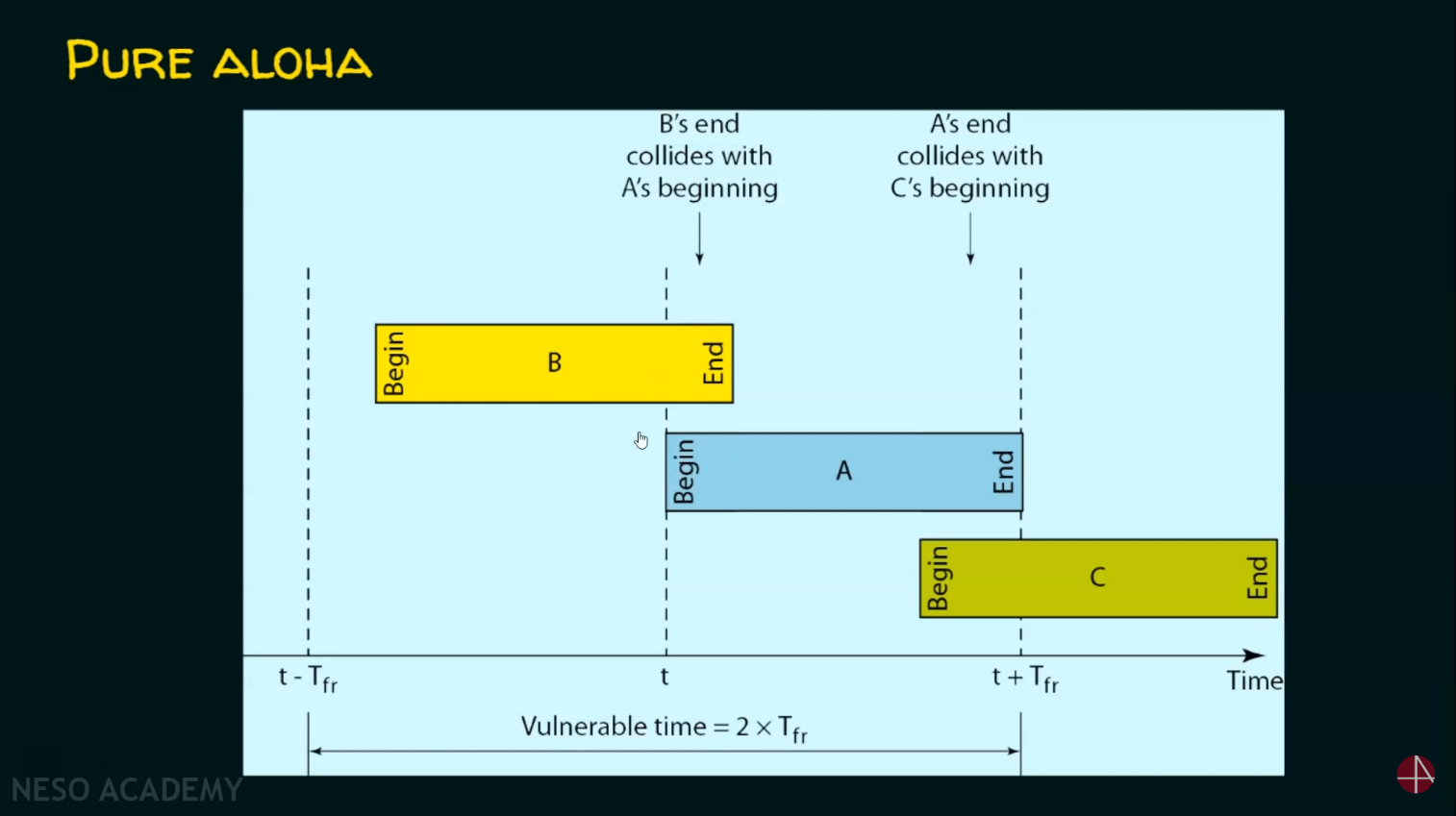
**Random Access Protocols: -**

* In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state (idle or busy).
* In a Random Access method, each station has the right to the medium without being controlled by any other station.
* If more than one station tries to send, there is an access conflict (COLLISION) and the frames will be either destroyed or modified.

**Pure ALOHA: -**

* Pure ALOHA allows stations to transmit whenever they have data to be sent.
* When a station sends data it waits for an acknowledgement.
* If the acknowledgement doesn't come within the allotted time, then the station waits for a random amount of time called **back-off time (Tb)** and re-sends the data.
* Since different stations wait for different amount of time, the probability of further collision decreases.
* The throughput of pure aloha is maximized when frames are of uniform length.

* Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be garbled.
* If the first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted later.



**Vulnerable Time** = 2\*Tfr, where Tfr is the frame transmission time.

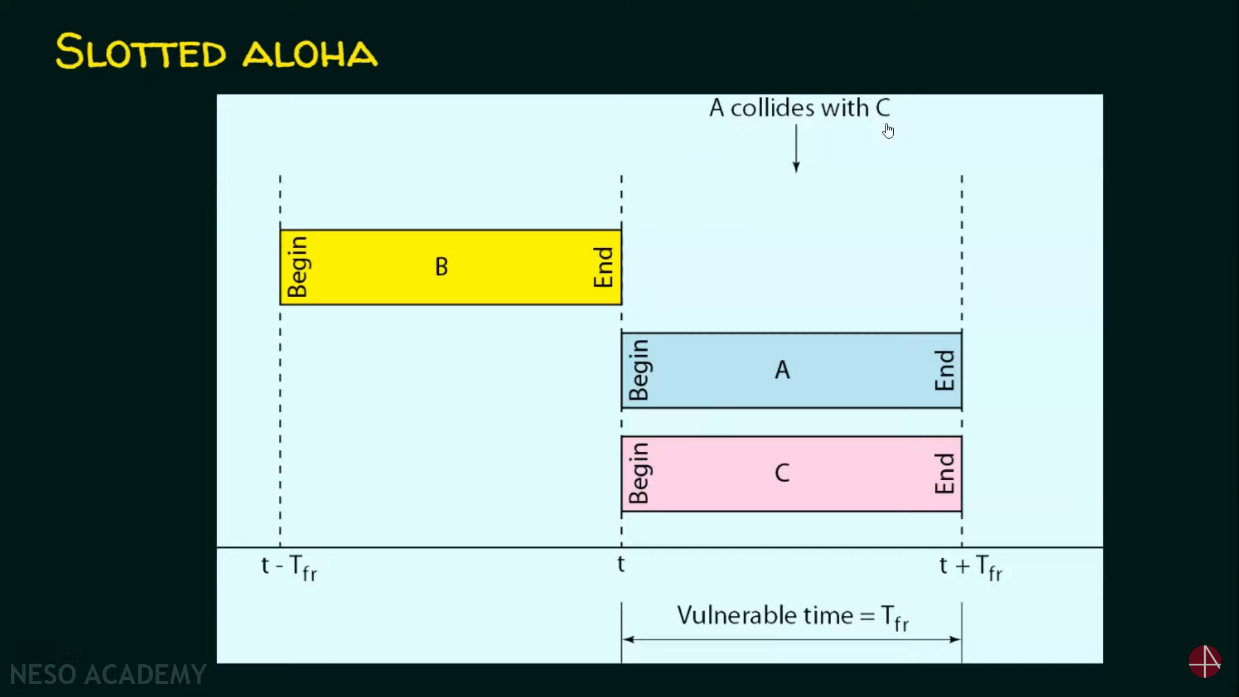
**Throughput** = , where G is the number of stations wish to transmit in the same time.

**Maximum throughput** = 0.184 for G = 0.5 (1/2).

Thus, **Maximum efficiency =** 18.4%

**Slotted ALOHA: -**

* It was developed just to improve the efficiency of pure aloha as the chances for collision in pure aloha are high.
* The time of the shared channel is divided into discrete time intervals called slots.
* Sending of data is allowed only at the beginning of these slots.
* If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.

