BUDDHA SERIES

(Unit Wise Solved Question & Answers)

Course – B.Tech. (CSE Allied-AIML/DS)

College – Buddha Institute of Technology

(AKTU CODE-525)

Department: Computer Science & Allied-
PROGRAM: AIML-DS

Subject: Computer Networks
(BCS 603)

Faculty Name: Mr. Shailesh Kumar Patel

Unit - 2

Q 1. Explain Selective Repeat ARQ protocols. What are the advantages of Selective Repeat ARQ over Go-Back-N ARQ protocol? **(AKTU 2022-23)**

Solution:

Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.

Windows: The Selective Repeat Protocol also uses two windows: a send window and a receive window. However, there are differences between the windows in this protocol and the ones in Go-Back-N. First, the size of the send window is much smaller; it is 2^m-1. The reason for this will be discussed later. Second, the receive window is the same size as the send window.

The send window maximum size can be 2^m-1. For example, if m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol). The smaller window size means less efficiency in filling the pipe.

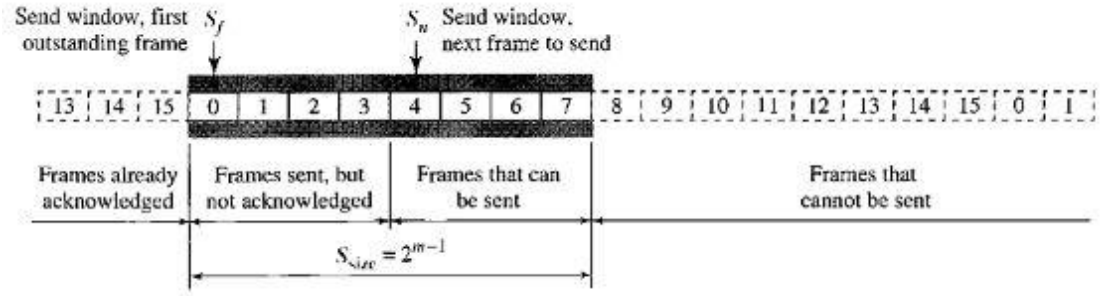


Figure: Send window for Selective Repeat ARQ

The receive window in Selective Repeat is totally different from the one in Go-Back-N. First, the size of the receive window is the same as the size of the send window(2^m-1). The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer. Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered.

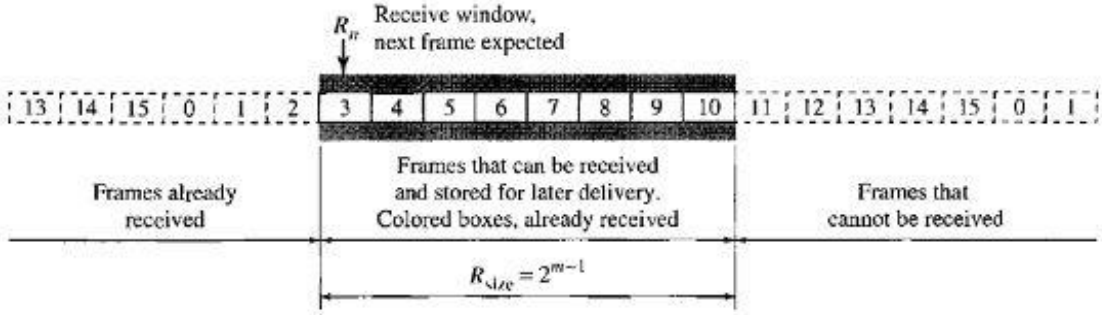


Figure: Receive window for Selective Repeat ARQ

Design The design in this case is to some extent similar to the one we described for the Go- Back-N, but more complicated.

