FRAMING

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

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

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

Although the whole message could be packed in one frame, that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

Fixed-Size Framing

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

Variable-Size Framing

Our main discussion in this chapter concerns variable-size framing, prevalent in local area networks. In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

1. The following character encoding is used in a data link protocol: A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000 Show the bit sequence transmitted (in binary) for the four-character frame: A B ESC FLAG when each of the following framing methods are used:
    a. (a) Character count.
    b. (b) Flag bytes with byte stuffing.
    c. (c) Starting and ending flag bytes, with bit stuffing.

The solution is
(a) 00000100 01000111 11100011 11100000 01111110
(b) 01111110 01000111 11100011 11100000 11100000 11100000 01111110
01111110
(c) 01111110 01000111 110100011 111000000 011111010 01111110

2. The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used: A B ESC C ESC FLAG FLAG D. What is the output after stuffing?

   The solution is
   After stuffing, we get A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D

3. A bit string, 0111101111101111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?

   Solution: The output is 011110111110011111010.

4. A channel has a bit rate of 4 kbps and a propagation delay of 20 msec. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 percent?
Solution
Efficiency will be 50% when the time to transmit the frame equals the roundtrip propagation delay. At a transmission rate of 4 bits/ms, 160 bits takes 40 ms. For frame sizes above 160 bits, stop-and-wait is reasonably efficient.

A channel has a bit rate of 4 kbps and one-way propagation delay of 20 ms. The channel uses stop and wait protocol. The transmission time of the acknowledgement frame is negligible. To get a channel efficiency of at least 50%, the minimum frame size should be

Explanation:
Bit rate = 4 kbps
One-way propagation delay = 20 ms

Efficiency =
Transmission time of packet/(Transmission time of packet + 2 * Propagation delay)

$0.5 = x/(x + 2 * 20 * 10^{-3})$
$x = 20 * 10^{-3}$
$x = 40 * 10^{-3}$
Minimum frame size / Bit rate = $40 * 10^{-3}$
Therefore, Minimum frame size = $40 * 10^{-3} * 4 * 10^{3}$ = 160 bits

5. Consider an error-free 64-kbps satellite channel used to send 512-byte data frames in one direction with very short acknowledgements coming back the other way. What is the maximum throughput for window sizes of 1, 7, 15 and 127?

A 512 byte (4096 bits) data frame has a duration of 4096/64000 seconds – that is 64 msec. Assume that the satellite is at 36000 km distance. This leads to roundtrip propagation time of 240 msec. We should also add 64 msec to this to account for transmission time. Hence with a window size of 1, 4096 bits can be sent every 240+64=304 msec. This equates to the throughput of 4096 bits/304 msec or 13.5 kbps. For a window size greater than 5, the full 64 kbps is used. 3- Two adjacent nodes (A and B) use a sliding window protocol and a 3-bit sequence

OR

Sliding window – send specified amount of frames, wait for one ack for that set of frames
512 bytes x 8 bits/B = 4096 bits per frame
4096/64000 bps = 64 msec to send one frame
Round trip delay = 540 msec
Window size 1: send 4096 bits per 540msec
4096 bits / 540 msec = 7.585 x 10^{3} bps throughput
Window size 7: 7585 x 7 = 53096 bps
Window size 9 and greater: 7585 x 9 = 68265 bps but the maximum capacity is 64 kbps so for window sizes greater than 9 the maximum throughput is 64 kbps.

6. **Consider the use of 10 K-bit size frames on a 10 Mbps satellite channel with 270 ms delay. What is the link utilization for stop-and-wait ARQ technique assuming P = 10^{-3}?**

Ans: Link utilization = (1-P) / (1+2a)
Where a = (Propagation Time) / (Transmission Time)
Propagation time = 270 msec
Transmission time = (frame length) / (data rate)
= (10 K-bit) / (10 Mbps)
= 1 msec
Hence, a = 270/1 = 270
Link utilization = 0.999/(1+2*270) ≈0.0018 =0.18%

7. **What is the channel utilization for the go-back-N protocol with window size of 7 for the problem 3?**

Ans: Channel utilization for go-back-N
= N(1 – P) / (1 + 2a)(1-P+NP)
P = probability of single frame error ≈ 10^{-3}
Channel utilization ≈ 0.01285 = 1.285%

8. **In what way selective-repeat is better than go-back-N ARQ technique?**

Ans : In selective-repeat scheme only the frame in error is retransmitted rather than transmitting all the subsequent frames. Hence it is more efficient than go-back-N ARQ technique.

