2. Datalink Layer

Data link layer design issues

1. By utilising the physical layer's services, the data link layer transmits and receives information across communication channels.

2. Providing the network layer with a service interface that is precisely defined.

3. Managing errors in transmission.

4. Controlling the data flow to prevent overloading of slow receivers by fast senders.

4. Frame management forms the heart of what the data link layer does.

The data link layer can be designed to offer various services.

a. Unacknowledged connectionless service.

b. Acknowledged connectionless service.

c. Acknowledged connection-oriented service.

For unacknowledged connectionless service, the source machine sends distinct frames to the destination machine, but the destination machine is unable to acknowledge them. There is no logical connection established or released beforehand. If a frame is lost due to noise on the line, no attempt is made in the data link layer to identify or recover from the loss. When the error rate is very low, this type of service is appropriate.

The next level of reliability is acknowledged connectionless service. When this service is provided, no logical connections are used, yet each frame delivered is individually acknowledged. This service is useful when communicating via unstable channels, such as wireless networks. Connection-oriented service is the most sophisticated service that the data link layer can offer the network layer. Before any data is sent, this service establishes a connection between the source and destination machines. The data link layer ensures that every frame submitted is received, and each frame is numbered before being sent over the connection. It also ensures that all frames are received in the correct order and that each frame is received precisely once. The data link layer must use the service offered by the physical layer to give service to the network layer. The physical layer accepts a raw bit stream and attempts to transmit it to its destination. The data link layer does not guarantee that the bit stream it receives is error-free. Some bits values may differ, and the number of bits received may be less than, equal to, or greater than the number of bits transmitted. It is the data link layer's responsibility to detect and, if required, repair problems.

Framing

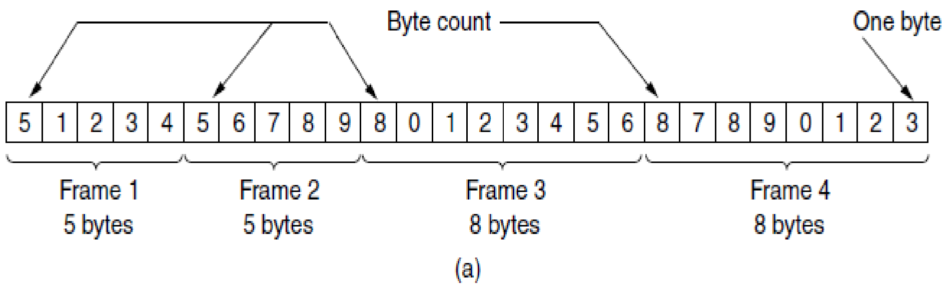
Typically, the datalink layer splits the bit stream into discrete frames, computes a small token called a checksum for each frame, and transmits the frame with the checksum included. Upon reaching the destination that was intended, the checksum is recomputed. The data link layer detects an error has happened and takes action if the freshly computed checksum differs from the one in the frame.

A good design should make it simple for a receiver to determine when a new frame begins and ends.

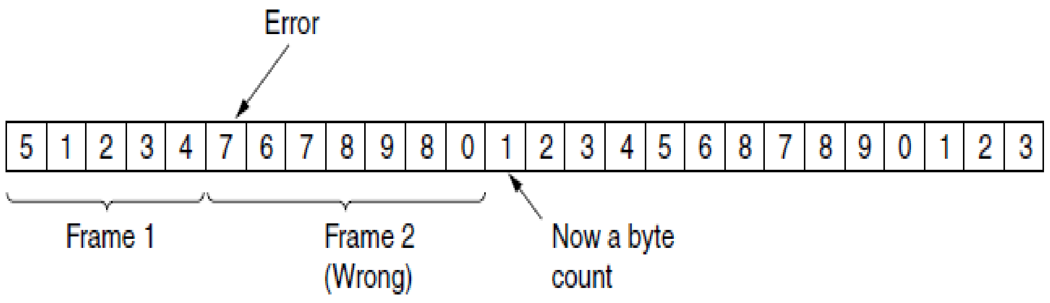
Three approaches are available:

1. Byte count. 2. Flag bytes with byte stuffing. 3. Flag bits with Bit stuffing.

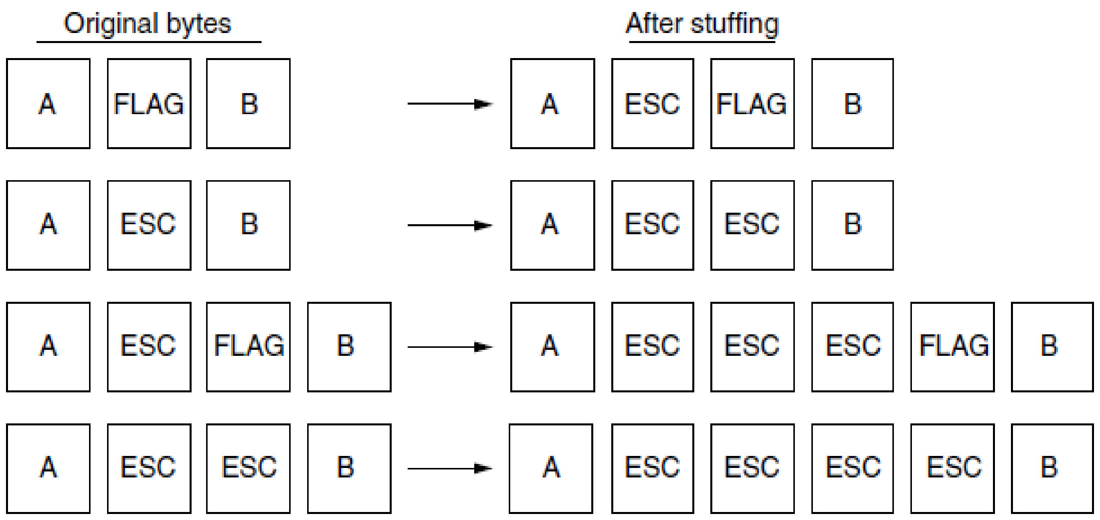
Byte Count: The first framing technique specifies the number of bytes in the frame using a field in the header. The data link layer at the destination may determine the end of the frame by looking at the byte count, which lets it know the end of the frame.



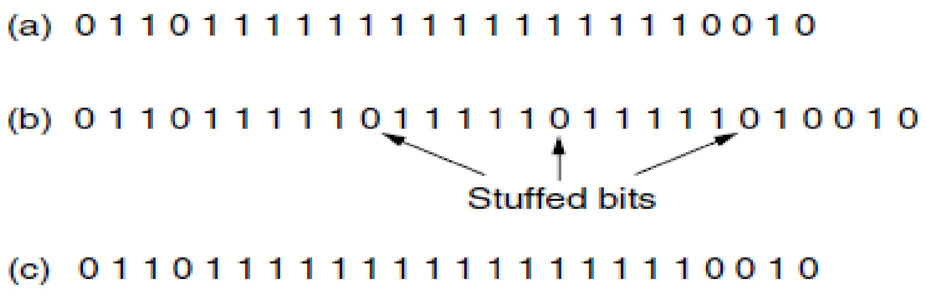
The issue with this approach is that a transmission fault may distort the count. For instance, the destination will lose synchronisation if a single bit flip causes the byte count of 5 in the second frame of Fig. to change to a 7. After then, it won't be able to determine when the next time frame should begin.



Flag bytes with byte stuffing: The second framing technique fixes the resynchronization issue by adding unique bytes at the beginning and end of each frame. Frequently, the beginning and ending delimiters are employed to be the same byte, known as a flag byte. The end of one frame and the beginning of the next are indicated by two consecutive flag bytes. Therefore, the receiver only needs to look for two flag bytes to determine the end of the current frame and the beginning of the next if it ever loses synchronisation. Still, there's a problem that needs to be resolved. The flag byte could occasionally appear in the data; in this case, the framing would be affected. A potential solution to this issue is to have an escape byte (ESC) added by the sender's datalink layer immediately before each "accidental" flag byte in the data. Therefore, the presence or lack of an escape byte preceding a byte can be used to distinguish a framing flag byte from one in the data. Before transferring the data to the network layer, the data link layer on the receiving end eliminates the escape bytes. We refer to this method as "byte stuffing." The next question is, what happens if an escape byte appears in the middle of the data? The answer is that it, too, has an escape byte. The initial escape byte gets removed at the receiver, leaving the data byte that follows it.



Flag bits with bit stuffing: Byte stuffing's drawback is that it requires the use of 8-bit bytes.
Additionally, framing can be done at the bit level, allowing frames to have any number of bits of any size. Every frame starts and finishes with the unique bit sequence 01111110. It is a flag byte pattern.
The sender's data link layer automatically inserts a 0 bit into the outgoing bit stream whenever it finds five consecutive 1s in the data. The receiver automatically destuffs, or deletes, the 0 bit when it detects five consecutive incoming 1 bits followed by a 0 bit. When the user data contains the flag pattern 01111110, it is transmitted as 011111010 but is saved as 01111110 in the memory of the receiver. The flag pattern can unambiguously recognise the boundary between two frames when bit stuffing is used.
As a result, if the receiver becomes confused, all it has to do is check the input for flag sequences, which can only occur at frame boundaries and never within the data.


(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0
Stuffed bits
(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

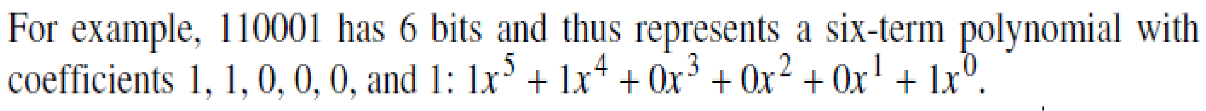
Error Detection & Correction

Transmission errors are to be expected in ageing local circuits, wireless connections. Network designers have devised two fundamental approaches to handling errors. Both add redundant information to the transmitted data. One approach is Incorporating sufficient redundant information to allow the recipient to determine the nature of the transmitted data. The alternative is to have just enough redundancy such that in the case of an error, the receiver can determine what went wrong and request to send it again. The first strategy uses codes to correct errors, while the second strategy uses codes that find errors. We are going to look at two different error-detecting codes.

1. Parity 2. Cyclic redundancy check

Let us examine the initial error-detecting code, which appends one single parity bit to the data. In order to ensure that the number of ones in the code word is even (or unusual), the parity bit is used. A bit is appended to the end of 1011010 when it is transmitted with even parity, resulting in 10110100. Odd parity transforms 1011010 into 10110101. Any error of a single bit results in a code word that has the incorrect parity. This indicates that single-bit errors can be detected.

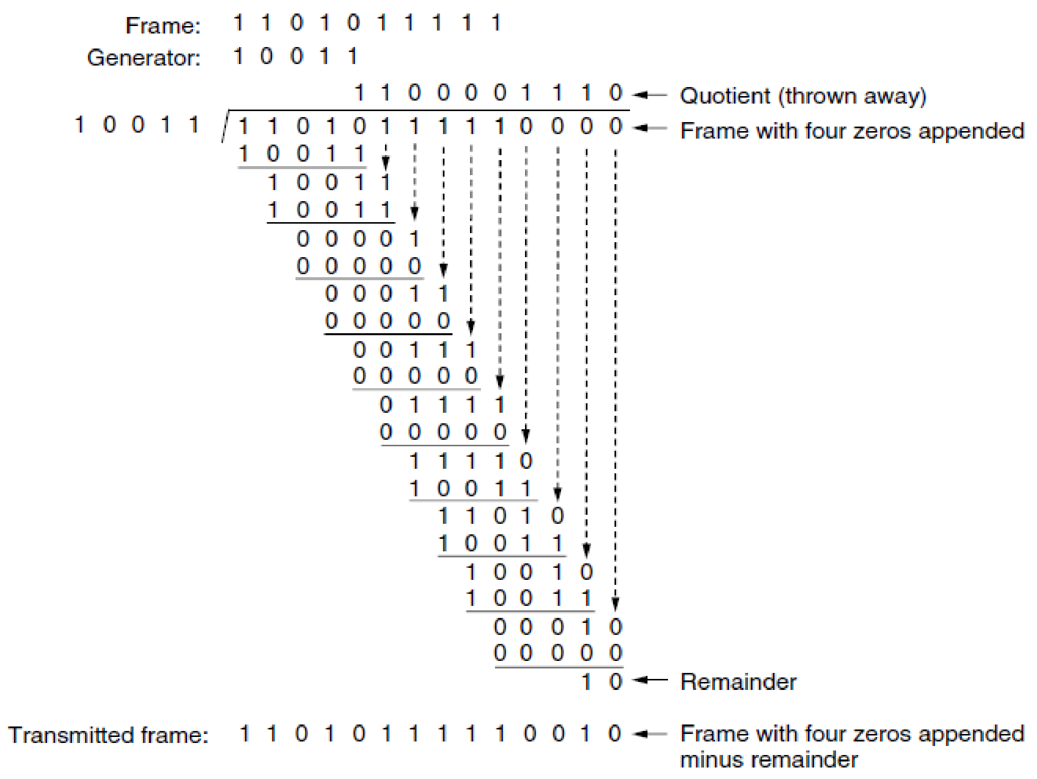
Cyclic Redundancy Check: The datalink layer employs a more robust type of error-detection code known as the CRC (Cyclic Redundancy Check), which is also referred to as a polynomial code. Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 o


For example, 110001 has 6 bits and thus represents a six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1: $1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$.