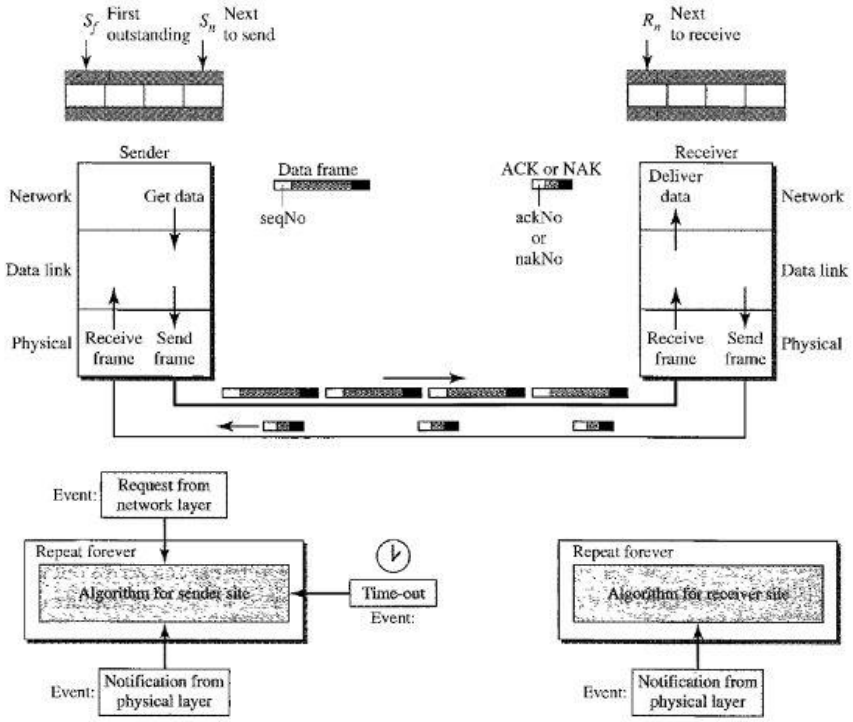


Figure: Design of Selective Repeat ARQ

Window Sizes the size of the sender and receiver windows must be at most one half of 2^m.

If the rate of error is high, then Go-Back-N will consume a lot of bandwidth. Selective Repeat is a better option if you have to be considering bandwidth requirement, as it would resend only the defective or missing packets and not the entire windows.

The basic difference between go-back-n protocol and selective repeat protocol is that the "go-back-n protocol" retransmits all the frames that lie after the frame which is damaged or lost. The "selective repeat protocol" retransmits only that frame which is damaged or lost.

Q 2. What is piggybacking? **(AKTU 2017-18, 2024-25)**

Solution:

In a real-life network, the data link protocols are implemented as bidirectional; data flow in both directions. In these protocols the flow and error control information such as ACKs and NAKs is included in the data frames in a technique called piggybacking.

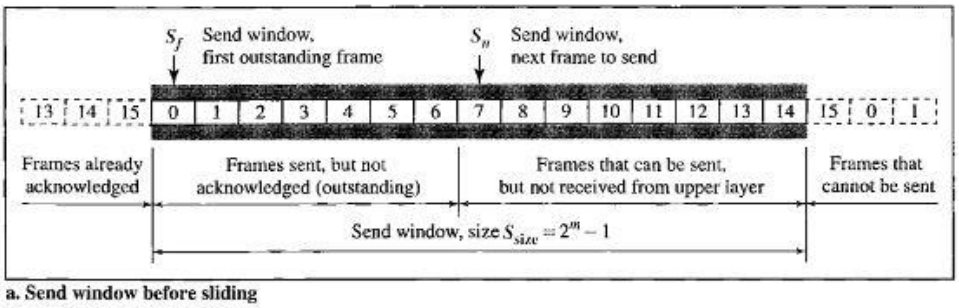
Q 3. Explain Sliding window protocol for Go-Back-N ARQ. How selective repeat ARQ is different from GO-BACK-N. **(AKTU 2022-23)**

Solution:

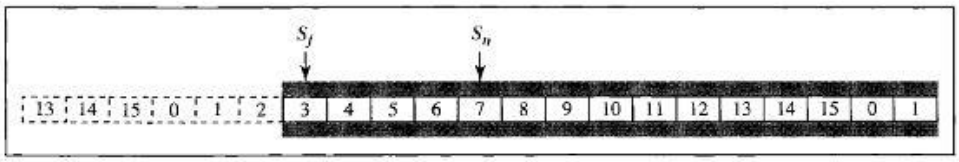
Go-Back-N Automatic Repeat Request: To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

Sequence Numbers Frames from a sending station are numbered sequentially. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2^m- 1. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive.

Sliding Window The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receive sliding window.
The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit. In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent. The maximum size of the window is $2^m - 1$.
Figure shows a sliding window of size 15 (m = 4). The window at any time divides the possible sequence numbers into four regions. The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged.
The sender does not worry about these frames and keeps no copies of them.