9. **In what situation Stop-and-Wait protocol works efficiently?**

Ans: In case of Stop-and-Wait protocol, the transmitter after sending a frame waits for the acknowledgement from the receiver before sending the next frame. This protocol works efficiently for long frames, where propagation time is small compared to the transmission time of the frame.

10. **How the inefficiency of Stop-and-Wait protocol is overcome in sliding window protocol?**

Ans: The Stop-and-Wait protocol is inefficient when large numbers of small packets are send by the transmitter since the transmitter has to wait for the acknowledgement of each individual packet before sending the next one. This problem can be overcome by sliding window protocol. In sliding window protocol multiple frames (up to a fixed number of frames) are send before receiving an acknowledgement from the receiver.

11. For a k-bit numbering scheme, what is the range of sequence numbers used in sliding window protocol?

Ans: For k-bit numbering scheme, the total number of frames, N, in the sliding window can be given as follows (using modulo-k).
N = 2k – 1
Hence the range of sequence numbers is: 0, 1, 2, and 3 … 2k – 1

12. Using a 5 bit sequence number , what is the max. size of the send and receive window for each of the ARQ protocols?

Solution

Stop-And-Wait ARQ send window = **1**	receive window = **1**
Go-Back-N ARQ send window = $2^5 - 1$ = **31**	receive window = **1**
Selective-Repeat ARQ send window = 2^4 = **16**	receive window = **16**

Computer Network | Sliding Window protocols

Before starting with the questions a quick recap for all the protocols.

Stop and wait –

1. Sender window size (Ws) = 1
2. Receiver window size (Wr) = 1
3. Sequence Number >= 1 + 1
4. Uses independent acknowledgement
5. Discards out of order packets
6. Packet Loss → Retransmit packet after time out
7. Acknowledgement loss → Resends packet after time out
8. Efficiency = 1/(1+2a) where a = Tp/Tt

Go Back N –

1. Sender window size Ws = N
2. Receiver window size Wr = 1
3. Sequence number >= N + 1
4. Can use both cumulative or independent acknowledgement depends on acknowledge timer
5. Discards out of order packets
6. Packet Loss → Track back N size from the last packet within the window limit to the lost packet and retransmit them
7. Acknowledgement loss → If not received before timeout the entire window N size is resend
8. Efficiency = N/(1+2a) where a = Tp/Tt

Selective Repeat –

1. Sender window size Ws = N
2. Receiver window size Wr = N
3. Sequence Number >= N + N
4. Uses only independent acknowledgement
5. Can Accept out of order packets
6. Packet Loss → Resend only the lost packet after timeout
7. Acknowledgement loss → Resend if not receive before timeout
8. Efficiency = N/(1+2a) where a = Tp/Tt

Practice Questions –

- **Example-1.** In Stop and wait protocol every 4th packet is lost and we need to send total 10 packets so how many transmission it took to send all the packets ?
- **Explanation –**
- 1 2 3 4 5 6 7 8 9 10 (Initially)
- ^
- 1 2 3 4 4 5 6 7 8 9 10 (Packet no. 4 retransmitted)
- ^
- 1 2 3 4 4 5 6 7 7 8 9 10 (Packet no. 10 retransmitted)
- ^
- 1 2 3 4 4 5 6 7 7 8 9 10 10 (Result)

  So, we retransmitted packet noumber 4, 7, 10
  Total count = 13
- **Example-2.** In S&W protocol if Error probability is p and no. of packets to send is ‘n’.How many packets we have to send ?

- **Explanation –**
  Total retransmissions
  = n+ n*p+ n*p^(2) + n*p^(3) + n*p^(4) + …
  = n(1 + p + p^2 + p^3 + p^4 + …)
  = n*(1/1-p) using infinite GP formula
- **Example-3.** In GBN sender Window size = 10 and Tp = 49.5ms & Tt = 1ms. What is the Efficiency of the protocol and Throughput given Bandwidth=1000 bps ?
- **Explanation –**
  Efficiency = N/(1+2a), N = 10 (given), a = Tp/Tt = 49.5
  Efficiency = 10/(1+2*49.5) = 10/100 = 0.1 or 10%
  Throughput = Efficiency * Bandwidth
  = 0.1*1000 = 100
- **Example-4.** In GB3 if every 5th packet is lost & we need to send 10 packets so how many retransmissions are required ?
- **Explanation –**
- 1 2 3 4 5 6 7 | 8 9 10
-         ^ $ (packet no. 5 lost)
- 1 2 3 4 5 6 7 5 6 7 8 9 | 10
-               *   ^ $
- 1 2 3 4 5 6 7 5 6 7 8 9 7 8 9 10
-                          *   ^ $
- 1 2 3 4 5 6 7 5 6 7 8 9 7 8 9 10 9 10 (count starts from * till ^)
- (from ^ to $ retranmission is done)