When using the polynomial code approach, the sender and receiver must first agree on a generator polynomial, G(x).The frame must be longer than the generator polynomial in order to compute the CRC for some frame with m bits corresponding to the polynomial M(x).

The goal is to add a CRC to the end of the frame so that the polynomial represented by the checksummed frame is divisible by G(x). When the receiver receives the checksummed frame, it attempts to divide it by G(x). There was a transmission error if there is a remainder.

Example: Calculate the CRC for a frame 1101011111 using the generator $G(x) = x^4 + x + 1$.


Frame: 1 1 0 1 0 1 1 1 1 1
Generator: 1 0 0 1 1
1 1 0 0 0 0 1 1 1 0
Quotient (thrown away)
1 0 0 1 1
1 1 0 1 0 1 1 1 1 1 0 0 0 0
Frame with four zeros appended
1 0
Remainder
Transmitted frame: 1 1 0 1 0 1 1 1 1 1 0 0 1 0
Frame with four zeros appended minus remainder

Error Correction: Hamming Code:

Working at Bell Telephone Laboratories, mathematician Richard W. Hamming created the Hamming code. Hamming was a pioneer in the development of error-detection and -correction techniques. The Hamming code is a type of error-correcting code that is used to correct transmission errors. The Hamming code, on the other hand, will only correct single-bit errors. It cannot correct multiple-bit or burst errors, nor can it detect errors in the Hamming bits themselves.The Hamming code requires the addition of overhead to the message, which increases the length of the transmission. Hamming bits are randomly introduced into a character at various positions. The Hamming code is the term used to refer to the combination of the data bits and the Hamming bits. A important condition for the placement of the Hamming bits is the mutual agreement between the sender and the receiver regarding their respective positions. In order to determine the appropriate number of redundant Hamming bits required for a specific character length, it is essential to establish a relationship in between the number of bits representing the characters and the number of Hamming bits. A data unit contains m character bits and n Hamming bits. Therefore, the total number of bits in one data unit is m+n. The number of Hamming bits is determined by the following expression

$$2^n \geq m + n + 1$$

Example: For a 12-bit data string of 101100010010, determine the number of Hamming bits required, arbitrarily place the Hamming bits into the data string, determine the logic condition of each Hamming bit, assume an arbitrary single-bit transmission error, and prove that the Hamming code will successfully detect the error.

Ans: Five Hamming bits are sufficient, and a total of 17 bits make up the data stream. Arbitrarily placing five Hamming bits into bit positions 4, 8, 9, 13, and 17 yields

bit position	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	H	1	0	1	H	1	0	0	H	H	0	1	0	H	0	1	0

To determine the logic condition of the Hamming bits, express all bit positions that contain a logic 1 as a five-bit binary number and XOR them together.

Bit position	Binary number	
2	00010	
6	00110	
XOR	00100	
12	01100	
XOR	01000	
14	01110	
XOR	00110	
16	10000	
XOR	10110	= Hamming bits

$b_{17} = 1, \quad b_{13} = 0, \quad b_9 = 1, \quad b_8 = 1, \quad b_4 = 0$

The 17-bit Hamming code is

```
H               H               H   H               H
1   1   0   1   0   1   0   0   1   1   0   1   0   0   0   1   0
```

Assume that during transmission, an error occurs in bit position 14. The received data stream is

$$1 \quad 1 \quad 0 \quad \underbrace{\underline{0}}_{\text{error}} \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0$$

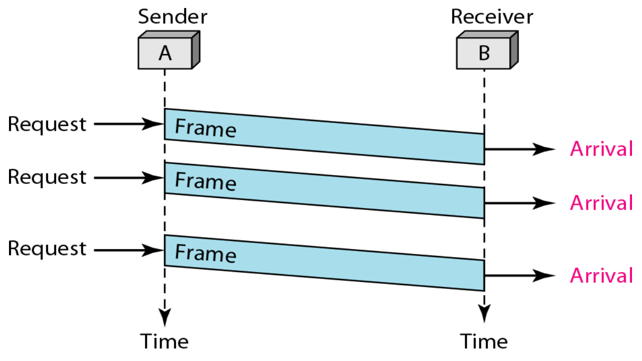
At the receiver, to determine the bit position in error, extract the Hamming bits and XOR them with the binary code for each data bit position that contains a logic 1:

Bit position	Binary number	
Hamming bits	10110	
2	00010	
XOR	10100	
6	00110	
XOR	10010	
12	01100	
XOR	11110	
16	10000	
XOR	01110	= 14

Therefore, bit position 14 contains an error.

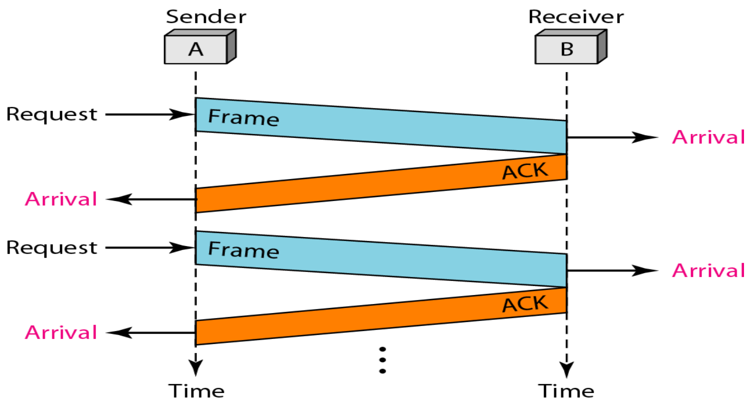
Elementary Data Link Protocols

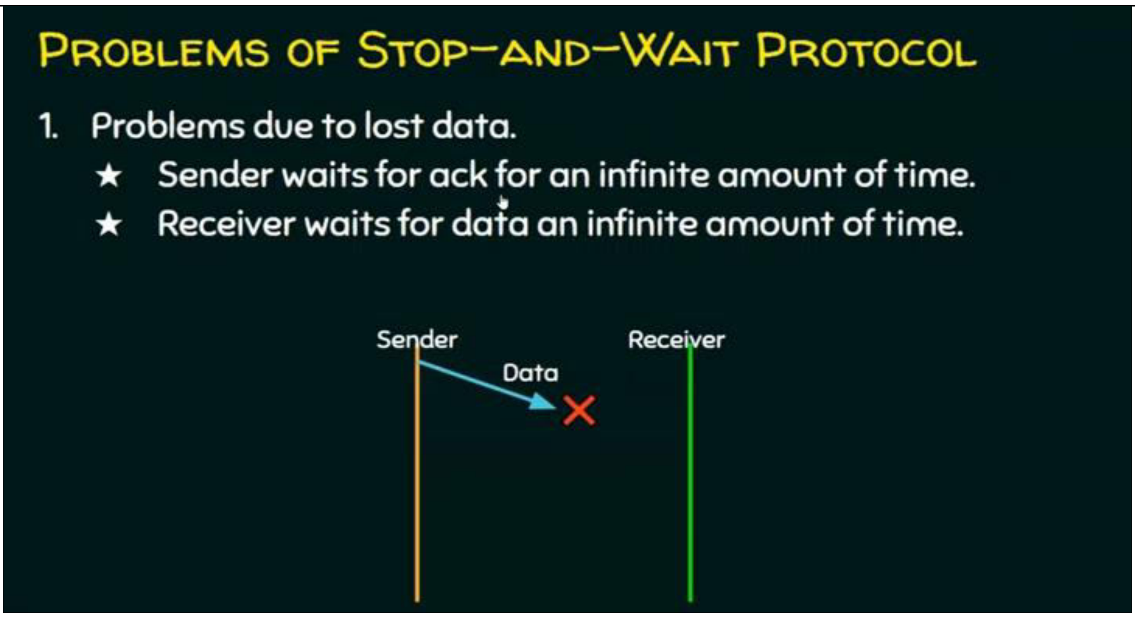
Simplest Protocol: It's quite simple. The transmitter transmits a series of frames without consideration for the recipient. Data may only be transmitted in one direction. Both the transmitter and the receiver are constantly ready to send and receive frames. The processing time is ignored. There is an infinite amount of buffer space. The most important feature is that frames are never lost or corrupted in the communication channel. We're going to name our extremely pointless scheme "Utopia." The utopia protocol cannot function since it does not manage either flow control or error correction.



Stop-and-wait Protocol:

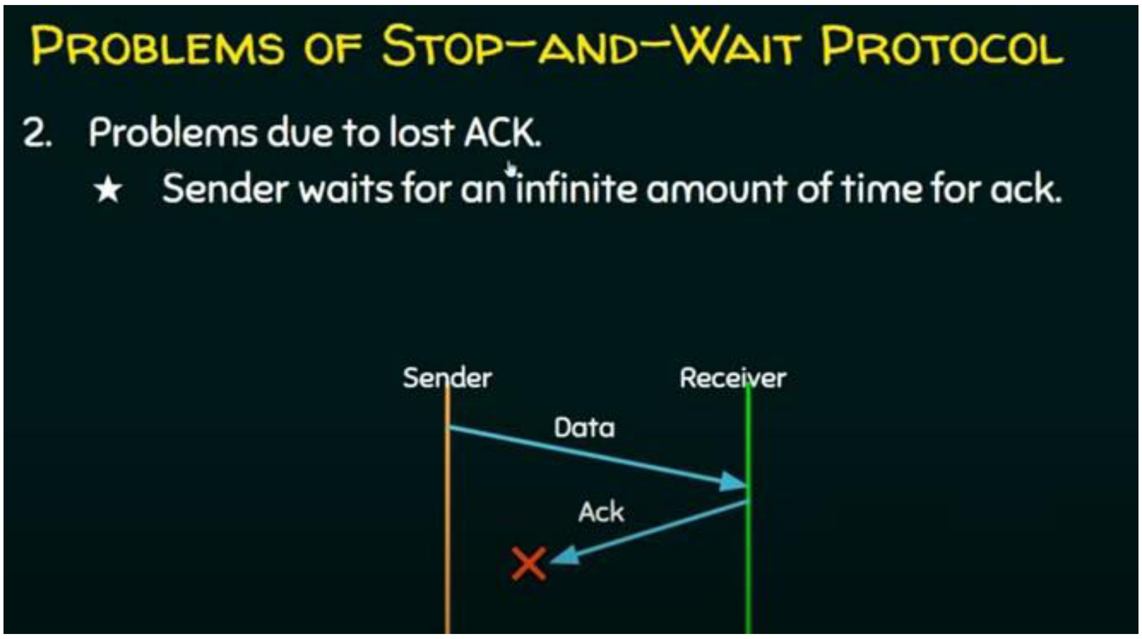
After sending one frame, the sender waits for the recipient's acknowledgment. The sender transmits the subsequent frame after receiving the ACK. The protocol is known as "stop-and-wait" because after sending a frame, the sender waits for the recipient to indicate that it is alright to proceed before sending next frame. This scheme includes the flow control.





PROBLEMS OF STOP-AND-WAIT PROTOCOL

1. Problems due to lost data.
   - ★ Sender waits for ack for an infinite amount of time.
   - ★ Receiver waits for data an infinite amount of time.



PROBLEMS OF STOP-AND-WAIT PROTOCOL

2. Problems due to lost ACK.
   - ★ Sender waits for an infinite amount of time for ack.

NOISY CHANNELS:

Although its predecessor, known as the Stop-and-Wait Protocol, demonstrates how to implement flow control, noiseless channels are non-existent. Either we add error control to our protocols, or we choose to ignore the error that occurred. These are protocols which considered the existence of transmission errors.