**Vulnerable Time** = Tfr, where Tfr is the frame transmission time.

**Throughput** = , where G is the number of stations wish to transmit in the same time.

**Maximum throughput** = 0.368 for G = 1

Thus, **Maximum efficiency =** 36.8%

**CSMA Protocol: -**

* Carrier Sense Multiple Access Protocol.
* To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.
* Principle of CSMA: "sense before transmit" or "listen before talk."
* Carrier busy = Transmission is taking place.
* Carrier idle = No transmission currently taking place.
* The possibility of collision still exists because of propagation delay. A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

**Types of CSMA: -**

1. 1-Persistent CSMA
2. P-Persistent CSMA
3. Non-Persistent CSMA
4. O-Persistent CSMA

**Modified versions of CSMA: -**

CSMA/CD (CSMA with Collision Detection)

CSMA/CA (CSMA with Collision Avoidance)

**1-Persistent CSMA: -**

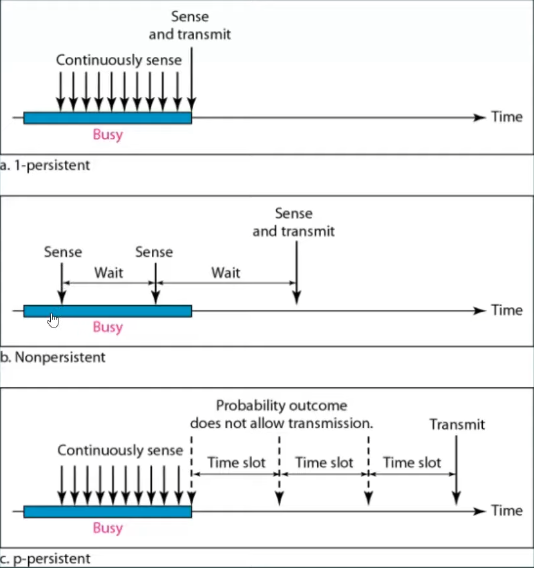
* Before sending the data, the station first listens to the channel to see if anyone else is transmitting the data at that moment.
* If the channel is idle, the station transmits a frame.
* If busy, then it **senses the transmission medium continuously** until it becomes idle.
* Since the station transmits the frame with the probability of 1 when the carrier of channel is idle, this scheme of CSMA is called as 1-Persistent CSMA.
* The propagation delay has an important effect on the performance the protocol. The longer the propagation delay, the more chances there will be of collision, since the bit will take longer time to travel to its destination and another station might sense the channel as free in that time period.

**Non-Persistent CSMA: -**

* Before sending, a station senses the channel. If no one else is sending, the station begins doing so itself.
* However, if the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission.
* Instead, it waits a random period of time and then repeats the algorithm. Consequently, this algorithm leads to better channel utilization but longer delays than 1-persistent CSMA.

**P-Persistent CSMA: -**

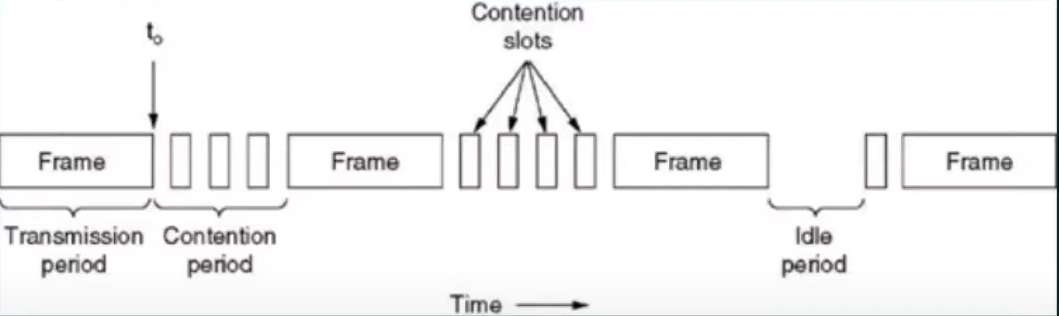
* It applies to slotted channels.
* When a station becomes ready to send, it senses the channel.
* If it is idle, it transmits with a probability P.
* With a probability Q=1-P, it defers until the next slot.
* If that slot is also idle, it either transmits or defers again, with probabilities P and Q.
* This process is repeated until either the frame has been transmitted or another station has begun transmitting.
* In the latter case, the unlucky station acts as if there had been a collision (i.e., it waits a random time and starts again).
* If the station initially senses the channel busy, it waits until the next slot and applies the above algorithm.



**O-Persistent CSMA: -**

Each node is assigned a transmission order by a supervisory node.

**CSMA/CD (Collision Detection): -**

* If two stations sense the channel to be idle and begin transmitting simultaneously, they will both detect the collision almost immediately.
* Rather than finish transmitting their frames, which are irretrievably garbled anyway, they should abruptly stop transmitting as soon as the collision is detected.
* Quickly terminating damaged frames saves time and bandwidth.
* This protocol, known as CSMA/CD (CSMA with Collision Detection) is widely used on LANs in the MAC sublayer.
* Access method used by Ethernet: CSMA/CD.
* At the point marked t0, a station has finished transmitting its frame.
* Any other station having a frame to send may now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision.
* Collisions can be detected by looking at the power or pulse width of the received signal and comparing it to the transmitted signal.
* After a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again, assuming that no other station has started transmitting in the meantime.
* Therefore, model for CSMA/CD will consist of alternate contention and transmission periods, with idle periods occurring when all stations are quiet.
* **Efficiency =** where a = Tp/Tt
* If distance increases, efficiency of CSMA decreases.
* CSMA is not suitable for long distance networks like WAN but works optimally for LAN.
* If length of packet is bigger, the efficiency of CSMA also, maximum limit for length is 1500 Bytes.
* Transmission Time >= Round Trip Time of 1 bit
* Transmission Time >= 2\*Propagation Time

**CSMA/CA: -**

* When a frame is ready, the transmitting station checks whether the channel is idle or busy.
* If the channel is busy, the station waits until the channel becomes idle.
* If the channel is idle, the station waits for an **Inter-frame gap (IFG)** amount of time and then sends the frame.
* After sending the frame, it sets a timer.
* The station then waits for acknowledgement from the receiver. If it receives the acknowledgement before expiry of timer, it marks a successful transmission.
* In some implementations, a Request-To-Send (RTS) and Clear-To-Send (CTS) handshake is used to reserve the channel before transmission. This reduces the chance of collisions and increases efficiency.
* Otherwise, it waits for a back-off time period and restarts the algorithm.

**Network layer: -**

1. **Source to destination / Machine to machine** delivery. It is done using logical address (IP address).
2. **Routing** – Send data from source to destination using optimal(shortest) path. Protocols used are: -

* **RIP (Routing Information Protocol)**
* **OSPF (Open Shortest Path First)**
* **BGP (Border Gateway Protocol)**
* **Link-State Routing**

1. **Fragmentation** – Fragmentation is done when the maximum size of datagram is greater than maximum size of data that can be held in a frame i.e., its Maximum Transmission Unit (MTU). The network layer divides the datagram received from the transport layer into fragments so that data flow is not disrupted.
2. **Congestion control** – Protocols used are: -

* **Leaky Bucket method**
* **Token Bucket method**
* **ICMP (Internet Control Message Protocol)**

IP address is divided into two parts: -

* Host ID
* Network ID

There are two types of IP addressing: -

* Classful IP addressing (For IPv4)
* Classless IP addressing (For IPv4 and IPv6)

**Dotted representation: -**

All IPv4 addresses are of size 32 bit. They are divided into 4 octets, each separated by a dot. The representation looks as: -

**10.10.10.100** or **11001010.01100101.10011010.1001101**

**Classful Addressing: -**

* Given by IANA (Internet Assigned Numbers Authority)
* There are 5 classes in Classful Addressing: -

1. Class A
2. Class B
3. Class C
4. Class D
5. Class E

* Total IP addresses in Classful addressing = **232**.

**Class A: -**

* 1st bit of 1st octet = 0. Thus, no. of IP addresses in Class A = 231.
* The entire 1st octet, i.e., first 8 bits represent Network ID. No. of networks possible = 27-2. We do -2 because 1st and last bit are reserved.
* Remaining 24 bits represent Host ID. No. of networks possible = 224-2. We do -2 because 1st IP is address of network and last IP is reserved for broadcast.
* Default mask for Class A = **255.0.0.0**. (Set 1 for all bits of network ID, i.e., first 8 bits).