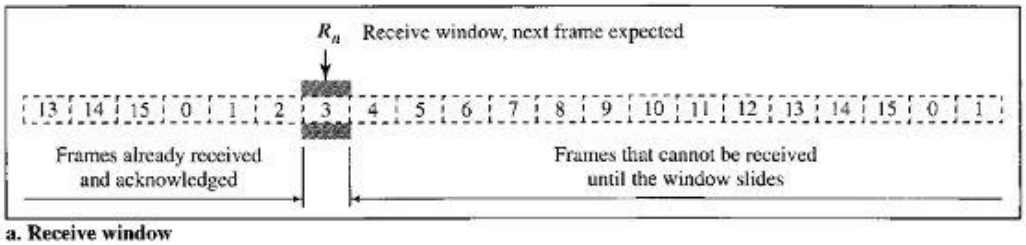


a. Send window before sliding

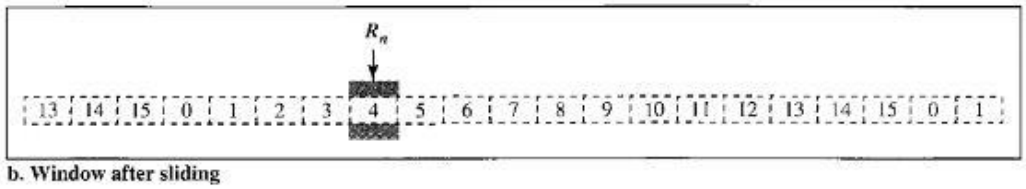


b. Send window after sliding

The second region, colored in Figure (a), defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames. The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer. Finally, the fourth region defines sequence numbers that cannot be used until the window slides.
The send window is an abstract concept defining an imaginarybox of size 2^m -1 with three variables: S_f,S_n, and S_{size}.
We call these variables S_f(send window, the first outstanding frame), S_n (sendwindow, the next frame to be sent), and S_{size} (send window, size). The variable S_f definesthe sequence number of the first (oldest) outstanding frame. The variable S_n holds thesequence number that will be assigned to the next frame to be sent. Finally, the variableS_{size} defines the size of the window, which is fixed in our protocol.
Figure (b) shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. In this figure, frames 0, 1, and 2 are acknowledged,so the window has slid to the right three slots. Note that the value of S_f is 3 because frame 3is now the first outstanding frame.
The receive window makes sure that the correct data frames are received and thatthe correct acknowledgments are sent. The size of the receive window is always 1. Thereceiver is always looking for the arrival of a specific frame. Any frame arriving out oforder is discarded and needs to be resent. Following figure shows the receive window.



a. Receive window



b. Window after sliding

We need only one variable R_n (receive window, next frame expected) todefine this abstraction. The sequence numbers to the left of the window belong to theframes already received and acknowledged; the sequence numbers to the right of thiswindow define the frames that cannot be received. Any received frame with a sequencenumber in these two regions is discarded. Only a frame with a sequence number matchingthe value of R_n is accepted and acknowledged.The receive window also slides, but only one slot at a time. When a correct frameis received (and a frame is received only one at a time), the window slides.

Timers There can be a timer for each frame that is sent; in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

Acknowledgment The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.

Resending a Frame When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again. That is why the protocol is called Go-Back-N ARQ.

Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. **This mechanism is called Selective Repeat ARQ. It is more efficient for noisy links**, but the processing at the receiver is more complex.

Q 4. Given the dataword 1010011010 and the divisor 10111. **(AKTU 2021-22, 2022-23)**

i) Show the generation of the codeword at the sender site (using binary division).

ii) Show the checking of the codeword at the receiver site (assume no error).

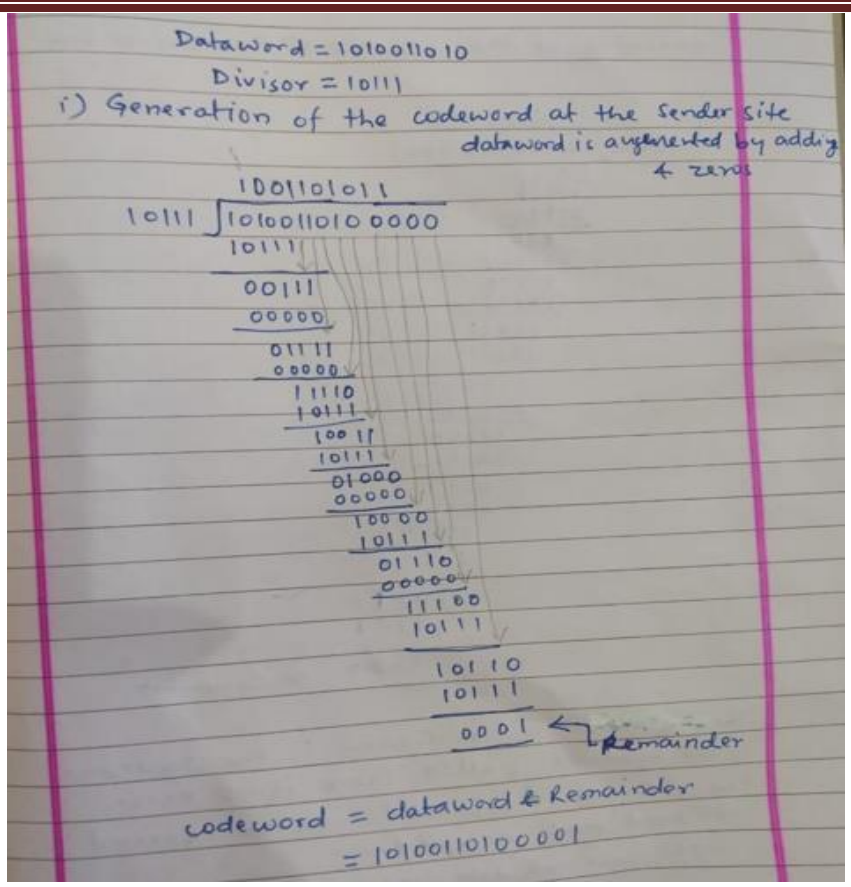
Solution:

Dataword = 1010011010

Divisor = 10111

i) Generation of the codeword at the sender site

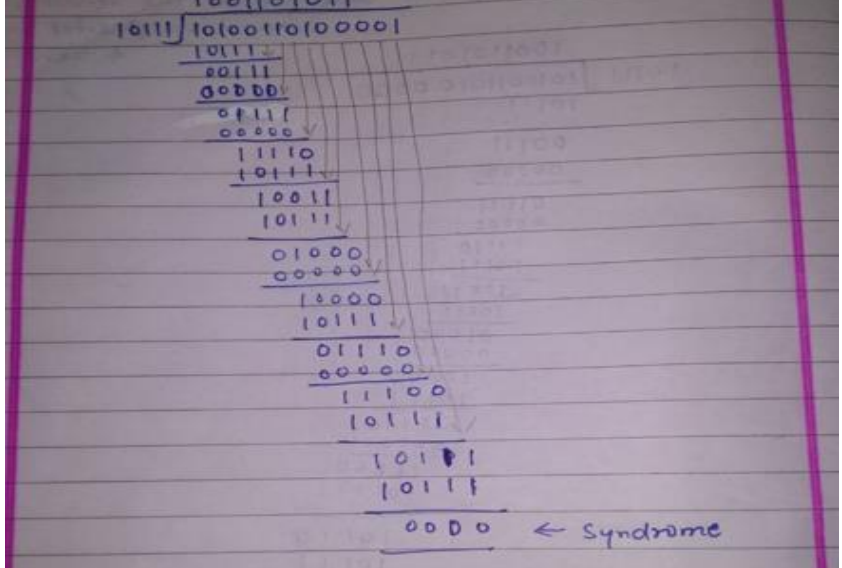
dataword is augmented by adding 4 zeros



codeword = dataword & Remainder

= 1010011010 0001

checking of the codeword at the receiver site



The remainder of the division is the syndrome. If the syndrome is all 0s, there is no error. The dataword is separated from the received codeword and accepted.

The received data word = 1010011010

Q 5.

If the 7-bit Hamming codeword received by a receiver is 1110101. Assuming the even parity, state whether the received codeword is correct or incorrect. If incorrect, locate the position of error. What will be the correct code? **(AKTU 2022-23)**

Solution:

D7	D6	D5	P4	D3	P2	P1
1	0	1	1	0	1	1

{ 1011→ to convert in 7 bit, add parity bits…….location of parity bits should be 2^n, Where n=0, 1, 2, 3, ……}

Selection of parity bits:

i) Selection of P1-
P1 is adjusted 0 or 1, to set even parity over bits 1, 3, 5, 7

ii) Selection of P2-
P2 is adjusted 0 or 1, to set even parity over bits 2, 3, 6, 7

iii) Selection of P3-
P3 is adjusted 0 or 1, to set even parity over bits 4, 5, 6, 7

For P1:
P1, D3, D5, D7
1, 0, 1, 1
This show odd parity means error exists.

For P2:
P2, D3, D6, D7
1, 0, 0, 1
This show even parity means no error.

For P4:
P4, D5, D6, D7
1, 1, 0, 1
This show odd parity means error exists.