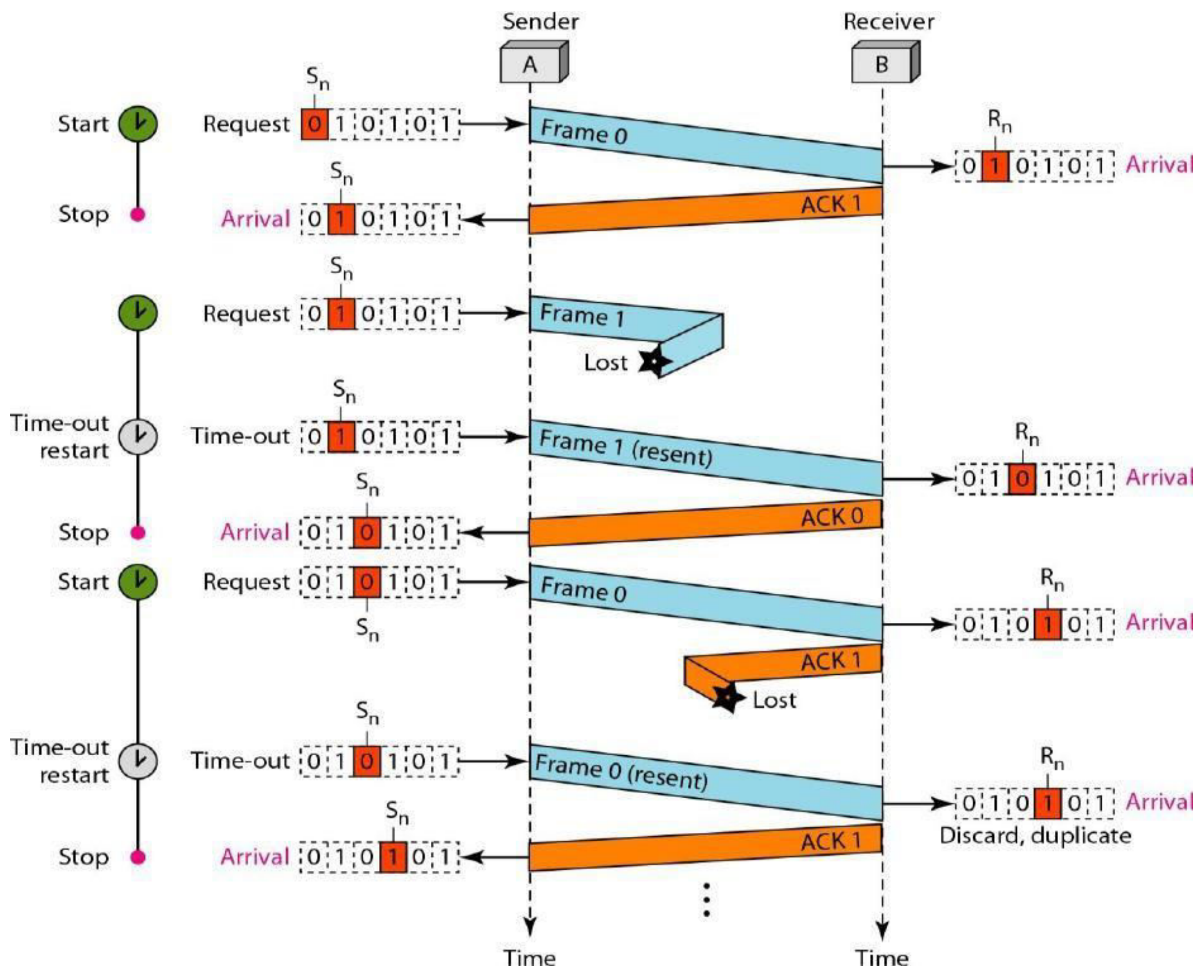
1 Stop-and-Wait Automatic Repeat Request

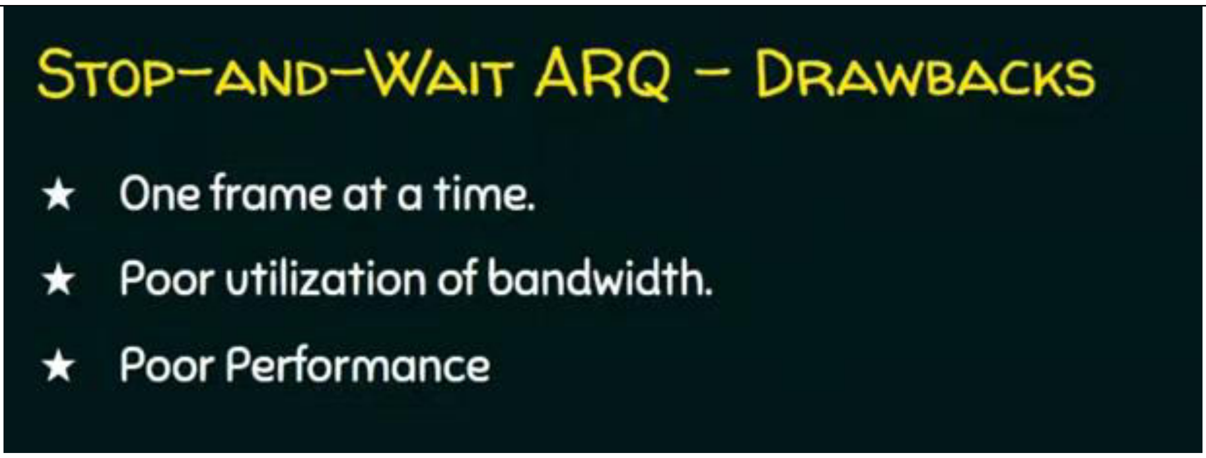
2 Go-Back-N Automatic Repeat Request

3 Selective Repeat Automatic Repeat Request

Stop-and-Wait Automatic Repeat Request:

We must include redundancy bits in our data frame in order to identify and correct defective frames. The frame is examined upon arrival to the recipient site, and if it is corrupted, it is quietly deleted. The receiver's silence indicates that there are errors in this frame. Handling lost frames is more challenging than handling corrupted ones. The frame that was received can be an out-of-order frame, a duplicate, or the proper one. Assigning a sequence number to the frames is the answer. How does the sender determine which frame to transmit if the recipient is silent about an error? The sender retains a copy of the transmitted frame in order to fix this issue. It also sets off a timer at the same moment. The frame is resent, the copy is retained, and the countdown is repeated if the timer ends without an acknowledgment for the transmitted frame. Because the protocol employs the stop-and-wait technique, an ACK is only required for one particular frame. Sequence numbers are used in Stop-and-Wait ARQ to number the frames. The sequence numbers are derived from arithmetic modulo-2. The acknowledgment number in Stop-and-Wait ARQ always transmits the sequence number of the expected next frame in modulo-2 arithmetic.



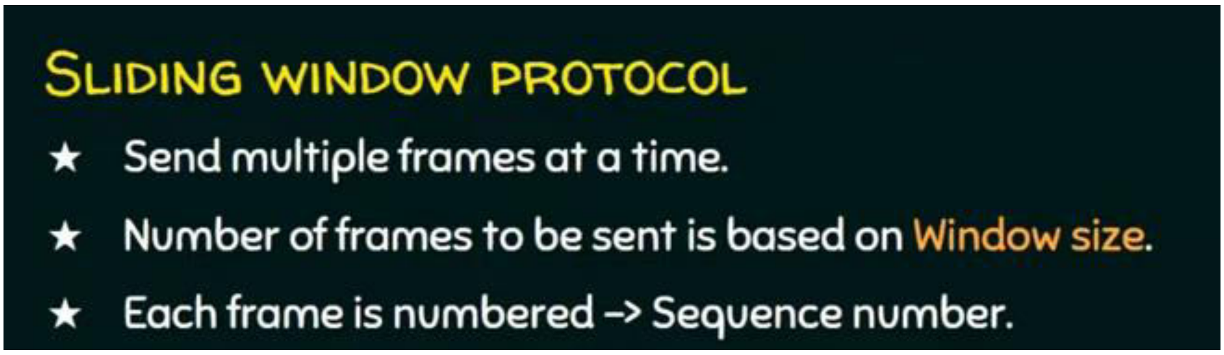


STOP-AND-WAIT ARQ – DRAWBACKS

- ★ One frame at a time.
- ★ Poor utilization of bandwidth.
- ★ Poor Performance

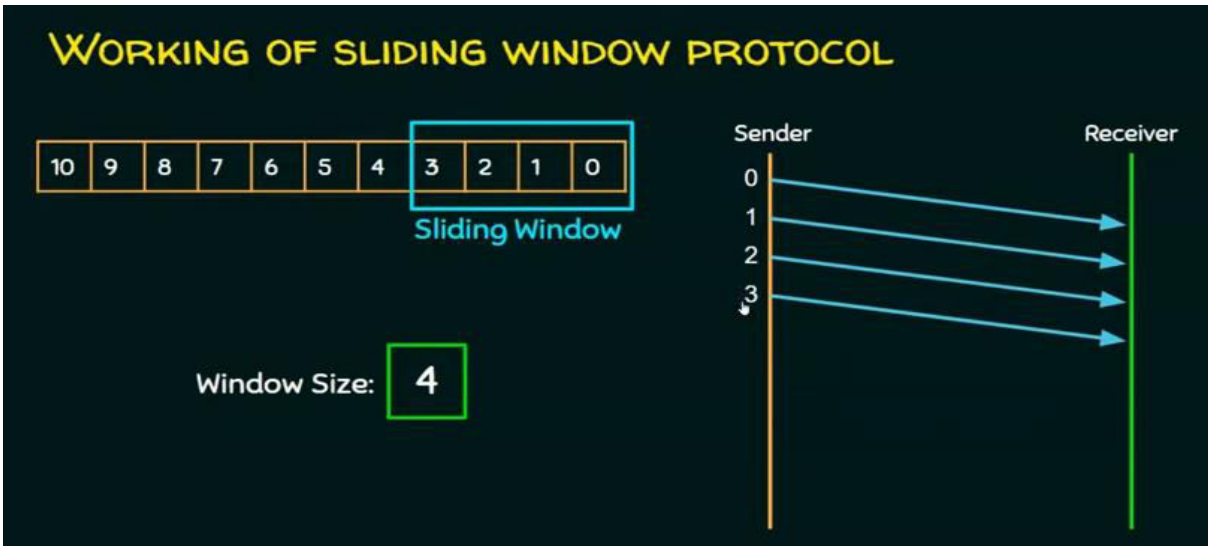
2. Go-Back-N Automatic Repeat Request

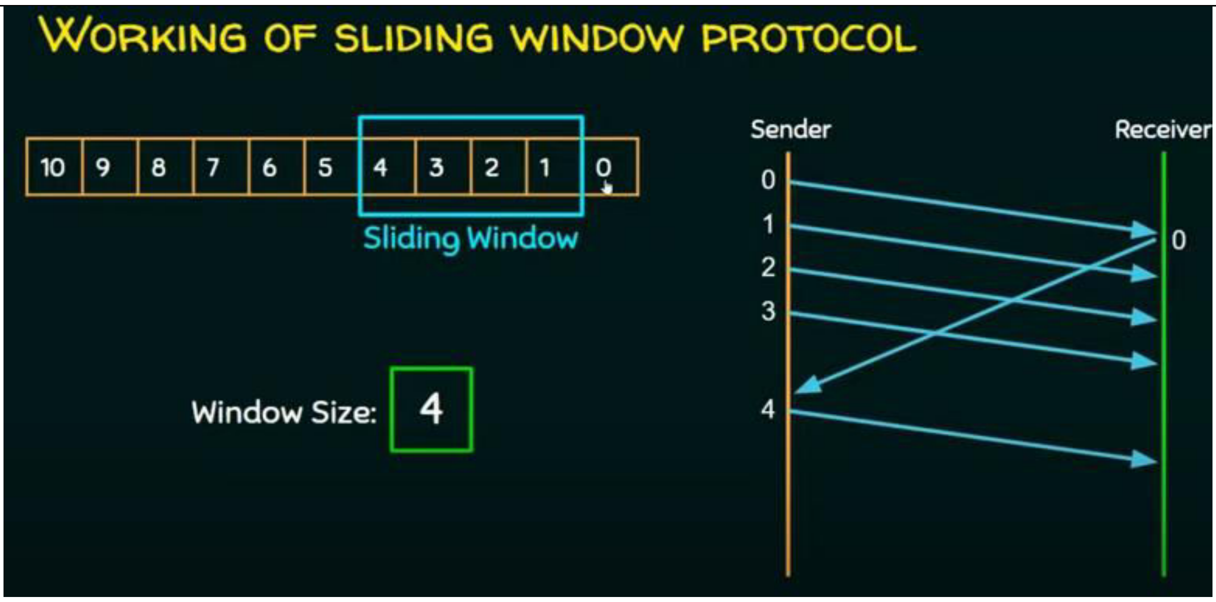
In order to enhance transmission efficacy, it is necessary for multiple frames to be in transition while awaiting acknowledgment. To clarify, it is necessary to permit multiple frames to remain outstanding in order to maintain channel engagement during the sender's waiting period for acknowledgement. The initial option is known as Go-Back-N Automatic Repeat. This protocol permits the transmission of multiple frames prior to the receipt of acknowledgments; a copy of each frame is retained until the acknowledgments are received.