**Class B: -**

* First 2 bits bit of 1st octet = 10. Thus, no. of IP addresses in Class A = 230.
* The first 2 octets, i.e., first 16 bits represent Network ID. No. of networks possible = 214-2. We do -2 because 1st and last bit are reserved.
* Remaining 16 bits represent Host ID. No. of networks possible = 216-2. We do -2 because 1st IP is address of network and last IP is reserved for broadcast.
* Default mask for Class B = **255.255.0.0**. (Set 1 for all bits of network ID, i.e., first 16 bits).

**Class C: -**

* First 3 bits bit of 1st octet = 110. Thus, no. of IP addresses in Class A = 229.
* The first 3 octets, i.e., first 24 bits represent Network ID. No. of networks possible = 221-2. We do -2 because 1st and last bit are reserved.
* Remaining 8 bits represent Host ID. No. of networks possible = 28-2. We do -2 because 1st IP is address of network and last IP is reserved for broadcast.
* Default mask for Class C = **255.255.255.0**. (Set 1 for all bits of network ID, i.e., first 24 bits).

**Class D: -**

* First 4 bits bit of 1st octet = 1110. Thus, no. of IP addresses in Class A = 228.
* There are no host and network ID in Class D.
* Class D is reserved for Multicasting, Group E-mail and broadcasting.

**Class E: -**

* First 4 bits bit of 1st octet = 1111. Thus, no. of IP addresses in Class A = 228.
* There are no host and network ID in Class E.
* Class E is reserved for military purpose.

**Problems of Classful IP address: -**

1. Wastage of IP address.
2. Maintenance is complex and consuming.
3. More prone to errors.

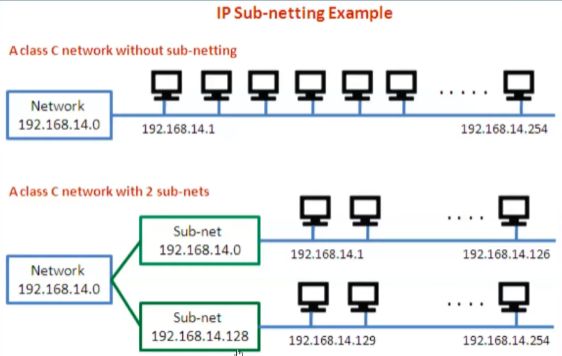
**Classless Addressing: -**

* There are no classes. Only blocks.
* Here IP is divided into Block ID and Host ID.
* Notation in classless addressing = **a.b.c.d/n**. Here, a, b, c and d are 8-bit octets. ‘n’ represents mask or no. of bits representing block. Ex = 200.10.20.40/28. Thus first 28 bits will be Block ID. Last 4 bits will be for Host ID.
* We get the default mask using n. Then we can AND default mask and the IP to get network address.
* We can OR the IP and complement of mask to get broadcast address.

**Rules: -**

1. Addresses should be contiguous.
2. No. of addresses must be in power of 2.
3. First address of every block must be evenly divisible with size of block.

**Subnetting: -**

* A subnetwork or subnet is a logical subdivision of an IP network.
* The practice of dividing a network into two or more networks is called subnetting.
* Computers that belong to a subnet are addressed with an identical most-significant bit-group in their IP addresses.

**Steps in subnetting: -**

1. Identify the class of the IP address and note the Default Subnet Mask.
2. Convert the Default Subnet Mask into Binary.
3. Note the number of hosts required per subnet and find the Subnet Generator (SG) and octet position.
4. Generate the new subnet mask.
5. Use the SG and generate the network ranges (subnets) in the appropriate octet position.

Example: Subnet the IP address 216.21.5.0 into 30 hosts in each subnet.

Solution: -

We will follow the 5 steps in subnetting.

1. Class C – Default Mask: **255.255.255.0**
2. The Default mask in binary = **11111111.11111111.11111111.00000000**
3. No. of hosts per subnet = **30 (11110) – 5 bits**.

Minimum bits required to represent 30 – 5 bits. Thus, we reserve 5 bits from the rightmost position in the subnet mask and make the rest.

1. New subnet mask = **11111111.11111111.11111111.11100000** or **255.255.255.224** or **/27** (classless notation).

The position of the first 1 we encounter from the right is the SG. Thus, **SG = 25 = 32**.

Note that the position will start from the 0 to the right to 7 till the left for each octet.

Suppose if first 1 from right is in 3rd octet, then we start from 0 of the 3rd octet.

This SG is in the 4th octet. Thus, **Octet position = 4**.

**No. of hosts per subnet = 25 = 32 (5 – min. bits to represent no. of hosts)**

**No. of subnets = 23 = 8 (No. of 1’s in the subnet mask for those octets in which are 0 in default mask)**

1. Network ranges (Subnets)

First subnet = 216.21.5.0. Add 32 (SG) to 4th octet (Octet position) to get starting address of each subnet

The last IP of each subnet will be starting IP of next subnet – 1.

216.21.5.0 - 216.21.5.31

216.21.5.32 - 216.21.5.63

216.21.5.64 - 216.21.5.127

216.21.5.128 - 216.21.5.159

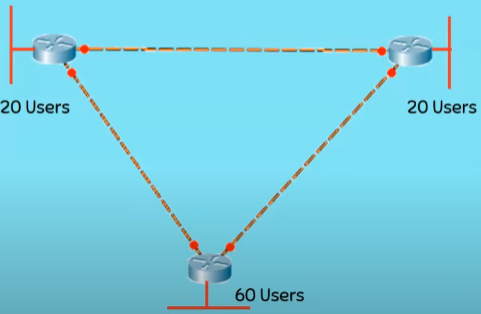
And so on.

The first and last IP of each subnet is the network address of the subnet and broadcast address respectively.

**VLSM (Variable Length Subnet Masking): -**

* Variable Length Subnet Mask (VLSM) is a technique used in IP network design to create subnets with different subnet masks.
* The no. of hosts in each subnet will vary.

Example: -

Subnet 192.168.10.0/24 to address the network by using the most efficient addressing possible.

There are 6 networks possible, 3 with each router and 3 between the 3 routers.

Steps: -

1. Start with the largest subnet (60 users) and normally subnet it as we did in FLSM.
2. Class C – Default Mask: **255.255.255.0**
3. The Default mask in binary = **11111111.11111111.11111111.00000000**
4. No. of hosts per subnet = **60 (11110) – 6 bits**.
5. New subnet mask = **11111111.11111111.11111111.11000000** or **255.255.255.192** or **/26** (classless notation).

This SG is in the 4th octet. Thus, **Octet position = 4**.

**No. of hosts per subnet = 26 = 64 (5 – min. bits to represent no. of hosts)**

1. Now we for the subnets with 20 users.
2. Class C – Default Mask: **255.255.255.0**
3. The Default mask in binary = **11111111.11111111.11111111.00000000**
4. No. of hosts per subnet = **20 (10100) – 5 bits**.
5. New subnet mask = **11111111.11111111.11111111.11100000** or **255.255.255.224** or **/27** (classless notation).

This SG is in the 4th octet. Thus, **Octet position = 4**.

**No. of hosts per subnet = 25 = 32 (5 – min. bits to represent no. of hosts)**

1. Now for the 3 crossover links which require 2 hosts each.
2. Class C – Default Mask: **255.255.255.0**
3. The Default mask in binary = **11111111.11111111.11111111.00000000**
4. No. of hosts per subnet = **2 (10) – 2 bits**.
5. New subnet mask = **11111111.11111111.11111111.11111100** or **255.255.255.252** or **/30** (classless notation).

This SG is in the 4th octet. Thus, **Octet position = 4**.