Therefore, position of error:
P1, P2, P4

P4	P2	P1
1	0	1

5^{th} position is having error i.e. D5
Therefore, correct message is:

1	**0**	**0**	**1**	**0**	**1**	**1**

Q 6. Explain the working of pure ALOHA and slotted ALOHA protocols. How slotted ALOHA improve the performance of pure ALOHA? **(AKTU 2018-19)**

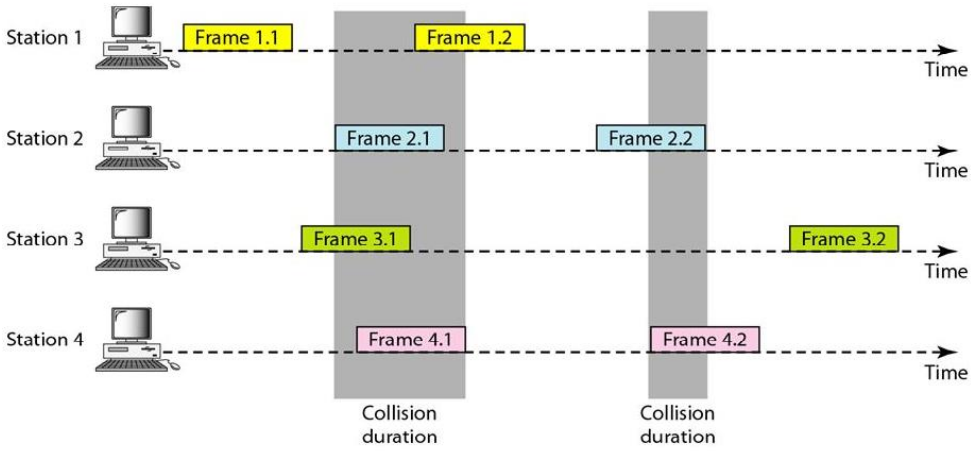
Solution:

ALOHA, the earliest random-access method, was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium. It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

- **Pure ALOHA**

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one

channel to share, there is the possibility of collision between frames from different stations. Fig shows an example of frame collisions in pure ALOHA.



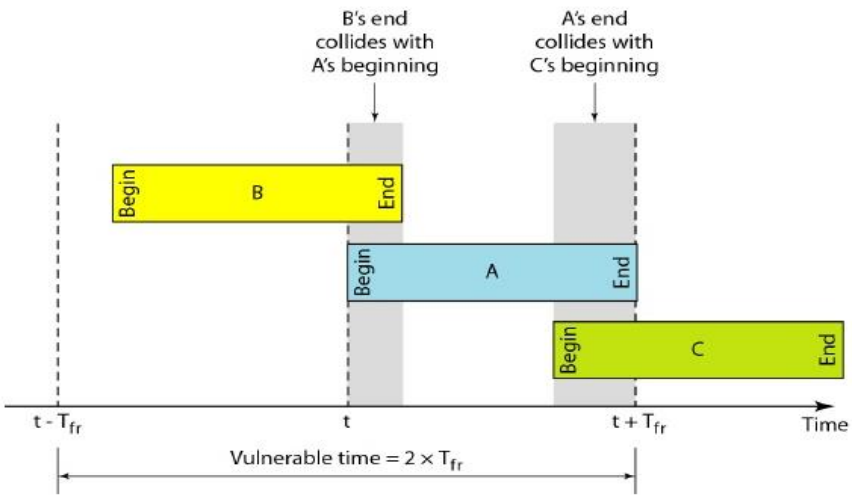
There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel. Fig.3 shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed. It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame. A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time TB.

Vulnerable time Let us find the length of time, the vulnerable time, in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking Tfr S to send. Fig.4 shows the vulnerable time for station A.

Station A sends a frame at time t. Now imagine station B has already sent a frame between t - Tfr and t. This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between t and t + Tfr. Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.

Looking at following Fig., we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.

Pure ALOHA vulnerable time = 2 x Tfr



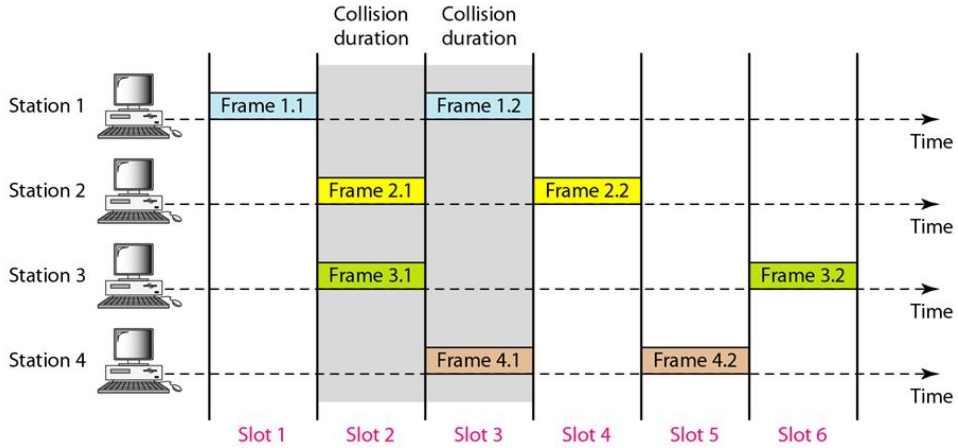
Throughput Let us call G the average number of frames generated by the system during one frame transmission time. Then it can be proved that the average number of successful transmissions for pure ALOHA is S = G x e-2G. The maximum throughput Smax is 0.184, for G = ½ . In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully. This is an expected result because the vulnerable time is 2 times the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully.