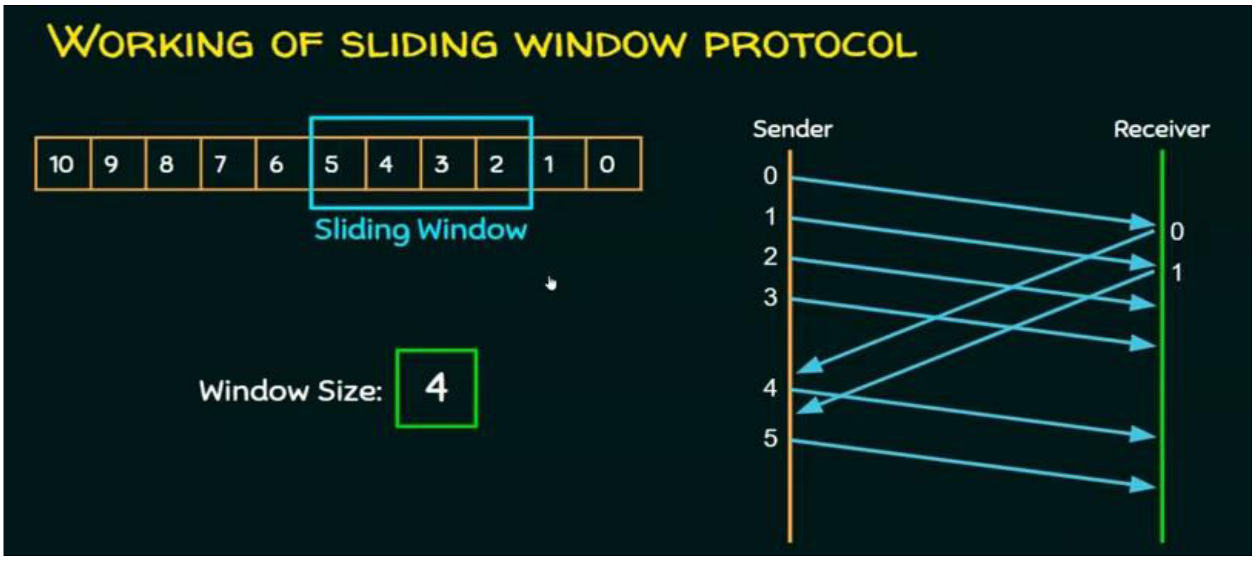


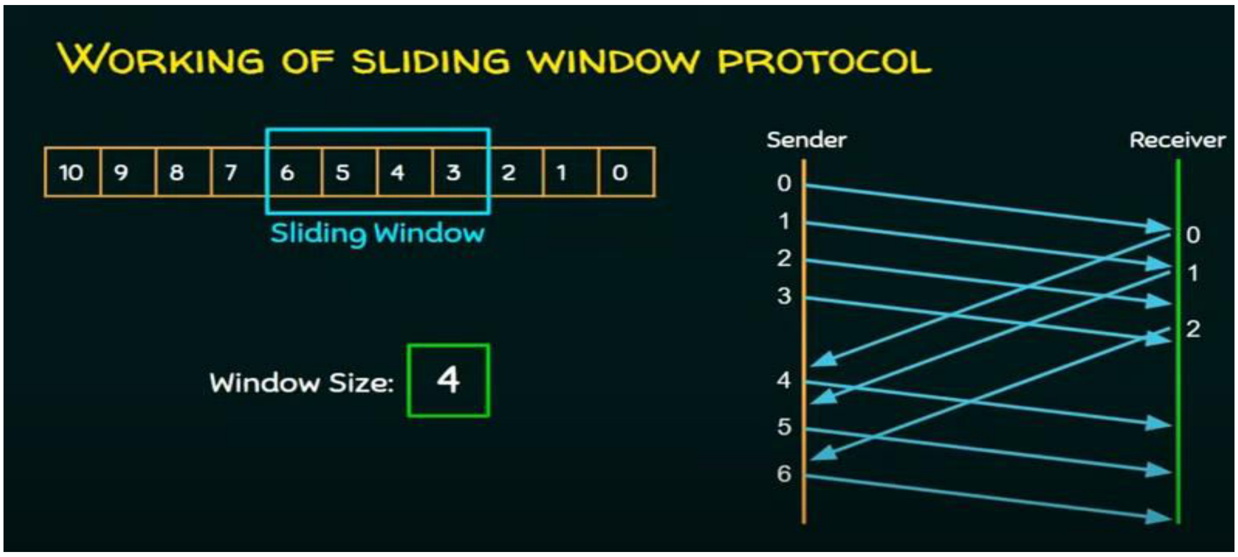
SLIDING WINDOW PROTOCOL

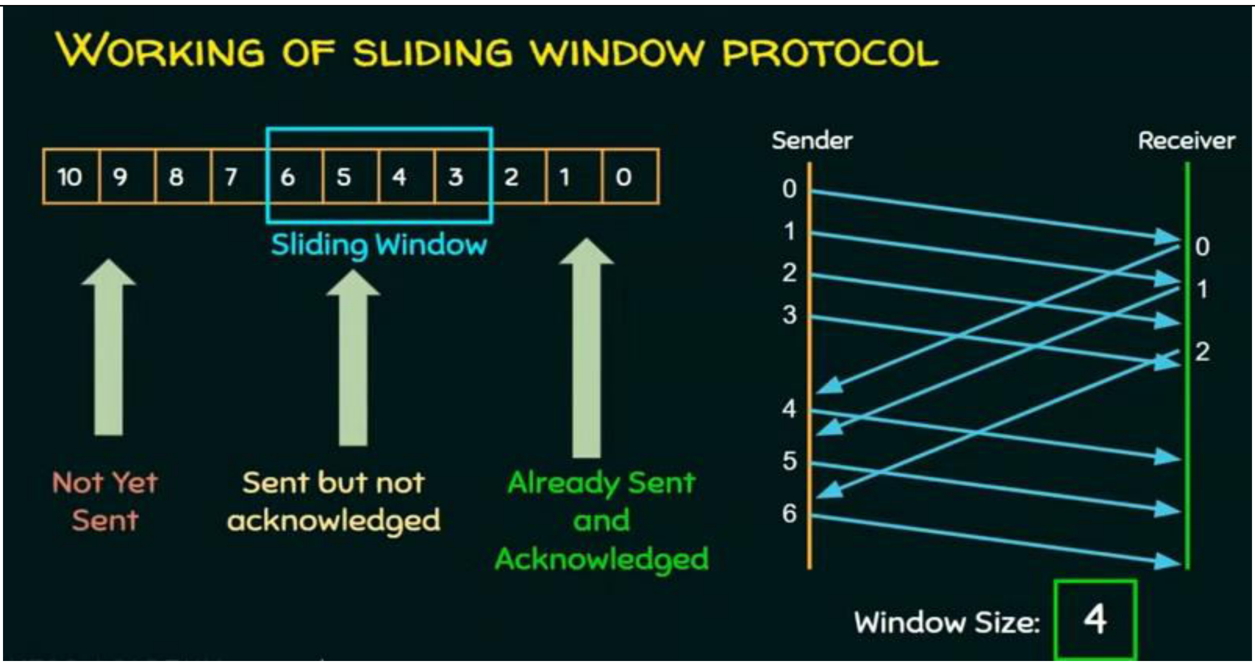
- ★ Send multiple frames at a time.
- ★ Number of frames to be sent is based on Window size.
- ★ Each frame is numbered -> Sequence number.




WORKING OF SLIDING WINDOW PROTOCOL

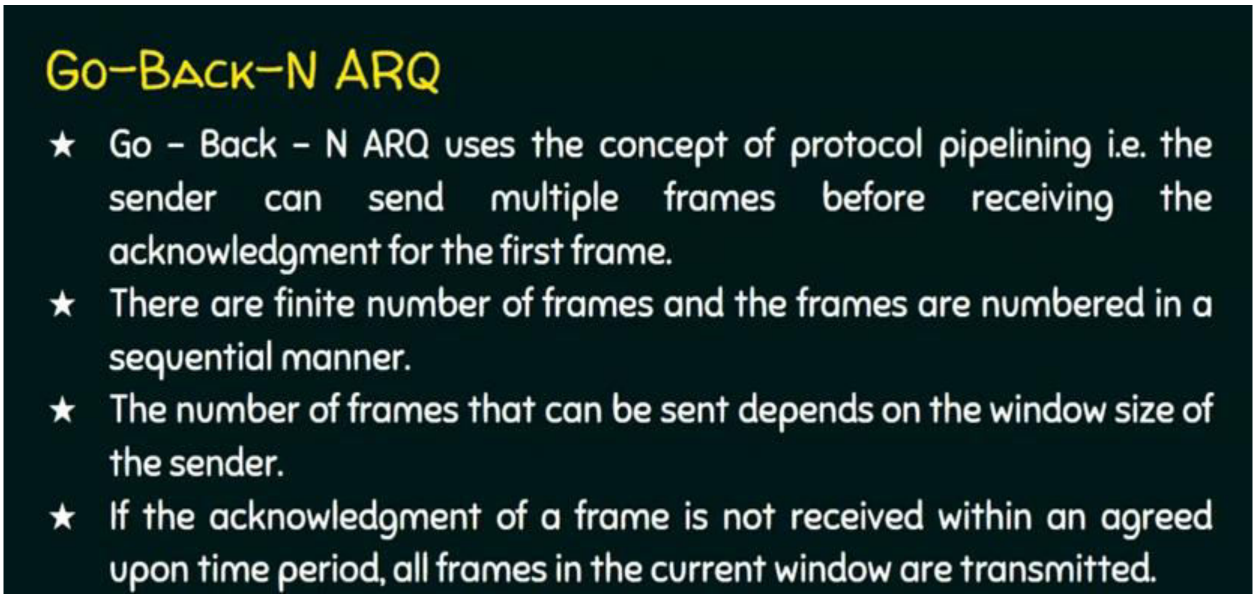

WORKING OF SLIDING WINDOW PROTOCOL


WORKING OF SLIDING WINDOW PROTOCOL



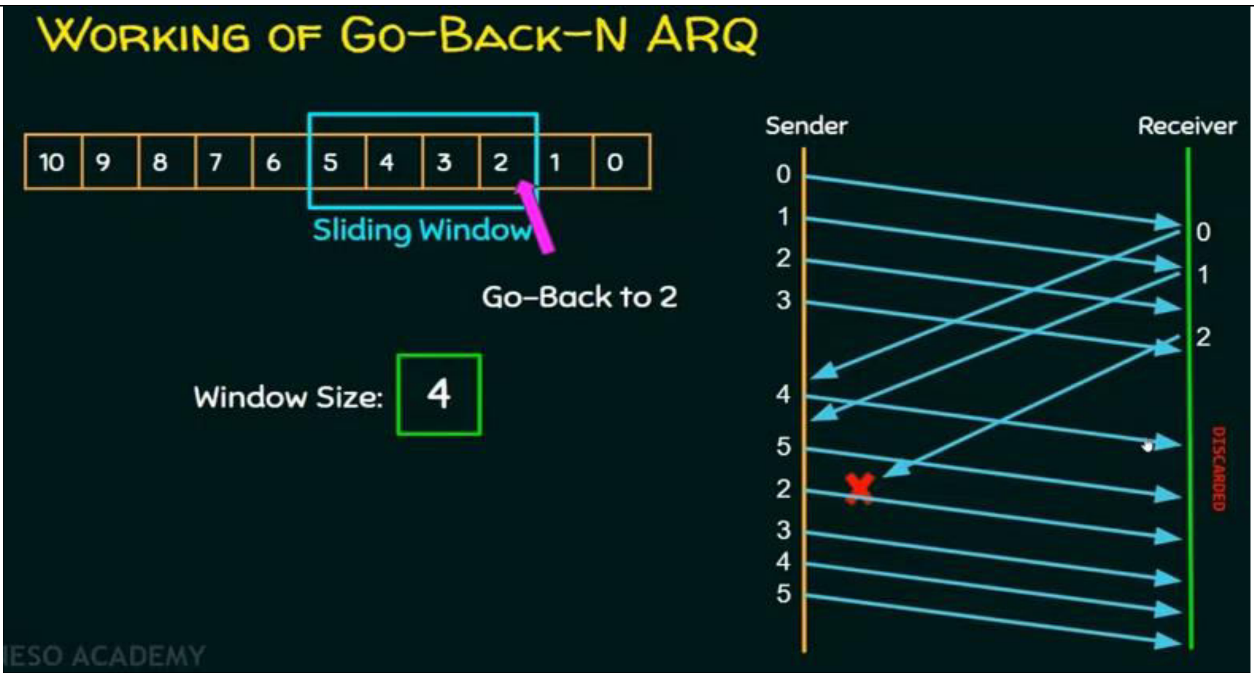
Working of sliding window protocol
10
9
8
7
6
5
4
3
2
1
0
Sliding Window
Not Yet Sent
Sent but not acknowledged
Already Sent and Acknowledged
Sender
Receiver
Window Size: 4

The sequence numbers for frames that have already been acknowledged are shown in the first area to the far right of the window. The sender doesn't concern itself with these frames and doesn't save copies of them. The second area sets the range of sequence numbers for frames that have been sent but not yet acknowledged. The set of sequence numbers for frames that haven't been sent yet is shown in the window's upper left corner.