**No. of hosts per subnet = 22 = 4 (2 – min. bits to represent no. of hosts)**

1. Network ranges

192.168.10.0 - 192.168.10.63 /26 (Handover this to 60 Users Network)

192.168.10.64 - 192.168.10.95 /27 (Handover this to 20 Users Network)

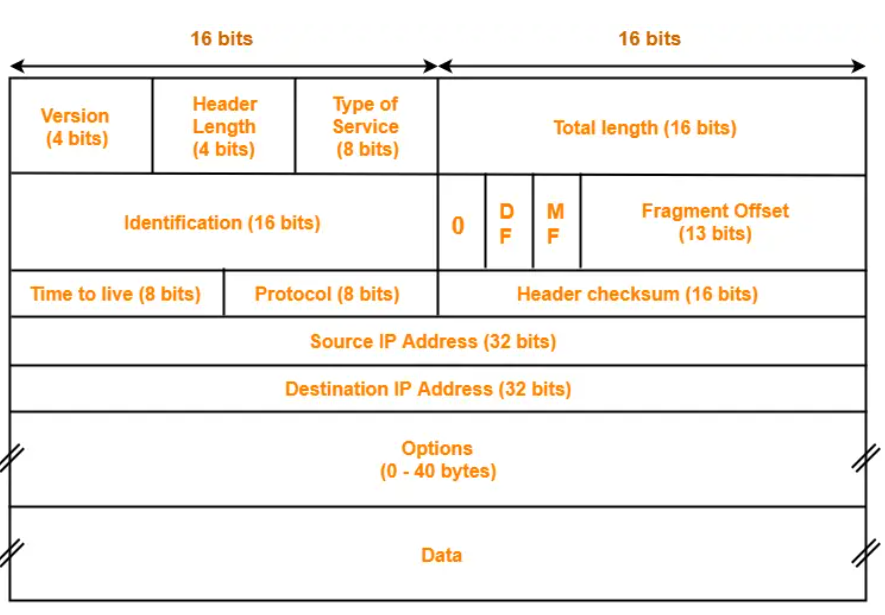
192.168.10.96 - 192.168.10.127 /27 (Handover this to another 20 Users Network)

192.168.10.128-192.168.10.131 /30 (Handover this to Crossover Link)

192.168.10.132-192.168.10.135 /30 (Handover this to Crossover Link)

192.168.10.136 -192.168.10.139 /30 (Handover this to Crossover Link)

**IPv4 header format: -**

IPv4 is a connectionless protocol for use on packet-switched networks. It is a datagram service.

1. **Version**- Version is a 4-bit field that indicates the IP version used. The most popularly used IP versions are version-4 (IPv4) and version-6 (IPv6). Only IPv4 uses the above header. So, this field always contains the decimal value 4.
2. **Header Length** - Header length is a 4-bit field that contains the length of the IP header. It helps in knowing from where the actual data begins. The length of IP header always lies in the range- [20 bytes , 60 bytes]. Because options might vary from 0 to 40.
3. **Type Of Service** - Type of service is a 8 bit field that is used for Quality of Service (QoS). The datagram is marked for giving a certain treatment using this field.
4. **Total Length** - Total length is a 16 bit field that contains the total length of the datagram (in bytes).

Total length = Header length + Payload length

Minimum total length of datagram = 20 bytes (20 bytes header + 0 bytes data)

Maximum total length of datagram = Maximum value of 16 bit word = 65535 bytes

1. **Identification** - Identification is a 16 bit field. It is used for the identification of the fragments of an original IP datagram.
2. **DF Bit** - DF bit stands for Do Not Fragment bit. Its value may be 0 or 1.When DF bit is set to 0, It grants the permission to the intermediate devices to fragment the datagram if required. When DF bit is set to 1, It indicates the intermediate devices not to fragment the IP datagram at any cost.
3. **MF Bit** - MF bit stands for More Fragments bit. Its value may be 0 or 1. When MF bit is set to 0, It indicates to the receiver that the current datagram is either the last fragment in the set or that it is the only fragment. When MF bit is set to 1, It indicates to the receiver that the current datagram is a fragment of some larger datagram. More fragments are following. MF bit is set to 1 on all the fragments except the last one.
4. **Fragment Offset** - Fragment Offset is a 13 bit field. It indicates the position of a fragmented datagram in the original unfragmented IP datagram. The first fragmented datagram has a fragment offset of zero.
5. **Time To Live** -Time to live (TTL) is a 8 bit field. It indicates the maximum number of hops a datagram take to reach the destination. The main purpose of TTL is to prevent the IP datagrams from looping around forever in a routing loop.
6. **Protocol** - Protocol is a 8 bit field. It tells the network layer at the destination host to which protocol the IP datagram belongs to. In other words, it tells the next level protocol to the network layer at the destination side. Protocol number of ICMP is 1, IGMP is 2, TCP is 6 and UDP is 17.
7. **Header Checksum** - Header checksum is a 16 bit field. It contains the checksum value of the entire header. The checksum value is used for error checking of the header. At each hop, the header checksum is compared with the value contained in this field. If header checksum is found to be mismatched, then the datagram is discarded.
8. **Source IP Address** - Source IP Address is a 32 bit field. It contains the logical address of the sender of the datagram.
9. **Destination IP Address** - Destination IP Address is a 32 bit field. It contains the logical address of the receiver of the datagram.
10. **Options** - Options is a field whose size vary from 0 bytes to 40 bytes. This field is used for several purposes such as-

* Record route
* Source routing
* Padding

**IPv6 header format: -**

****

1. **Version** - Indicates version of Internet Protocol which contains bit sequence 0110.
2. **Traffic Class** - The Traffic Class field indicates class or priority of IPv6 packet which is similar to Service Field in IPv4 packet. It helps routers to handle the traffic based on the priority of the packet. If congestion occurs on the router, then packets with the least priority will be discarded.
3. **Flow Label** - Flow Label field is used by a source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers.
4. **Payload Length** - It is a 16-bit (unsigned integer) field, indicates the total size of the payload which tells routers about the amount of information a particular packet contains in its payload.
5. **Next Header** - Next Header indicates the type of extension header (if present) immediately following the IPv6 header. Whereas In some cases it indicates the protocols contained within upper-layer packets, such as TCP, UDP.
6. **Hop Limit** - Hop Limit field is the same as TTL in IPv4 packets. It indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel.
7. **Source Address** - Source Address is the 128-bit IPv6 address of the original source of the packet.
8. **Destination Address** - The destination Address field indicates the IPv6 address of the final destination (in most cases). All the intermediate nodes can use this information in order to correctly route the packet.

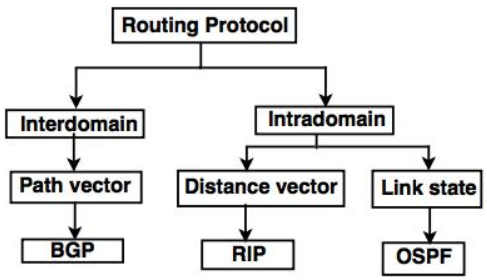
**Routing: -**

Routing in the network layer refers to the process of **selecting the optimal path** for data packets to travel from the source to the destination in a computer network. The network layer, typically implemented using the Internet Protocol (IP), is responsible for logical addressing and routing of data across different networks.

When a packet is sent from a source device, such as a computer or a router, it needs to be forwarded through multiple intermediate network devices, such as routers, switches, and gateways, to reach its intended destination.

There are two primary types of routing:

* **Static Routing**: In static routing, network administrators manually configure the routing table of each router. The routes remain fixed unless manually updated, which makes static routing less adaptable to network changes.
* **Dynamic Routing**: Dynamic routing protocols automate the process of exchanging routing information among routers. These protocols allow routers to dynamically update their routing tables based on changes in the network, such as link failures or new network connections.

**Types of Routing protocols: -**

**Distance Vector Routing (DVR) Protocol: -**

* It requires that a router inform its neighbors of topology changes periodically. Historically known as the old ARPANET routing algorithm (or known as **Bellman-Ford algorithm**).
* Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbors’ distance vectors.
* Information kept by DV router –

1. Each router has an ID associated with each link connected to a router,
2. there is a link cost (static or dynamic).
3. Intermediate hops

* Distance Vector Table Initialization -

Distance to itself = 0

Distance to ALL other routers = infinity number.