The throughput for pure ALOHA is S =G x e-2G.

The maximum throughput Smax =0.184 when G =(1/2).

- **Slotted ALOHA**

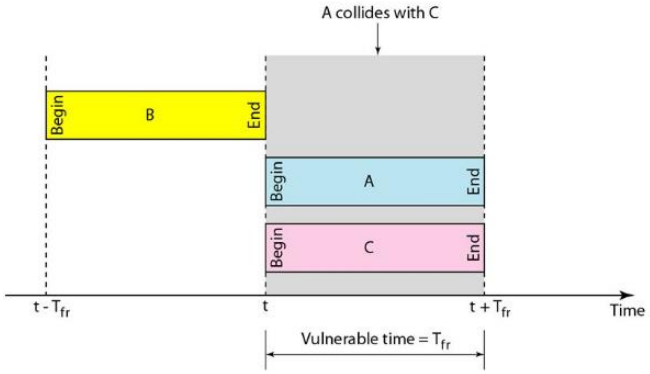
Pure ALOHA has a vulnerable time of 2 x Tfr. This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of Tfr s and force the station to send only at the beginning of the time slot. Fig. shows an example of frame collisions in slotted ALOHA.



Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to Tfr following Fig. shows

the situation. Fig. shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

Slotted ALOHA vulnerable time = Tfr



Throughput It can be proved that the average number of successful transmissions for slotted ALOHA is S = G x e-G. The maximum throughput Smax is 0.368, when G = 1. In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully. This result can be expected because the vulnerable time is equal to the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other station generates a frame during this time), the frame will reach its destination successfully.

The throughput for slotted ALOHA is S =: G x e-G.

The maximum throughput Smax == 0.368 when G =1.

Slotted ALOHA improves the performance of pure ALOHA by reducing the chances of collisions and increasing network efficiency.

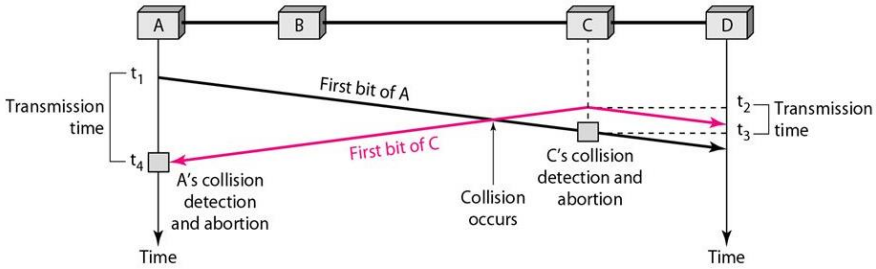
Q 7. How CSMA/CD protocol is different from CSMA/CA protocol? Explain.

(AKTU2021-22, 2022-23)

Solution:

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again. To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In the following Fig., stations A and C are involved in the collision.



At time t 1, station A has executed its persistence procedure and starts sending the bits of its frame. At time t2, station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision

occurs sometime after time t2 Station C detects a collision at time t3 when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission. Station A detects collision at time t4 when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A transmits for the duration t4 - tl; C transmits for the duration t3 - t2. Later we show that, for the protocol to work, the length of any frame divided by the bit rate in this protocol must be more than either of these durations. At time t4, the transmission of A:s frame, though incomplete, is aborted; at time t3, the transmission of B's frame, though incomplete, is aborted.

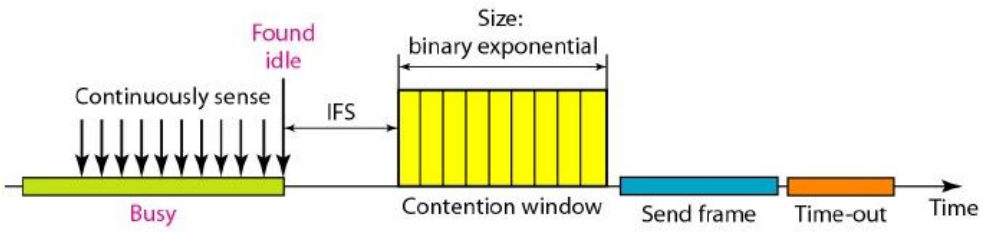
Minimum Frame Size

For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time Tfr must be at least two times the maximum propagation time Tp. To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time Tp to reach the second, and the effect of the collision takes another time Tp to reach the first. So the requirement is that the first station must still be transmitting after 2Tp .