Go-Back-N ARQ

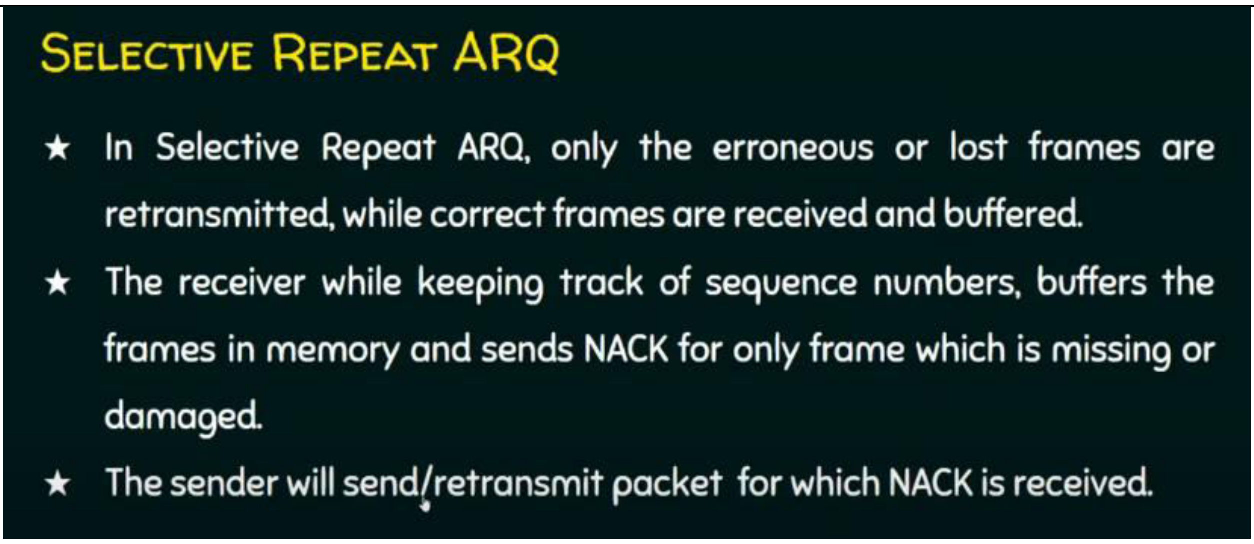
- ★ Go - Back - N ARQ uses the concept of protocol pipelining i.e. the sender can send multiple frames before receiving the acknowledgment for the first frame.
- ★ There are finite number of frames and the frames are numbered in a sequential manner.
- ★ The number of frames that can be sent depends on the window size of the sender.
- ★ If the acknowledgment of a frame is not received within an agreed upon time period, all frames in the current window are transmitted.



If a frame arrives safely and correctly, the user gives a positive acknowledgement. If a frame has been damaged or received out of order, the receiver stays quiet and can throw away all the frames that come after it until it gets the one it wants. The sender's timer for the ignored frame runs out when the receiver stays silent. This makes the sender send all frames again, starting with the one whose timer has already run out.

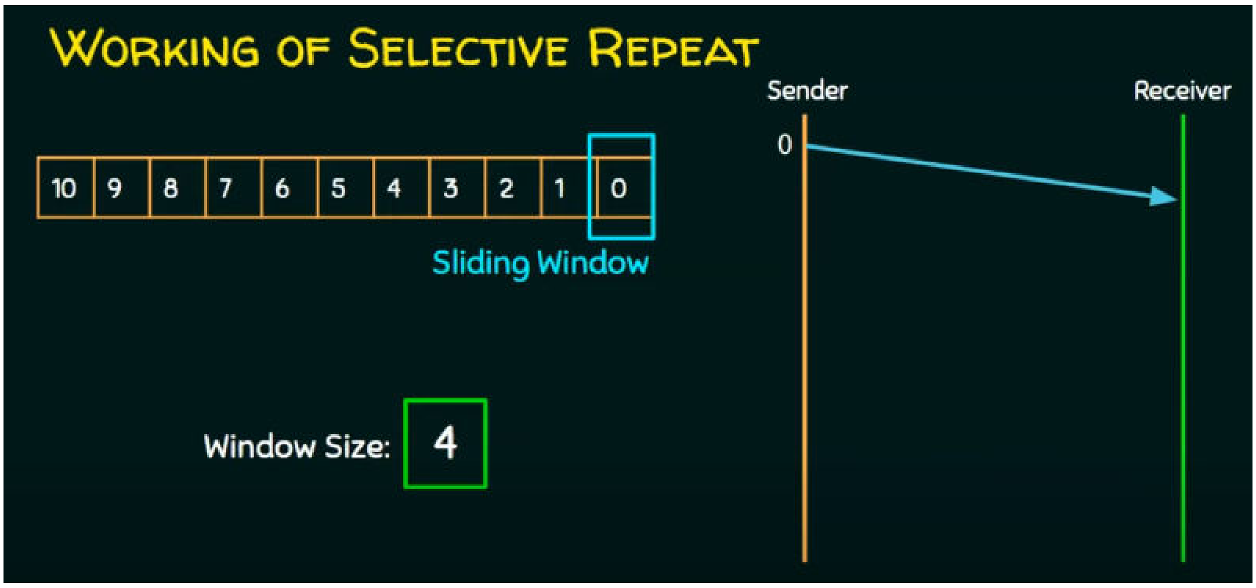
Selective Repeat ARQ:

Out-of-order frames are not stored in Go-Back-N ARQ because the receiver just throws them away. But this technique doesn't work well with a noisy link. In a noisy link, there is a greater chance that a frame will be damaged, which requires sending more than one frame again. This sending again takes up bandwidth and makes the communication go more slowly. There is another way to handle noisy links that doesn't send N frames again when only one has become damaged; only the broken frame is sent again. This method is known as Selective Repeat ARQ. It works better when the link is noisy, but it requires additional efforts at the receiver end.

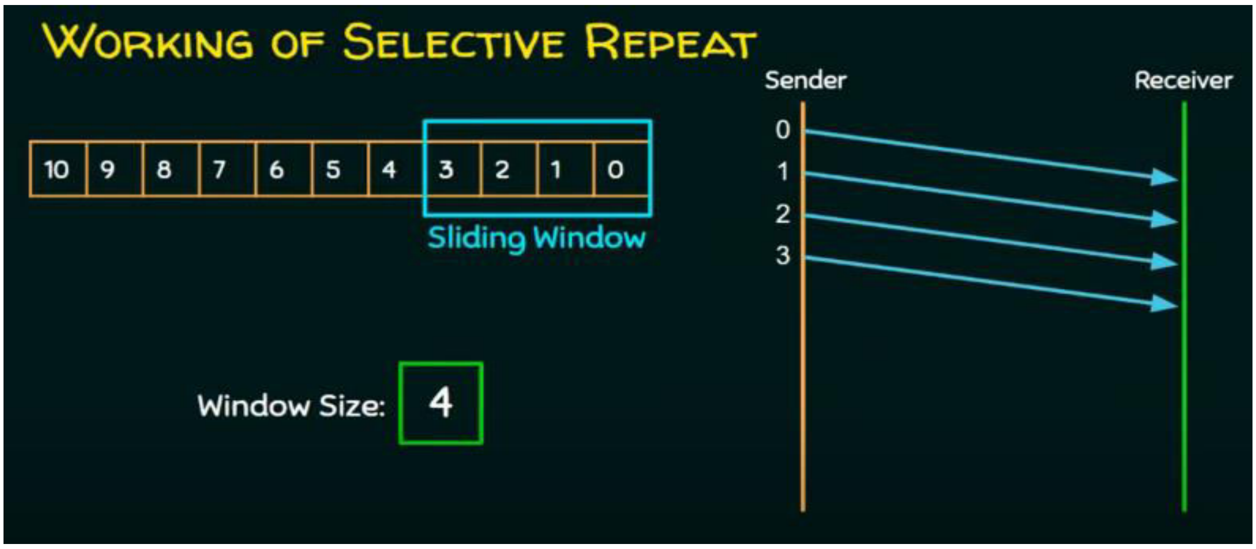


Selective Repeat ARQ

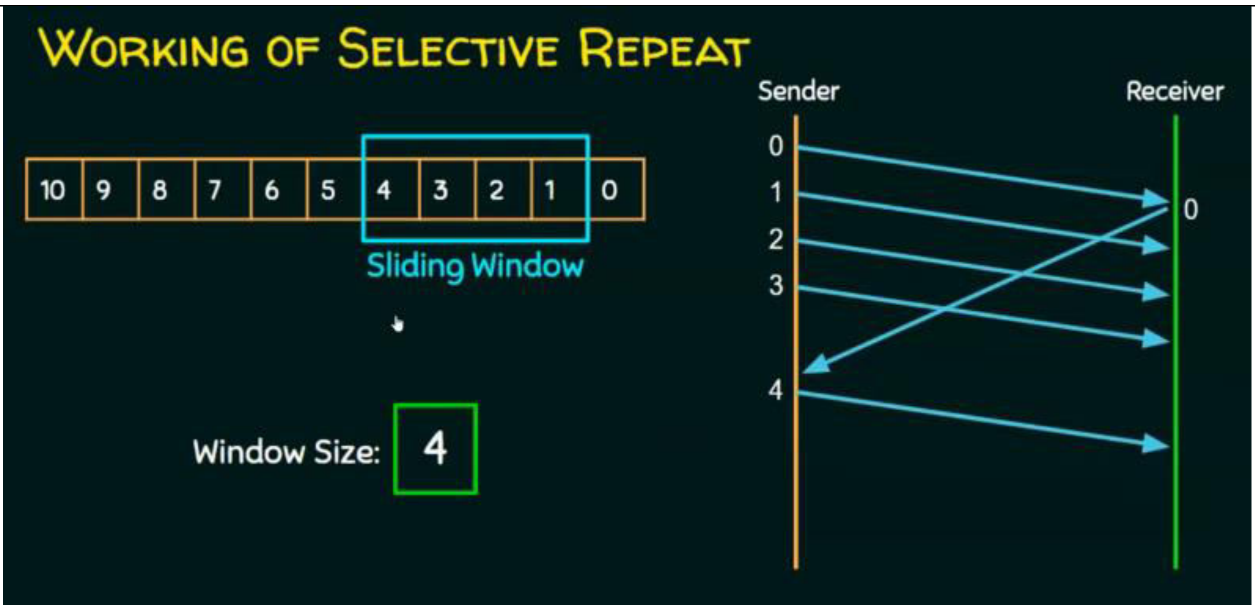
- ★ In Selective Repeat ARQ, only the erroneous or lost frames are retransmitted, while correct frames are received and buffered.
- ★ The receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.
- ★ The sender will send/retransmit packet for which NACK is received.