**Algorithm: -**

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
   * It receives a distance vector from a neighbor containing different information than before.
   * It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination

Dx(y) = Estimate of least cost from x to y

C(x,v) = Node x knows cost to each neighbor v

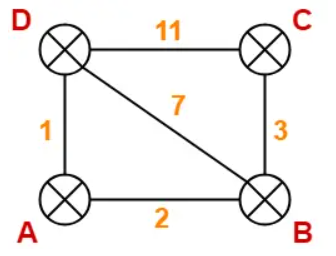
Dx = [Dx(y): y ∈ N ] = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors

– For each neighbor v, x maintains Dv = [Dv(y): y ∈ N ]

When a node x receives new DV estimate from any neighbor v, it saves v’s distance vector and it updates its own DV using B-F equation:

Dx(y) = min { C(x,v) + Dv(y), Dx(y) } for each node y ∈ N

Example: -

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 0 | A |
| B | 2 | B |
| C | ∞ | – |
| D | 1 | D |

**At Router B-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 2 | A |
| B | 0 | B |
| C | 3 | C |
| D | 7 | D |

**At Router C-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | ∞ | – |
| B | 3 | B |
| C | 0 | C |
| D | 11 | D |

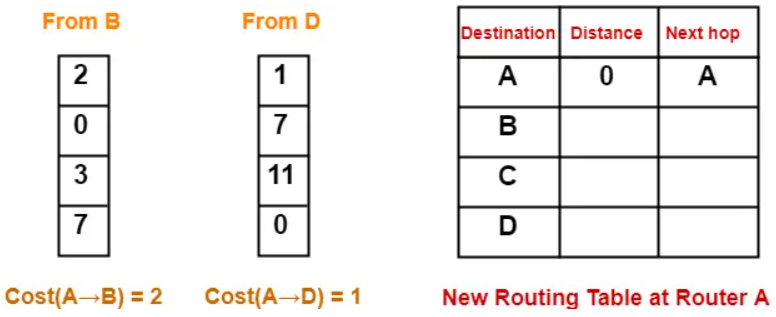
**At Router D-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 1 | A |
| B | 7 | B |
| C | 11 | C |
| D | 0 | D |

At Router A: -

Router A receives distance vectors from its neighbors B and D.

Router A prepares a new routing table as-



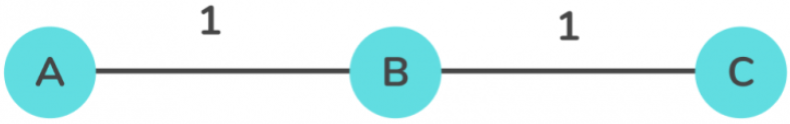
Thus, the new routing table at router A is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 0 | A |
| B | 2 | B |
| C | 5 | B |
| D | 1 | D |

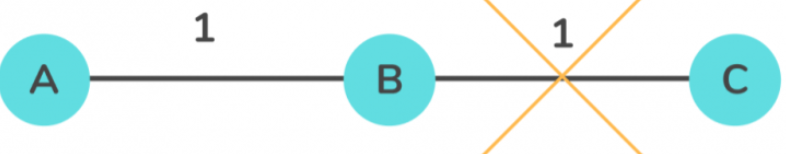
Similarly, routing tables of all others nodes are updated after they receive updated tables from their neighboring nodes.

Disadvantages of Distance Vector routing: –

1. It is slower to converge than link state routing.
2. It is at risk from the count-to-infinity problem.
3. It creates more traffic than link state since a hop count change must be propagated to all routers and processed on each router.
4. For larger networks, distance vector routing results in larger routing tables than link state since each router must know about all other routers.

**Count to infinity problem: -**

So in this example, the Bellman-Ford algorithm will converge for each router, they will have entries for each other. B will know that it can get to C at a cost of 1, and A will know that it can get to C via B at a cost of 2.



If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it from its table. Before it can send any updates it’s possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2. B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3. A will then receive updates from B later and update its cost to 4. They will then go on feeding each other bad information toward infinity which is called as Count to Infinity problem.

**Link State Routing: -**

* Each router shares its neighborhood’s knowledge with every other router in the network. This process is known as **Flooding**.
* Each router in the network understands the network topology then makes a routing table depend on this topology.
* Initial state: Each node knows the cost of its neighbors.
* Final state: Each node knows the entire graph.
* Link state routing uses **Dijkstra’s algorithm** to determine single source shortest path.

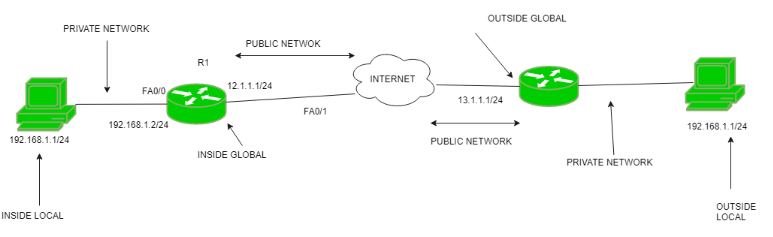
**ARP (Address Resolution Protocol): -**

* It is responsible to find the hardware address (MAC address) of a host from a known IP address.
* The purpose of ARP is to facilitate communication between devices within a local network, such as Ethernet.
* If sending device does not have MAC address of target device in cache then it sends an ARP request packet to the broadcast address of the local network.
* All devices on the local network receive the ARP request packet, but only the device with the matching IP address responds.
* The responding device sends an ARP reply packet containing its MAC address back to the sender.
* The sender then updates its ARP cache with the IP-to-MAC address mapping and uses the MAC address to send the data directly to the target device.

**RARP (Reverse Address Resolution Protocol): -**

* It is a network protocol used to obtain an IP address when the MAC address is known.
* It sends a broadcast RARP request packet onto the network, containing its MAC address. All devices on the network receive the RARP request, but only the RARP server responds.
* The RARP server looks up the MAC address in its configuration database to find the corresponding IP address. Once it retrieves the IP address, it sends a unicast RARP reply packet back to the requesting device, providing the IP address.

**NAT (Network Address Translation): -**

* The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required. This is because a private IP address cannot be routed onto the internet.
* NAT is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts.
* NAT generally operates on a router or firewall.

**Advantages of NAT: –**

1. NAT conserves legally registered IP addresses.
2. It provides privacy as the device’s IP address, sending and receiving the traffic, will be hidden.
3. Eliminates address renumbering when a network evolves.

**Disadvantage of NAT: –**

1. Translation results in switching path delays.
2. Certain applications will not function while NAT is enabled.
3. Complicates tunneling protocols such as IPsec.

**Transport Layer: -**

* End to end/Port to Port delivery.
* Error Control **(Checksum done by TCP)**
* Congestion control
* Flow control **(Stop and wait, GBN, Selective Repeat)**
* Segmentation
* Multiplexing and Demultiplexing – Handle multiple applications sending data at same time.

**Port: -**

* In a computer network, a **Port is a process-specific or an application-specific software defined number that acts as a communication endpoint.**
* Port is an address of a 16-bit unsigned integer number which ranges from 0 to 65535.
* Port is just a unique number assigned to every application of a computer. However, the operating system can automatically assign a port number to the application running on the computer.
* The ports 0 to 1023 are called well-known ports or system ports, these ports are especially associated with particular services.
* The ports from 1024 to 49151 are called registered ports and this range port can be registered with the Internet Assigned Numbers Authority for a specific use.
* The ports from 49152 to 65535 are unassigned ports, called dynamic or ephemeral ports and can be utilized for any type of service.
* The IP address identifies the device e.g., computer. However, an IP address alone is not sufficient for running network applications, as a computer can run multiple applications and/or services. Just as the IP address identifies the computer, The network port identifies the application or service running on the computer.

**General idea of how we can access multiple websites from multiple tabs from a single browser: -**

Port 80 is the default port on which a web server listens for incoming requests. It is not the browser tabs which use port 80. The web servers are designed to listen and handle multiple incoming requests on the port 80 (or any other specified port).

Let’s say your system has an IP address 202.54.15.100 and you open a browser tab having a URL for the web site http:// www. example. com/. Your system will send a request from a random source port (above 1024) of your computer to the web server running on port 80 of www. example. com host.