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second station needs to add a significant amount of energy to the one created by the first station.

In a wired network, the received signal has almost the same energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender and the receiver. This means that in a collision, the detected energy almost doubles. However, in a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection. We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for this network. Collisions are avoided through the use of CSMA/CA's three strategies: the inter-frame space, the contention window, and acknowledgments, as shown in Fig.



- Interframe Space (IFS)

First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time (described next). The IFS variable can also be used to prioritize stations or frame types.

For example, a station that is assigned a shorter IFS has a higher priority.
In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

- Contention Window

The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station. One interesting point about the contention window is that the station needs to sense the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.
In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

- Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.
Procedure Note that the channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time. For each time slot of the contention window, the channel is sensed. If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.

Q 8. A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200-kbps bandwidth. Find the throughput if the system (all stations together) produces

a. 1000 frames per second
b. 500 frames per second
c. 250 frames per second **(AKTU 2018-19)**

Solution:

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is 200/200 kbps or 1 ms.
a. In this case G is 1. So S =G x e-G or S =0.368 (36.8 percent). This means that the throughput is 1000 x 0.0368 =368 frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.
b. Here G is ½ . In this case S =G x e-G or S =0.303 (30.3 percent). This means that the throughput is 500 x 0.0303 =151. Only 151 frames out of 500 will probably survive.
c. Now G is ¼ . In this case S =G x e-G or S =0.195 (19.5 percent). This means that the throughput is 250 x 0.195 = 49. Only 49 frames out of 250 will probably survive.

Q 9. State the requirements of CRC. **(AKTU 2013-14)**

Solution:

Reliability: CRC ensures accurate data delivery by identifying transmission errors.

Efficiency: The method is computationally efficient, suitable for high-speed data transmission.

Versatility: CRC is used in various communication protocols, including Ethernet, USB, and Bluetooth.

Q 10. List out discuss the disadvantages in stop and wait protocol. **(AKTU 2021-22)**

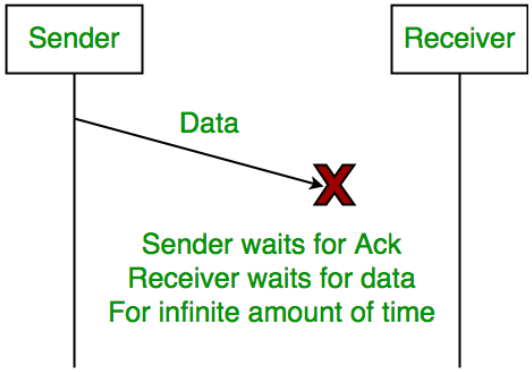
Solution:

The receiver is waiting for the data for a long time. Since the data is not received by the receiver, so it does not

send any acknowledgment. Since the sender does not receive any acknowledgment so it will not send the next packet. This problem occurs due to the lost data.

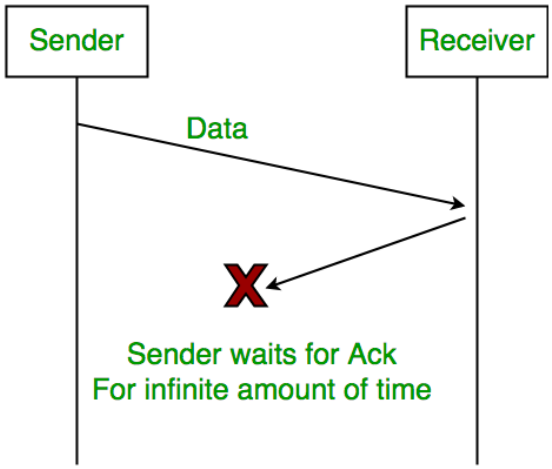
1. Lost Data

Assume the sender transmits the data packet and it is lost. The receiver has been waiting for the data for a long time. Because the data is not received by the receiver, it does not transmit an acknowledgment. The sender does not receive an acknowledgment, it will not send the next packet. This problem is caused by a loss of data.



2. Lost Acknowledgement

Assume the sender sends the data, which is also received by the receiver. The receiver sends an acknowledgment after receiving the packet. In this situation, the acknowledgment is lost in the network. The sender does not send the next data packet because it does not receive acknowledgement, under the stop and wait protocol, the next packet cannot be transmitted until the preceding packet's acknowledgment is received.