Working of Selective Repeat



Working of Selective Repeat



Medium Access Control Sublayer

Two types of network connections exist: those that use broadcast channels and those that use point-to-point connections.Random access channels and multi-access channels are other names for broadcast channels. The most important problem in any broadcast network is how to decide who gets the use of the channel in the event of competition. The protocols that decide who gets to talk on a multi-access channel are part of the data link layer's MAC sublayer.In LANs, the MAC sublayer is very crucial.How contending users are to be allocated a single broadcast channel is the central theme of this chapter.In both scenarios, the channel serves as a connection between each user and every other user, and full channel usage disrupts the progress of users who also desire to use the channel.Conventionally, in order to distribute a single channel among numerous contending consumers, its capacity is distributed through the use of frequency division multiplexing.Considerable spectrum will be wasted if the spectrum is divided into N regions and fewer than N users are presently interested in communicating. Some users will be denied permission to communicate if N or more users attempt to do so, due to insufficient bandwidth.This is the static channel allocation issue that is resolved by the dynamic channel allocation issue.In the presence of N users, the bandwidth is partitioned into N portions of equal size, wherein one portion is allocated to each user. Since every user is allocated a unique frequency band, interference among users has been eliminated.A straightforward and effective allocation mechanism is this division when the number of users is small and consistent, and each user has a consistent flow of traffic.When the number of senders is substantial and fluctuating, FDM poses certain challenges.

Prior to discussing the numerous channel allocation methods covered in this chapter, it is essential to grasp the following five fundamental assumptions.

Independent Traffic: The model comprises N autonomous stations, wherein each station is occupied by a user responsible for generating frames intended for transmission.

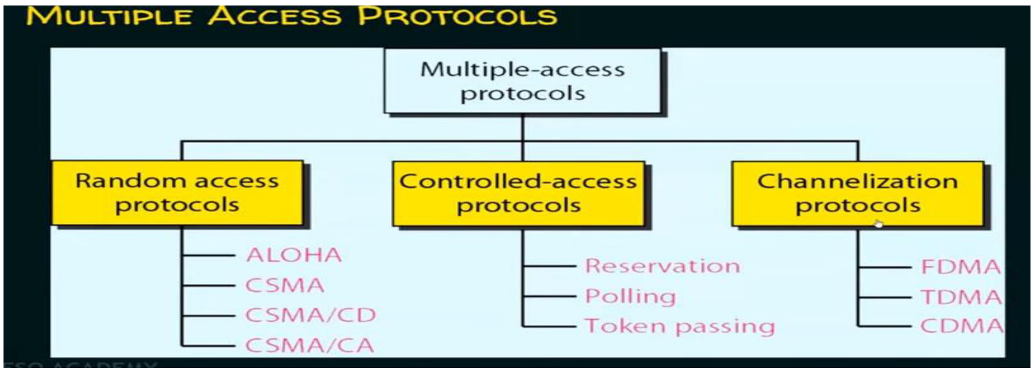
After the generation of a frame, the station enters a blocked state and remains inactive until the frame is transmitted successfully.

Single Channel: All communications are conducted through a single channel. It is capable of transmitting and receiving signals from all stations. While stations may be assigned priorities by protocols, their capabilities have been considered to be equivalent.

Observable collisions: Collisions occur when two frames are transmitted concurrently, leading to a temporal overlap that results in a distorted signal.Each station is capable of identifying a collision. Following a collision, a frame must be retransmitted.

Carrier sense assumption: it enables stations to determine whether a particular channel is in use prior to attempting to make use of it. A station that perceives the channel as active will not attempt to use it. Without carrier sense, stations are unable to detect the channel prior to attempting to use it. They proceed with the transmission. They don't know whether the transmission was successful until a later time.

MULTIPLE ACCESS PROTOCOLS

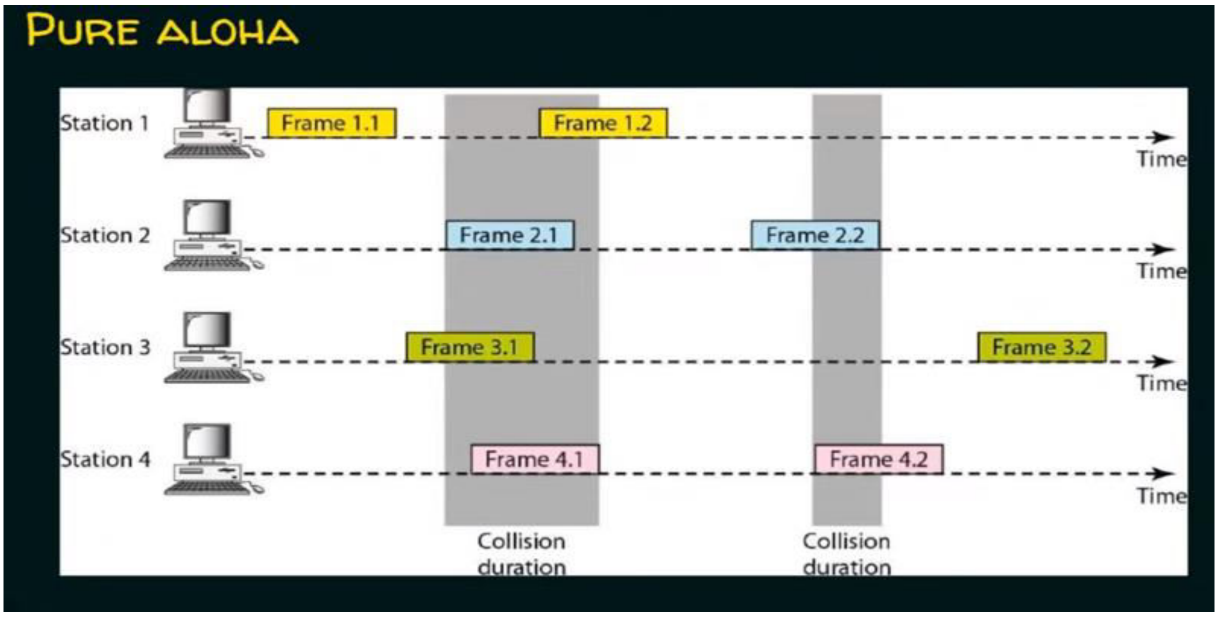


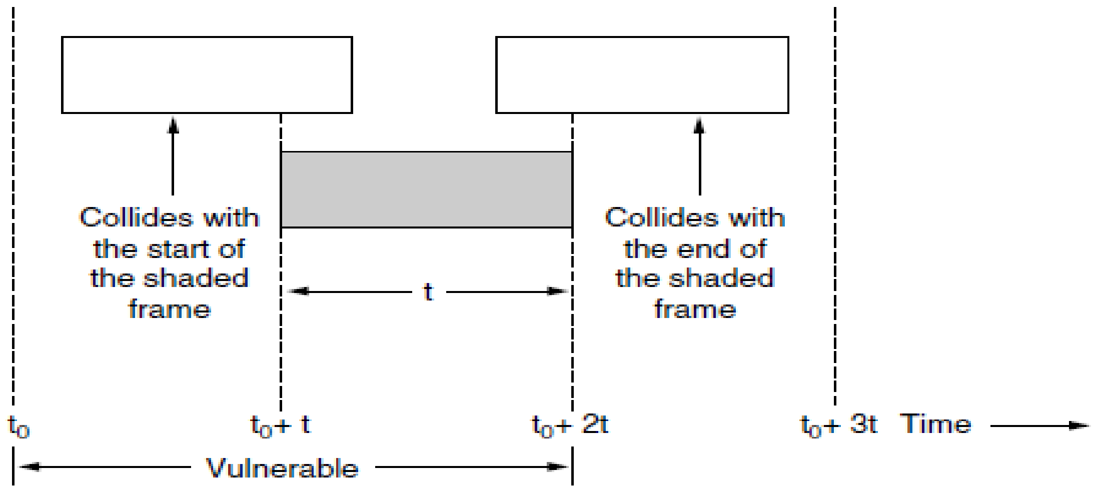
ALOHA:

Norman Abramson and his associates at the University of Hawaii were attempting to establish a connection between the main computer system in Honolulu and users located on remote islands. Short-range radios were employed, whereby each user terminal used an identical upstream frequency to transmit frames to the central computer. The fundamental concept can be implemented in any system wherein uncoordinated users contend for access to a single shared channel. Two variants of ALOHA will be examined in this context: pure and slotted. Distinction arises as to whether time is discrete segments, requiring all frames to suit, or continuous, as in the pure version.

Pure ALOHA:

An ALOHA system's fundamental concept is straightforward: allow users to communicate anytime they have data to send. Naturally, there will be collisions, and the colliding frames will suffer damage.If this is the case, there is a means for senders to learn about it. The sender just waits a random length of time and transmits the frame again if it was destroyed. If the waiting period isn't random, the same frames will keep colliding.A uniform frame size maximises the throughput of ALOHA systems. When two frames attempt to occupy the channel simultaneously, they will collide and become distorted.Both frames will be completely destroyed if the beginning of a new frame and the very last portion of a nearly completed frame overlap.

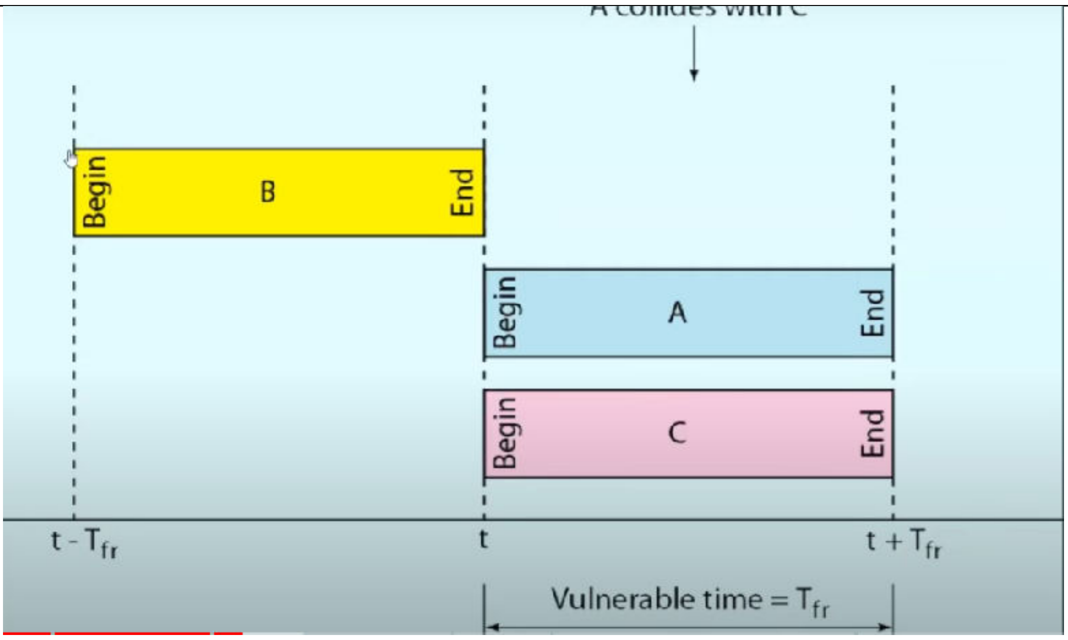


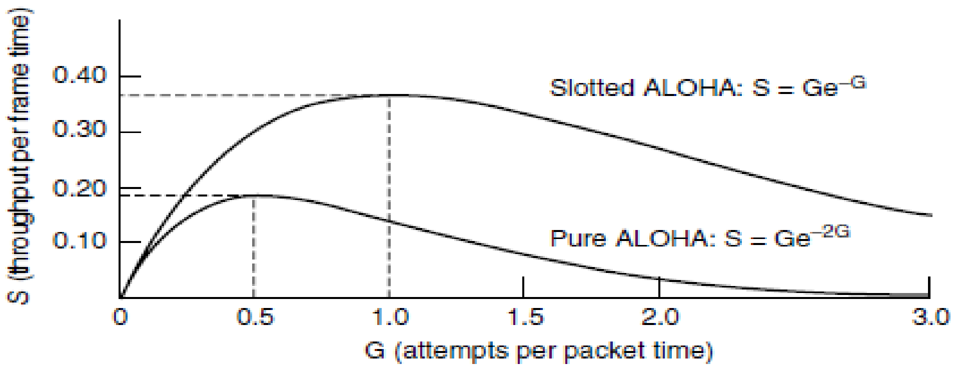

Collides with the start of the shaded frame
Collides with the end of the shaded frame
t
t_0
t_0+ t
t_0+ 2t
t_0+ 3t
Time
Vulnerable

If no further frames are transmitted within one frame after the start of the frame, the frame won't collide.