202.54.15.100: 1025 ——-> www. example. com: 80

The reply for this request will be sent by the web server port 80 to the port 1025 of your system.

www. example. com: 80 ——-> 202.54.15.100: 1025

When you open another tab for the same site, your system will use a different source port but send the request to the same destination port.

202.54.15.100: 1027 ——-> www. example. com: 80

And the reply to this request will be sent to the port 1027 of your system.

www. example. com: 80 ——-> 202.54.15.100: 1027

When you open a third tab for a different website, the network communication will be like:

202.54.15.100: 1028 ——-> www. website2. com: 80

www. website2. com: 80 ——-> 202.54.15.100: 1028

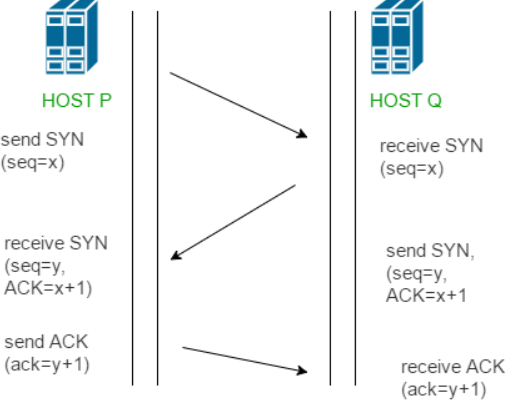
The operating system and the networking stack will identify and associate the incoming replies on the different ports (1025, 1027 and 1028 in the examples) with the different browser tabs and there will not be any confusion.

**TCP (Transmission Control Protocol): -**

Features of TCP: -

1. **Segment Numbering System** - TCP keeps track of the segments being transmitted or received by assigning numbers to each and every single one of them. A specific Byte Number is assigned to data bytes that are to be transferred while segments are assigned sequence numbers. Acknowledgment Numbers are assigned to received segments.
2. **Connection Oriented** - It means sender and receiver are connected to each other till the completion of the process. The order of the data is maintained i.e., order remains same before and after transmission.
3. **Full Duplex** - In TCP data can be transmitted from receiver to the sender or vice – versa at the same time. It increases efficiency of data flow between sender and receiver.
4. **Flow Control** - Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery. The receiver continually hints to the sender on how much data can be received (using a sliding window)
5. **Error Control** - TCP implements an error control mechanism for reliable data transfer. Error control is byte-oriented. Segments are checked for error detection Error Control includes – Corrupted Segment & Lost Segment Management, Out-of-order segments, Duplicate segments, etc.
6. **Congestion Control** - TCP takes into account the level of congestion in the network. Congestion level is determined by the amount of data sent by a sender.

**3-Way Handshake process: -**

* This could also be seen as a way of how TCP connection is established.
* TCP provides reliable communication with something called Positive Acknowledgement with Re-transmission (PAR).
* **Step 1 (SYN)**: In the first step, the client wants to establish a connection with a server, so it sends a segment with **SYN (Synchronize Sequence Number)** which informs the server that the client is likely to start communication and with what sequence number it starts segments with.
* **Step 2 (SYN + ACK)**: Server responds to the client request with SYN-ACK signal bits set. **Acknowledgement (ACK)** signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with.
* **Step 3 (ACK)**: In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer.
* Data transfer can start after Step-3.

**Congestion and Congestion Control: -**

* Network Congestion occurs when the traffic flowing through a network exceeds its maximum capacity.
* In most cases, congestion is a temporary issue with the network caused due to a sudden upsurge of traffic, however, sometimes, a network is continually congested, indicating a deeper problem. End-users perceive network congestion as Network Slowdown or a very large delay in processing requests.
* As delay increases, performance decreases. If delay increases, retransmission occurs, making situation worse.

**SCTP: -**

* SCTP stands for Stream Control Transmission Protocol.
* It is a connection- oriented protocol in computer networks which provides a full-duplex association i.e., transmitting multiple streams of data between two end points at the same time that have established a connection in network.
* It is sometimes referred to as next generation TCP or TCPng, SCTP makes it easier to support telephonic conversation on Internet. A telephonic conversation requires transmitting of voice along with other data at the same time on both ends, SCTP protocol makes it easier to establish reliable connection.

**Characteristics of SCTP: -**

1. **Unicast with Multiple properties** – It is a point-to-point protocol which can use different paths to reach end host.
2. **Reliable Transmission** – It uses SACK and checksums to detect damaged, corrupted, discarded, duplicate and reordered data. It is similar to TCP but SCTP is more efficient when it comes to reordering of data.
3. **Message oriented** – Each message can be framed and we can keep order of data stream and tabs on structure. For this, In TCP, we need a different layer for abstraction.
4. **Multi-homing** – It can establish multiple connection paths between two end points and does not need to rely on IP layer for resilience.
5. **Security** – Another characteristic of SCTP that is security. In SCTP, resource allocation for association establishment only takes place following cookie exchange identification verification for the client (INIT ACK). Man-in-the-middle and denial-of-service attacks are less likely as a result. Furthermore, SCTP doesn’t allow for half-open connections, making it more resistant to network floods and masquerade attacks.

**QoS (Quality of Service): -**

* QoS refers to a broad collection of network technologies and techniques. The goal of QoS is to provide guarantees on the ability of a network to deliver predictable results. Elements of network performance within the scope of QoS often include availability, bandwidth, latency and error rates.
* QoS involves prioritization of network traffic. QoS can be targeted at a network interface toward a given server or router’s performance, or in terms of specific applications. A network monitoring system must be typically be deployed as part of QoS, to ensure that networks are performing at a desired level.
* QoS is especially important for the new generation of Internet applications such a VoIP (Video on demand) and other consumer services. Some core networking technologies like Ethernet were not designed to support prioritized traffic or guaranteed performance making it difficult to implement QoS solutions over the Internet.

**Techniques to improve QoS: -**

* **Scheduling** - Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Several scheduling techniques are designed to improve the quality of service. We discuss three of them here: FIFO queuing, priority queuing, and weighted fair queuing.

1. FIFO Queuing **-** In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them.
2. Priority Queuing - In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest- priority queue are processed last.
3. Weighted Fair Queuing - A better scheduling method is weighted fair queuing. In this technique, the packets are still assigned to different classes and admitted to different queues. The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight. The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight.

* **Traffic Shaping -**Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.
* **Resource Reservation -** A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand.
* **Admission Control -** Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

**Differences between Leaky bucket and Token bucket algorithm: -**

| **Basis** | **Leaky Bucket algorithm** | **Token Bucket algorithm** |
| --- | --- | --- |
| Operation | An imaginary bucket is used to hold incoming packets. Each packet represents a fixed-size unit of data. The bucket has a constant leak rate, which determines the rate at which packets are released from the bucket. If the bucket overflows, the excess packets are discarded. | A bucket holds a certain number of tokens. Each token represents a fixed-size unit of data. Tokens are generated at a fixed rate and added to the bucket. When a packet arrives, it is checked against the number of available tokens in the bucket. If there are enough tokens, the packet is allowed to be transmitted, and the corresponding tokens are removed from the bucket. If there are not enough tokens, the packet is delayed or discarded. |
| Packet Treatment | Packets that arrive when the bucket is full are typically discarded, as there is no provision for allowing bursts. The leak rate determines the maximum sustainable rate at which packets are released from the bucket. | Packets can be transmitted even if the bucket is empty, as long as there are sufficient tokens available. This allows for short bursts of data to be transmitted at a rate higher than the average token generation rate. The algorithm provides more flexibility in handling bursts of traffic. |
| Handling traffic peaks | Effective in smoothing out traffic by enforcing a constant release rate. It regulates the flow of packets to a fixed average rate, preventing network congestion. However, it does not accommodate traffic bursts and may discard packets if the bucket overflows. | Allows for occasional bursts of traffic by allowing packets to be transmitted if sufficient tokens are available, even if the token generation rate is lower than the burst rate. It provides better support for handling traffic peaks without discarding packets. |
| Packet delay | Packets that cannot be transmitted immediately due to a full bucket are delayed until the bucket has sufficient capacity to accommodate them. The delay can vary depending on the rate at which packets are released from the bucket. | Packets that arrive when there are insufficient tokens in the bucket are either delayed until enough tokens are available or discarded immediately, depending on the specific implementation. The delay is typically shorter compared to the Leaky Bucket algorithm. |

**UDP (User Datagram Protocol): -**

* It is a connectionless protocol. Therefore, it is unreliable.
* There is no ordering of data i.e., the sequence in which data is sent is not the same in which it is received.
* It performs Checksum for error detection.
* UDP is stateless protocol. That means it does not keep record of any transmission between any server/client.
* It uses datagrams. A datagram is an independent, self-contained message sent over the network whose arrival, arrival time, and content are not guaranteed.