3. Delayed Acknowledgement/Data

Assume the sender sends the data, which is also received by the receiver. The receiver then transmits the acknowledgment, which is received after the sender's timeout period. After a timeout on the sender side, a long-delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

Q 11. In what situations, contention-based MAC Protocols are suitable? **(AKTU 2024-25)**

Solution: Contention-based MAC protocols (e.g., CSMA/CD, CSMA/CA) are best suited for networks with bursty traffic, low-to-moderate node density, and scenarios lacking tight time synchronization, such as Ethernet, Wi-Fi, and Ad-hoc sensor networks. They offer high scalability, low configuration overhead, and efficient performance when traffic is unpredictable.

The specific situations where contention-based protocols are ideal:

- **Bursty Traffic Environments:** They perform well when nodes transmit sporadically rather than

constantly, as they do not waste bandwidth with pre-allocated slots.

- **Decentralized/Ad-Hoc Networks:** They are suitable when no central controller is present, as they allow distributed access (e.g., in wireless ad-hoc networks).
- **Low to Moderate Network Load:** When the number of active users is low, contention mechanisms (like CSMA) have low latency.
- **Dynamic Network Topologies:** In scenarios where nodes frequently join or leave, the self-organizing nature of contention is superior to rigid, pre-scheduled systems.
- **Wired Local Area Networks (LANs):** Ethernet uses CSMA/CD to detect collisions on shared cables efficiently.
- **Wireless Sensor Networks (WSNs):** Due to their simplicity and lack of need for precise synchronization, they are often used in energy-constrained, low-duty-cycle networks.

Q 12. Illustrate the concept of slotted ALOHA with suitable diagram. **(AKTU 2024-25)**

Measurement of slotted ALOHA channel within finite number of users shows that 10% of slots are idle. Calculate

i) What is the channel load?

ii) What is the throughput?

Solution:

Given: Proportion of idle slots (P_0) = 10% = 0.1

i) Channel Load (G)

The probability of a slot being idle in slotted ALOHA follows the Poisson distribution, $P_0 = e^{-G}$.

$$0.1 = e^{-G}$$

Taking the natural logarithm on both sides:

$$\ln(0.1) = -G$$

$$-2.3025 = -G$$

$G \approx 2.30$ (packets per slot)

ii) Throughput (S)

The throughput is defined as $S = G \times e^{-G}$.

Using $G = 2.30$ and $e^{-G} = 0.1$:

$$S = 2.30 \times 0.1$$

$S \approx 0.23$ (packets per slot or 23%)

Summary: The channel load is 2.30, and the throughput is 0.23.

Q 13. A 4 MB frame is transmitted over a 1000KM link with 2 Mbps bandwidth. Propagation speed is 2×10^8 m/s. Compute total delay (latency) if there are 5 routers, each with 1µs processing and 2µs queuing delay. **(AKTU 2024-25)**

Solution:

1. Calculate Transmission Delay

Transmission delay is the time required to push all the bits of the frame onto the wire. It is calculated by dividing the frame size (L) by the bandwidth (R).

- **Frame Size (L):** $4\text{ MB} = 4 \times 1024 \times 1024\text{ bytes} \times 8\text{ bits/byte} = 33,554,432\text{ bits}$.
- **Bandwidth (R):** $2\text{ Mbps} = 2,000,000\text{ bps}$.
- $T_{trans} = \dfrac{33,554,432\text{ bits}}{2,000,000\text{ bps}} = 16.777216\text{ seconds}$.

2. Calculate Propagation Delay

Propagation delay is the time it takes for a single bit to travel from the source to the destination. It depends on the distance (d) and the propagation speed (s).

- **Distance (d):** $1000\text{ km} = 1,000,000\text{ meters}$.
- **Propagation Speed (s):** $2 \times 10^8\text{ m/s}$.
- $T_{prop} = \dfrac{1,000,000\text{ m}}{200,000,000\text{ m/s}} = 0.005\text{ seconds (5 ms)}$.

3. Calculate Router Delays

There are 5 routers, and each introduces both processing and queuing delays.

- **Processing Delay per Router:** $1\mu s = 0.000001\text{ s}$.
- **Queuing Delay per Router:** $2\mu s = 0.000002\text{ s}$.
- **Total Router Delay:** $5 \times (1\mu s + 2\mu s) = 15\mu s = 0.000015\text{ seconds}$.

4. Sum the Total Latency

The total latency is the sum of all individual delays:

$$\text{Total Delay} = T_{trans} + T_{prop} + T_{router}$$

$$\text{Total Delay} = 16.777216\text{ s} + 0.005\text{ s} + 0.000015\text{ s} = 16.782231\text{ s}$$