Slotted ALOHA:

Roberts (1972) presented a technique for doubling the capacity of an ALOHA system shortly after ALOHA was introduced.He suggested partitioning up time into discrete periods known as slots, each of which would represent a single frame. It is not allowed for a station to transmit whenever the user obtains a frame. It is necessary to wait until the start of the next slot instead.This reduces the vulnerable period half.





As you can see, slotted ALOHA has a throughput of S = 1/e, or around 0.368, which is double that of pure ALOHA. It peaks at G = 1. The chance of an empty slot in a system running at G = 1 is 0.368.

Carrier Sense Multiple Access Protocols

The maximum channel utilisation possible with slotted ALOHA is 1/e. This poor result is not unexpected, since there will always be numerous collisions when stations broadcast at random and are unaware of each other's activities.Stations on LANs may often sense what other stations are doing and modify their behaviour appropriately. Compared to 1/e, these networks are capable of much higher utilisation. We will talk about a few procedures for enhancing performance in this part. Carrier sense protocols are those in which stations listen for a carrier, or a signal, and respond appropriately.

Persistent and Nonpersistent CSMA: Here, we shall examine the initial carrier sense protocol, known as 1-persistent CSMA. The simplest CSMA plan. A station listens to the channel to determine whether anybody else is broadcasting at that particular time before sending data. The station transmits its data if the channel is empty. Otherwise, the station just waits for the channel to become idle if it is busy. The station then sends out a frame. In the event of a collision, the station restarts after waiting for an arbitrary period of time.

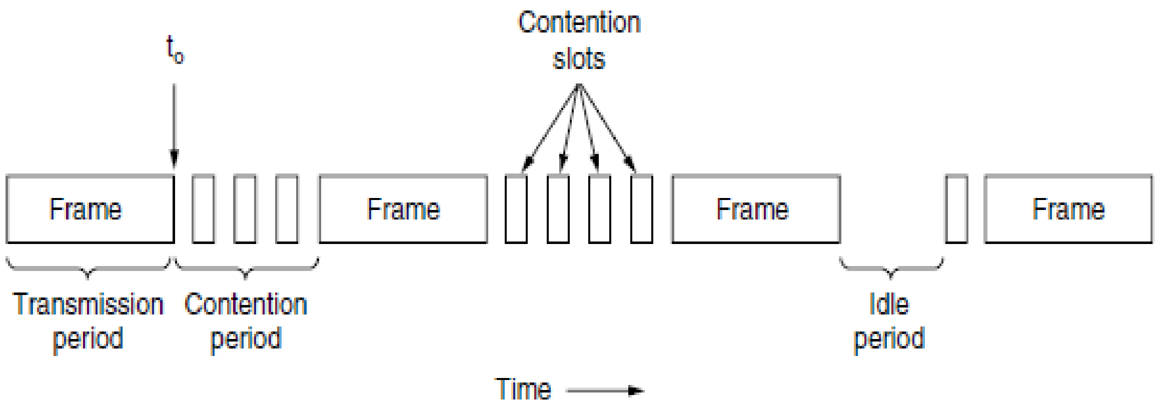
Because the station broadcasts with a probability of 1 when it detects that the channel is idle, the protocol is known as 1-persistent.This protocol exhibits superior performance in comparison to pure ALOHA due to the fact that both stations possess the decency to refrain from interrupting the frame of the third station.

The nonpersistent CSMA protocol is an additional carrier sense method. Compared to the preceding one, an attempt is made to be less selfish in this protocol .When a station wishes to transmit a frame, it observes the channel and initiates transmission if no other station is doing so.

The station does not continuously sense an already-occupied channel in order to seize it immediately after detecting the conclusion of the previous transmission if the channel is in use. However, after a random interval of time has passed, the algorithm is executed again. This algorithm therefore optimises channel utilisation, albeit at the expense of lengthier latency compared to 1-persistent CSMA.A further enhancement involves the stations promptly identifying the accident and suddenly ceasing transmission.

CSMA with Collision Detection: The foundation of the traditional Ethernet LAN is this protocol, which is referred to as CSMA/CD. While the channel is broadcasting, the hardware of the station has to be listening to it. It detects the presence of a collision if the signal it sends out differs from the signal it receives back. Utilising the conceptual model is CSMA/CD.

A station has completed broadcasting its frame at the t_0 point. Now, any other station with a frame to transmit is free to try. A collision will occur if two or more stations choose to broadcast at the same time.A station will stop transmitting if it detects a collision, wait an arbitrary amount of time, and then try again.Thus, we will have alternating congestion and transmission times in our CSMA/CD model, with idle periods happening when all stations are inactive.



Assume that at precisely moment t0, two stations simultaneously start broadcasting. How much time until they realise they've collided?The time it takes for a signal to go from one station to another is the minimal amount of time needed to detect a collision.Stated differently, a station cannot, in the worst situation, be certain that it has taken the channel until it has communicated for two times the propagation time without hearing a collision.

Collision-Free Protocols

This section will look at a few protocols that resolve the channel contention without ever causing a collision, not even during the contention period. Assuming N stations, each uniquely programmed with an address between 0 and N − 1.

A Bit-Map Protocol:

Every contention period in the fundamental bit-map approach, has precisely N slots. Station 0 broadcasts a 1 bit during slot 0 if it has a frame to send. During this time period, no other station is permitted to broadcast. A 1 bit may generally be inserted into slot j by station j to indicate that it has a frame to transmit. Each station knows exactly which other stations want to broadcast once all N slots have elapsed.

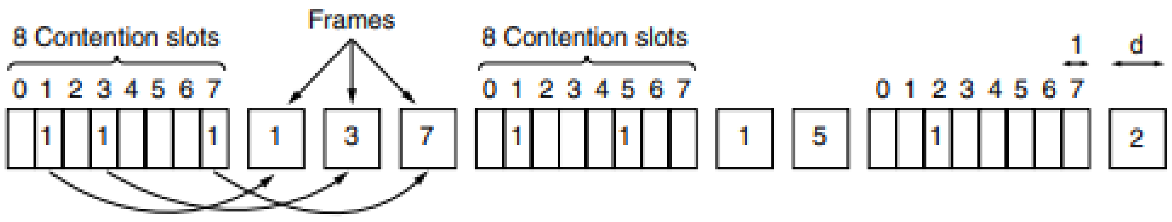


Figure 4-6. The basic bit-map protocol.

There will never be collisions since everyone agrees on who goes first. Another N-bit contention period starts when the final ready station transmits its frame.A station is out of luck if it gets ready just after its bit slot has gone by; it needs to stay quiet until all stations have had a chance and the bit map has rotated once again.

Such protocols, which announce the intention to transmit before the transmission actually happens, are known as reservation protocols as they reserve channel ownership ahead of time and avoid collisions.When there is little load, the bit map will just keep repeating itself as there aren't enough data frames. Examine the scenario from the perspective of a station with a low value, such 0 or 1.

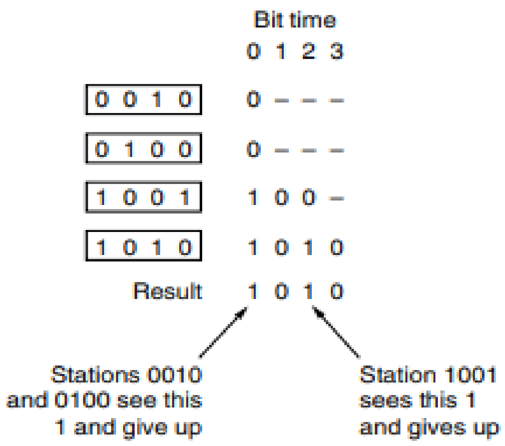
The "current" slot will normally be in the centre of the bit map when it's ready to transmit.Before the station can start broadcasting, it will often need to wait N /2 slots for the current scan to end and other full N slots for the next scan to complete.For high-numbered stations, the future seems more promising. These will often begin transmitting after only half a scan (N /2 bit slots).

Due to the fact that high-numbered stations must wait an average of 0.5N slots and low-numbered stations an average of 1.5N slots.Calculating the channel efficiency under low load is simple. For an efficiency of d /(d + N), the overhead per frame is N bits and the data quantity is d bits.

The N bit contention period is prorated over N frames during high load, when all stations have something to broadcast all the time. This results in an overhead of just 1 bit per frame, or an efficiency of d /(d + 1).The overhead of the standard bit-map protocol is one bit per station, which makes it unsuitable for networks with thousands of stations. We can use binary station addresses to improve upon that.

Binary Countdown:

Now, a station wishing to utilise the channel broadcasts, beginning with the high order bit, its address as a binary bit string. It is expected that every address has the same length. Each address position's bits are BOOLEAN ORed together with bits from various stations. A station quits up as soon as it notices that a high-order bit position that is 0 in its address has been replaced with a 1. For instance, in the first bit period, stations 0010, 0100, 1001, and 1010 send 0, 0, 1, and 1 in an attempt to get the channel. To create a 1, they are ORed together. After seeing the 1, stations 0010 and 0100 give up for the current round, realizing that a station with a higher number is competing for the channel. Proceed to stations 1001 and 1010.After bit zero, both stations carry on. The station 1001 loses up when it reaches bit 1. Station 1010 is victorious due to its highest address. It may now send a frame after winning the auction, and then another bidding cycle begins.

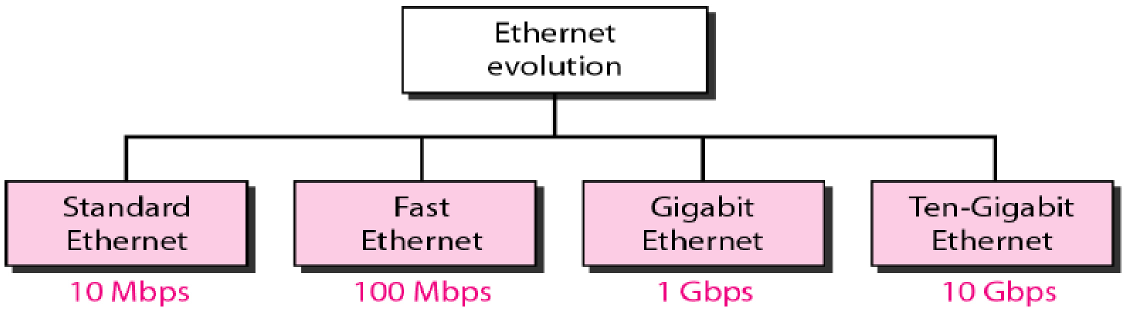


Its feature is that stations with higher numbers are given preference over ones with lower numbers.

This method's channel efficiency is d /(d + $\log_2 N$).

Ethernet

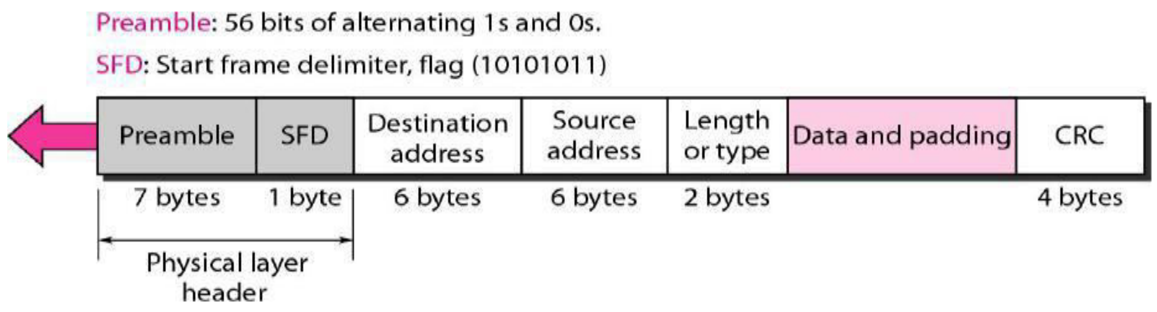
The Xerox company invented the first Ethernet. It has undergone four generations since then. Gigabit Ethernet (1 Gbps), Ten-Gigabit Ethernet (l0 Gbps), Fast Ethernet (100 Mbps), and Standard Ethernet (l0 Mbps), This section briefly discusses standard (or conventional) Ethernet.



The MAC sublayer in Standard Ethernet controls how the access method works. Additionally, it transfers data to the physical layer after framing data received from the higher layer.

Ethernet Frame Format:

Preamble, SFD, DA, SA, length or kind of protocol data unit (PDU), upper-layer data, and the CRC are the seven fields that constitute an Ethernet frame. Ethernet is referred to be an unreliable media as it lacks a method for acknowledging received frames. Higher levels need to incorporate acknowledgments. The image below illustrates the MAC frame format.