**Applications of UDP: -**

1. Query response protocol (One request one reply). E.g. – DNS.
2. Speed (Online games, voice over IP)
3. Broadcasting/Multicasting (RIP protocol). Very inefficient to make connections with all neighboring nodes.
4. Continuous streaming (Skype, YouTube)

| **Basis** | **Transmission Control Protocol (TCP)** | **User Datagram Protocol (UDP)** |
| --- | --- | --- |
| Type of Service | TCP is a connection-oriented protocol. Connection  orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data. | [UDP](https://www.geeksforgeeks.org/user-datagram-protocol-udp/)is the Datagram-oriented protocol. This is because  there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission. |
| Reliability | TCP is reliable as it guarantees the delivery of data to the destination router. | The delivery of data to the destination cannot be guaranteed in UDP. |
| Error checking mechanism | TCP provides extensive error-checking mechanisms.  It is because it provides flow control and acknowledgment of data. | UDP has only the basic error-checking mechanism using checksums. |
| Acknowledgment | An acknowledgment segment is present. | No acknowledgment segment. |
| Sequence | Sequencing of data is a feature of Transmission Control  Protocol (TCP). this means that packets arrive in order at the receiver. | There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer. |
| Speed | TCP is comparatively slower than UDP. | UDP is faster, simpler, and more efficient than TCP. |
| Retransmission | Retransmission of lost packets is possible in TCP, but not in UDP. | There is no retransmission of lost packets in the User Datagram Protocol (UDP). |
| Header Length | TCP has a (20-60) bytes variable length header. | UDP has an 8 bytes fixed-length header. |
| Weight | TCP is heavy-weight. | UDP is lightweight. |
| Handshaking Techniques | Uses handshakes such as SYN, ACK, SYN-ACK | It’s a connectionless protocol i.e. No handshake |
| Broadcasting | TCP doesn’t support Broadcasting. | UDP supports Broadcasting. |
| Protocols | TCP is used by [HTTP, HTTPs](https://www.geeksforgeeks.org/difference-between-http-and-https-2/),[FTP](https://www.geeksforgeeks.org/file-transfer-protocol-ftp/), [SMTP](https://www.geeksforgeeks.org/simple-mail-transfer-protocol-smtp/) and [Telnet](https://www.geeksforgeeks.org/introduction-to-telnet/). | UDP is used by [DNS](https://www.geeksforgeeks.org/details-on-dns/), [DHCP](https://www.geeksforgeeks.org/dynamic-host-configuration-protocol-dhcp/), TFTP, [SNMP](https://www.geeksforgeeks.org/simple-network-management-protocol-snmp/), [RIP](https://www.geeksforgeeks.org/routing-information-protocol-rip/), and [VoIP](https://www.geeksforgeeks.org/voice-over-internet-protocol-voip/). |
| Stream Type | The TCP connection is a byte stream. | UDP connection is a message stream. |
| Overhead | Low but higher than UDP. | Very low. |
| Applications | This protocol is primarily utilized in situations when a safe and trustworthy communication procedure is necessary, such as in email, on the web surfing, and in military services. | This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc. |

**Application layer: -**

* The Application Layer is topmost layer in the Open System Interconnection (OSI) model. This layer provides several ways for manipulating the data (information) which actually enables any type of user to access network with ease.
* Application Layer provides a facility by which users can forward several emails and it also provides a storage facility.
* This layer allows users to access, retrieve and manage files in a remote computer.
* It allows users to log on as a remote host.
* This layer provides access to global information about various services.
* This layer provides services which include: e-mail, transferring files, distributing results to the user, directory services, network resources and so on.
* It provides protocols that allow software to send and receive information and present meaningful data to users.
* It handles issues such as network transparency, resource allocation and so on.

**Protocols in Application layer: -**

**DNS (Domain Name System): -**

* DNS stands for Domain Name System.
* It works on port 53. It uses UDP as its transport protocol.
* The DNS service translates the domain name (selected by user) into the corresponding IP address.
* Every host is identified by the IP address but remembering numbers is very difficult for people also the IP addresses are not static therefore a mapping is required to change the domain name to the IP address.
* For example- If you choose the domain name as www.abcd.com, then DNS must translate it as 192.36.20.8 (random IP address written just for understanding purposes).
* There are various kinds of DOMAIN: -

1. Generic domains: .com(commercial), .edu(educational), .mil(military), .org(nonprofit organization), .net(similar to commercial) all these are generic domains.
2. Country domain: .in (India) .us .uk
3. Inverse domain: if we want to know what is the domain name of the website. Ip to domain name mapping.

**DDNS (Dynamic Domain Name System): -**

* When DNS (Domain Name System) was designed, nobody expected that there would be so many address changes such as adding a new host, removing a host, or changing an IP address.
* When there is a change, the change must be made to the DNS master file which needs a lot of manual updating and it must be updated dynamically.
* DDNS is a method of automatically updating a name server in the Domain Name Server (DNS), often in real-time, with the active DDNS configuration of its configured hostnames, addresses or other information.
* By using DDNS, users can create a hostname (e.g. “myhome.dyndns.org”) that will always point to their router’s current IP address, making it easy to connect to devices on their home network from anywhere in the world
* DDNS can use an authentication mechanism to provide security and prevent unauthorized changes in DNS records.

**TELNET (Teletype Network): -**

* TELNET stands for Teletype Network. It is a type of protocol that enables one computer to connect to the local computer.
* It works on port 23 and uses TCP as transport protocol.
* The computer which is being connected to i.e., which accepts the connection known as the remote computer.
* During telnet operation, whatever is being performed on the remote computer will be displayed by the local computer. Telnet operates on a client/server principle. The local computer uses a telnet client program and the remote computers use a telnet server program.
* The logging process can be further categorized into two parts:

1. Local Login: Whenever a user logs into its local system, it is known as local login.
2. Remote Login: Remote Login is a process in which users can log in to a remote site i.e., computer and use services that are available on the remote computer.

**SMTP (Simple Mail Transfer Protocol): -**

* SMTP stands for Simple Mail Transfer Protocol.
* It works in port 25 and uses TCP as transport 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 an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25.
* After successfully establishing a TCP connection the client process sends the mail instantly.
* SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) is used to retrieve those emails at the receiver’s side.
* The SMTP model is of two types:

1. End-to-end method - Used to communicate between different organizations
2. Store-and-forward method - Used within an organization.

**FTP (File Transfer Protocol): -**

* It is used to move files between local and remote file systems.
* FTP is an unusual service in that it utilizes two ports. Traditionally these are port 21 for the command port and port 20 for the data port. It uses TCP as transport protocol.
* Types of Connection in FTP: -
  1. **Control Connection:** For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of a control connection.
  2. **Data connection**: For sending the actual file, FTP makes use of a data connection. FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection.
* FTP allows three types of data structures: -

1. **File Structure**: In file structure, there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. **Record Structure**: In record structure, the file is made up of sequential records.
3. **Page Structure**: In page structure, the file is made up of independent indexed pages.

**WWW (World Wide Web): -**

* The World Wide Web is abbreviated as WWW and is commonly known as the web. The WWW was initiated by CERN (European library for Nuclear Research) in 1989.
* WWW can be defined as the collection of different websites around the world, containing different information shared via local servers (or computers).
* The World Wide Web is based on several different technologies: Web browsers, Hypertext Markup Language (HTML) and Hypertext Transfer Protocol (HTTP).
* There are 3 components of the web:

1. **Uniform Resource Locator (URL)**: serves as a system for resources on the web.
2. **HyperText Transfer Protocol (HTTP)**: specifies communication of browser and server.
3. **Hyper Text Markup Language (HTML)**: defines the structure, organization and content of a webpage.

**HTTP (HyperText Transfer Protocol): -**

* It works on port 80 and uses TCP as transport protocol.
* HyperText is the type of text which is specially coded with the help of some standard coding language called HyperText Markup Language (HTML). HTTP is used to transfer hypertext between two computers.
* HTTP provides a standard between a web browser and a web server to establish communication. It is a set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files are shared on the World Wide Web.
* HTTP request is simply termed as the information or data that is needed by Internet browsers for loading a website. This is simply known as HTTP Request.
* There is some common information that is generally present in all HTTP requests. These are mentioned below.

1. HTTP Version
2. URL
3. HTTP Method
4. HTTP Request Headers
5. HTTP Body

* HTTP Response is simply the answer to what a Server gets when the request is raised. There are various things contained in HTTP Response, some of them are listed below.

1. HTTP Status Code
2. HTTP Headers
3. HTTP Body

* HTTP Status Codes are the 3-Digit codes that tell the message or simply tell us about the HTTP Request whether it has been completed or not. There are simply 5 types of status codes.

1. Informational (100 – 199)
2. Successful (200 – 299)
3. Re-directional (300 – 399)
4. Client-Error (400 – 499)
5. Server-Error (500 – 599)

**SNMP (Simple Network Management Protocol): -**

* It uses two ports and both use UDP as their transport protocol.

1. 161 – For SNMP agents
2. 162 – For SNMP managers receiving traps

* It is used to monitor the network, detect network faults, and sometimes even used to configure remote devices.
* There are 3 components of SNMP:

1. **SNMP Manager** – It is a centralized system used to monitor network. It is also known as Network Management Station (NMS)
2. **SNMP agent** – It is a software management module installed on a managed device. Managed devices can be network devices like PC, routers, switches, servers, etc.
3. **Management Information Base** – MIB consists of information on resources that are to be managed. This information is organized hierarchically. It consists of objects instances which are essentially variables.

* There are 3 versions of SNMP: -

1. **SNMPv1** – It uses community strings for authentication and uses UDP only.
2. **SNMPv2c** – It uses community strings for authentication. It uses UDP but can be configured to use TCP.
3. **SNMPv3** – It uses Hash-based MAC with MD5 or SHA for authentication and DES-56 for privacy. This version uses TCP. Therefore, the conclusion is the higher the version of SNMP, the more secure it will be.

**Bluetooth: -**

* Bluetooth is universal for short-range wireless voice and data communication. It is a Wireless Personal Area Network (WPAN) technology and is used for exchanging data over smaller distances.
* It operates in the unlicensed, industrial, scientific, and medical (ISM) band from 2.4 GHz to 2.485 GHz.
* Maximum devices that can be connected at the same time are 7.
* Bluetooth ranges up to 10 meters. It provides data rates up to 1 Mbps or 3 Mbps depending upon the version.
* Bluetooth simply follows the principle of transmitting and receiving data using radio waves. It can be paired with the other device which has also Bluetooth but it should be within the estimated communication range to connect.
* The architecture of Bluetooth defines two types of networks:

1. **Piconet** - Piconet is a type of Bluetooth network that contains one primary node called the master node and seven active secondary nodes called slave nodes. The communication between the primary and secondary nodes can be one-to-one or one-to-many. It also has 255 parked nodes. These are secondary nodes and cannot take participation in communication unless it gets converted to the active state.
2. **Scatternet** - It is formed by using various piconets. A slave that is present in one piconet can act as master or we can say primary in another piconet. This kind of node can receive a message from a master in one piconet and deliver the message to its slave in the other piconet where it is acting as a master. This type of node is referred to as a bridge node. A station cannot be mastered in two piconets.

**Firewall: -**

* It is a device (software or hardware or both) which monitors and controls incoming and outgoing traffic based on predefined rules.
* Firewalls are generally of two types: Host-based and Network-based.

1. **Host- based Firewalls**: Host-based firewall is installed on each network node which controls each incoming and outgoing packet. It is a software application or suite of applications, comes as a part of the operating system. Host-based firewalls are needed because network firewalls cannot provide protection inside a trusted network. Host firewall protects each host from attacks and unauthorized access.
2. **Network-based Firewalls**: Network firewall function on network level. In other words, these firewalls filter all incoming and outgoing traffic across the network. It protects the internal network by filtering the traffic using rules defined on the firewall. A Network firewall might have two or more network interface cards (NICs). A network-based firewall is usually a dedicated system with proprietary software installed.

* There are different types of firewall techniques: -

1. **Packet filter** – Looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.
2. **Application gateway** – Applies security mechanisms to specific applications such as FTP and TELNET servers. This is very effective but can impose a performance degradation.
3. **Circuit-level gateway** – Applies security mechanisms when a TCP or UDP connection has been established. Once the connection has been made, packets can flow between the hosts without further checking.
4. **Proxy server** – A proxy server is a system or router that provides a gateway between users and the internet. It intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses.
5. **Next Generation Firewalls (NGFW)** – Next Generation Firewalls are being deployed these days to stop modern security breaches like advance malware attacks and application-layer attacks. NGFW consists of Deep Packet Inspection, Application Inspection, SSL/SSH inspection

**Cryptography: -**

* Cryptography is technique of securing information and communications through use of codes so that only those people for whom the information is intended can understand it and process it.
* In Network security, CIA must be maintained – **Confidentiality, Integrity** and **Availability**.
* The prefix “crypt” means “hidden” and suffix “graphy” means “writing”.
* In general, there are two types of cryptography: -

1. **Symmetric Key Cryptography**: It is an encryption system where the sender and receiver of message use a single common key to encrypt and decrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange key in a secure manner. The most popular symmetric key cryptography system are Data Encryption Standard (DES) and Advanced Encryption Standard (AES).
2. **Asymmetric Key Cryptography**: Under this system a pair of keys is used to encrypt and decrypt information. A receiver’s public key is used for encryption and a receiver’s private key is used for decryption. Public key and Private Key are different. Even if the public key is known by everyone the intended receiver can only decode it because he alone knows his private key. The most popular asymmetric key cryptography algorithm is RSA algorithm.

**RSA algorithm: -**

* It is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e., Public Key and Private Key. As the name describes that the Public Key is given to everyone and the Private key is kept private.
* Steps in RSA algorithm: -

1. Choose two different large random prime numbers – **p** and **q**. (For problem solving select small prime numbers).
2. Calculate **n = p \* q**.
3. Calculate **∅(n) = (p – 1) \* (q – 1)**.
4. Choose **‘e’** such that **1 < e < ∅(n)**. Here, **‘e’** is coprime to **∅(n)**, i.e., **gcd(e, ∅(n)) = 1**. The value of **‘e’** is the public key.
5. Calculate d such that such that, (**d \* e) mod ∅(n) = 1**. Here **‘d’** is the private key.

A better way to calculate **‘d’** is **(d \* e) = 1 + k.** **∅(n)**.

Thus, **d = (1 + k.** **∅(n)) / e**, where k = 0, 1, 2, …

Only consider positive integer values of d.

Pages in organizer

9 – VC, Datagram

10 – Pulse rate and bitrate; Inverse TDM

11 – QPSK signal

15 – QPSK, QAM, FSK

27 – Virtual packet switching

37 – polling

39 – 2D parity check

40 – Multicast address in MAC

76 – Advantages of subnetting

82 – Ipv4 vs IPv6

**What happens when we enter “google.com” or any valid url in the browser?**

1. Browser checks cache for DNS entry to find the corresponding IP address of website. It looks for following cache. If not found in one, then continues checking to the next until found.

* Browser Cache
* Operating Systems Cache
* Router Cache
* ISP Cache

1. If not found in cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.
2. The requests are sent using small data packets that contain information content of request and IP address it is destined for.
3. Browser initiates a TCP (Transfer Control Protocol) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
4. Browser sends an HTTP request to the web server. GET or POST request.
5. Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, XML and HTML.
6. Server sends out an HTTP response along with the status of response.
7. Browser displays HTML content
8. Finally, Done.