Preamble: Seven bytes, or 56 bits, of alternating 0s and 1s make up the first field of an 802.3 frame. This allows the receiving system to synchronise its input time and be informed that a new frame is about to arrive.

Start frame delimiter (SFD): The start of the frame is shown by the second field (l byte: 10101011). The SFD tells the station or stations that this is their last chance to get in sync. The last two bits are 11 and let the recipient know that the next field is the destination's address.

Destination address (DA): The DA field is made up of 6 bytes and has the actual address of the station that will receive the message.

Source address (SA): The physical address of the packet's sender is included in the six-byte SA field.

Length or type: This field served as the type field in the original Ethernet, which defined the upper-layer protocol via the MAC frame. It served as the length field in the IEEE standard to specify how many bytes were in the data field.

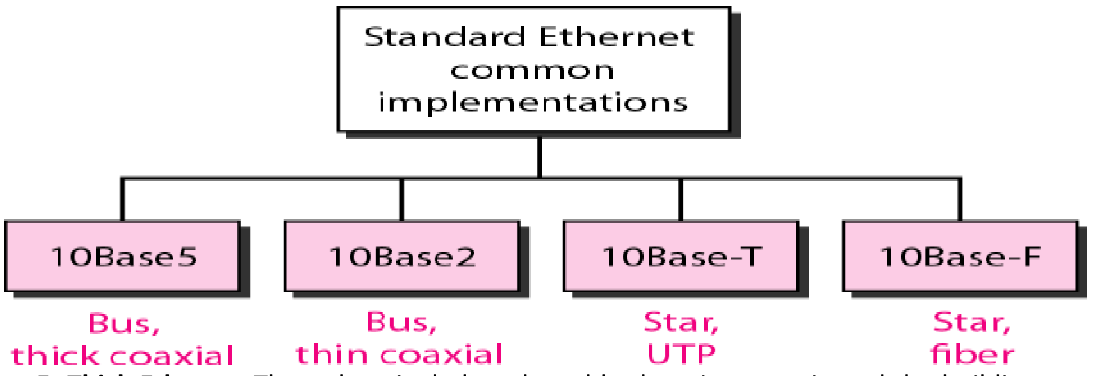
Data and padding: This field contains data that has been encapsulated from upper-layer protocols. The least amount is 46 bytes and the most is 1500 bytes.To compensate when the upper-layer payload is shorter than 46 bytes, padding is appended.

CRC: This field facilitates error detection by using CRC-32

To compensate when the upper-layer payload is shorter than 46 bytes, padding is appended.

CSMA/CD is used as MAC method in Ethernet.

The physical layer implementations specified by the Standard Ethernet are various; four of the most prevalent are illustrated in Figure.



10Base5: Thick Ethernet: Through a single lengthy cable that circumnavigated the building, every computer was connected to traditional Ethernet. The initial variant was known as "thick Ethernet." A yellow garden hose with every 2.5 metres of markings denoting computer attachment locations.

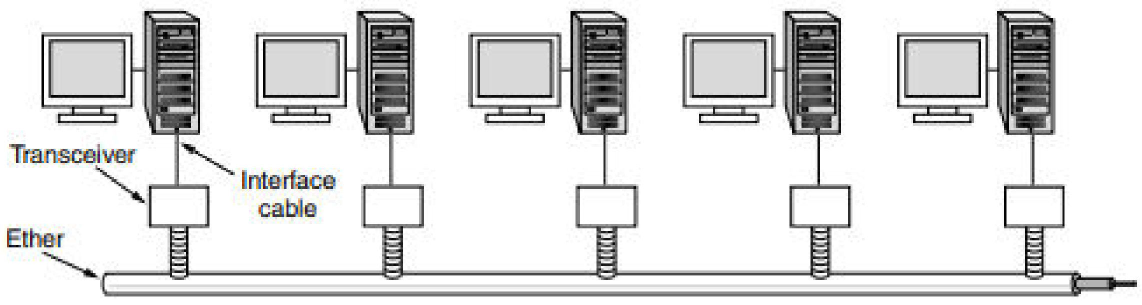
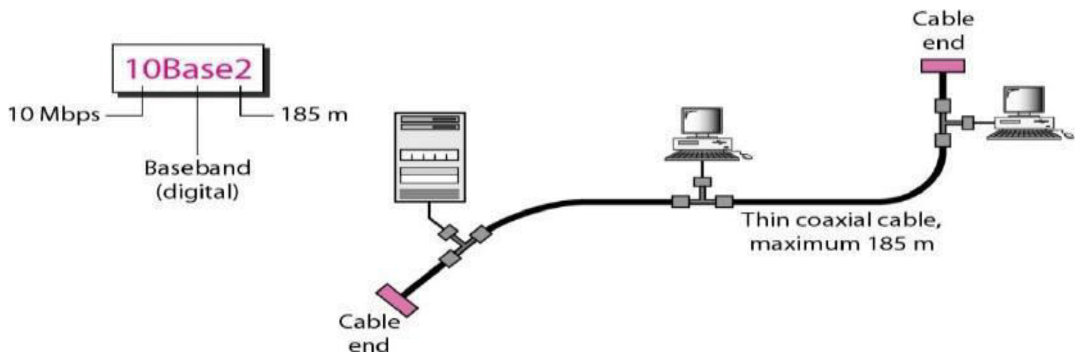


Figure 4-13. Architecture of classic Ethernet.

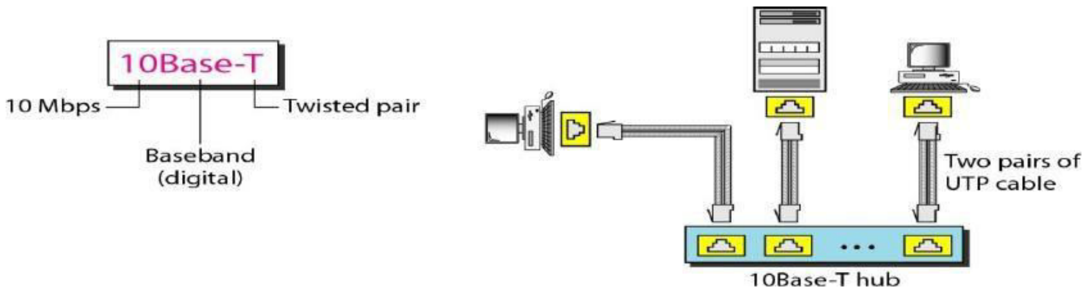
Thin Ethernet: The 10 Base2 implementation is alternatively referred to as Cheapernet or thin Ethernet. Although 10Base2 also employs a bus topology, the cable is considerably more flexible and thin. The schematic diagram of a 10Base2 implementation is depicted in the figure.



Thicker coaxial cable is more expensive than its thinner counterpart.Due to the extreme flexibility of the thin coaxial cable, installation is simplified.Due to the extensive attenuation in thin coaxial cable, the maximum length of each segment is 185 metres (approximately 200 metres).

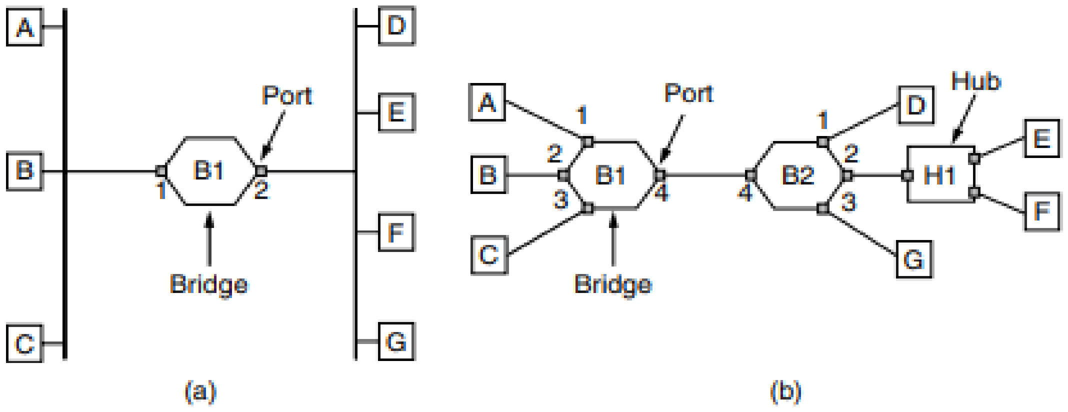
Twisted-Pair Ethernet:

10Base-T, also known as twisted-pair Ethernet, denotes the third scheme. A physical star topology is implemented. As shown in Figure, the stations are linked to a node via two pairs of twisted cable.A 100-meter maximum length is specified for the twisted cable in order to minimise the effect of attenuation in the cable.



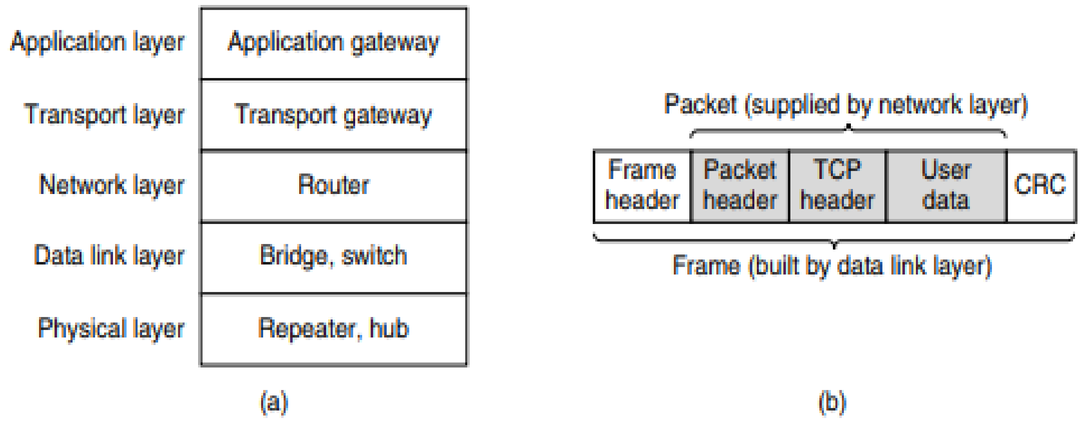
Datalink Layer Switching

Many organisations have multiple LANs and desire to link them.This is possible while establishing connections using apparatus known as bridges.Ethernet switches are bridges in the modern sense. Bridges function at the stratum of data links. In order to combine multiple physical LANs into a single logical LAN, bridges are utilised.



When station A transmits a frame to station B, bridge B1 will receive the frame via port 1. This particular frame may be promptly disregarded since it is already connected to the appropriate port.

In the depicted architecture shown in Figure (b), let us consider the scenario when node A transmits a frame to node D. Bridge B1 will receive the frame via port 1 and thereafter transmit it via port 4. The frame will be received by Bridge B2 on port 4 and then sent on port 1.



Repeater devices are situated in the lowermost layer, known as the physical layer. The aforementioned devices are analogue in nature and operate by manipulating signals sent via the connections to which they are linked. The signal present on a given cable undergoes a process of signal conditioning, including cleaning, amplification, and subsequent transmission onto another cable. Repeaters lack comprehension of frames, packets, and headers. The individuals possess comprehension of the symbols that serve as representations for bits via voltage encoding.A hub is equipped with many input lines that are electrically interconnected. Frames that are received on any of the lines are subsequently sent on all the other lines. In the event that two frames are received simultaneously, a collision will occur, similar to the scenario seen in a coaxial cable network.

A bridge serves as a means of interconnecting many Local Area Networks (LANs). Similar to a central hub, a contemporary bridge has several ports, often accommodating a range of 4 to 48 input lines of a certain kind. In contrast to a hub, each port in a network switch is designed to function as an independent collision domain.Thephrase "switch" has gained significant popularity in contemporary times. In contemporary installations, point-to-point connections are often used, whereby individual computers are directly connected to a switch, resulting in the switch typically possessing several ports.Next, we go to the topic of routers. Upon arrival of a packet at a router, the frame header and trailer are removed, and the packet contained inside the payload field of the frame is thereafter sent to the routing software. The software uses the packet header information in order to determine the appropriate output line.At a higher hierarchical level, transport gateways are located.

These devices establish a link between two computers that use dissimilar connection-oriented transport protocols.For instance, consider a scenario where a computer using the connection-oriented TCP/IP protocol is required to establish communication with another computer utilising a distinct connection-oriented transport protocol known as SCTP. The transport gateway has the capability to duplicate packets from one connection to another, while also adjusting their format as required.Application gateways has the capability to comprehend the structure and substance of data, enabling them to effectively convert messages from one format to another.