  Note – From Last packet is window size to lost pocket we resend the entire window.
  Total no. of transmissions = 18
- **Example-5.** In SR Ws= 5 and we are sending 10 packets where every 5th packet is lost find number of retransmissions ?
- **Explanation –**
- 1 2 3 4 5 6 7 8 9 10
-         ^
- 1 2 3 4 5 5 6 7 8 9 10
-                    ^
- 1 2 3 4 5 6 7 8 9 9 10

  We see here there is no role of Window size in SR only the lost packet is resent.
  Total transmissions = 12
- **Example-6.** If there is K bits sequence no. define require sender window size and receiver window size for S&W, GBN & SR?
  Explanation –
  Given, K bits, For S&W Ws = 1 and Wr= 1
  For GBN, Ws = 2^(K)-1 and Wr = 1
  For SR, Ws = 2^(K-1) and Wr = 2^(K-1

Station A uses 32 byte packets to transmit messages to Station B using a sliding window protocol. The round trip delay between A and B is 80 milliseconds and the bottleneck bandwidth on the path between A and B is 128 kbps. What is the optimal window size that A should use?

(A) 20
(B) 40
(C) 160
(D) 320
Answer (B)

Round Trip propagation delay = 80ms
Frame size = 32*8 bits
Bandwidth = 128kbps
Transmission Time = 32*8/(128) ms = 2 ms

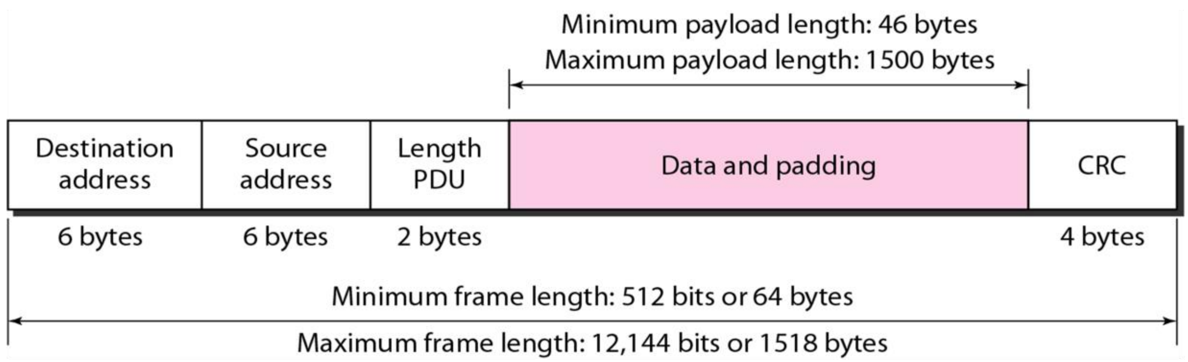
Let n be the window size.

UtiliZation = n/(1+2a) where a = Propagation time / transmission time
= n/(1+80/2)

For maximum utilization: n = 41 which is close to option (B)

In sliding window protocol total 8 packets are needed to send and if every 4th packet is lost then find the number of transmission is needed by the sender
(a) 10
(b) 14
(c) 16
(d) 20
Ans: (a)
Explanation:
1 2 3 (4) 5 6 7 8
1 2 3 4 4 5 6 (7) 8 pkt4 repeated
1 2 3 4 4 5 6 7 7 8 pkt7 repeated
Total transmission is 10

Minimum and maximum lengths



Frame length:
Minimum: 64 bytes (512 bits)
Maximum: 1518 bytes (12,144 bits)

Efficiency of CSMA/CD

Carrier sense multiple access with collision detection (CSMA/CD) – The CSMA method does not tells us what to do in case there is a collision. Carrier sense multiple access with collision detection (CSMA/CD) adds on to the CSMA algorithm to deal with collision. In CSMA/CD, the size of a frame must be large enough so that collision can be detected by sender while sending the frame. So, the frame transmission delay must be at least *two times* the maximum propagation delay.

Assume some station transmitted data packet and successfully get to destination but it just the *Best Case*, so we have to take *Worst Case* scenario in which there will be contention slots. Contention slots are those slot which are not able to transmit their journey due to collision. Suppose A station transmitted data but collide and worst case time wasted is **2Tp** and then some station B found out way to transmit the data so it took

Tp (propagation delay) + Tt(transmission time)

Efficiency = Tt / (C*2*Tp + Tt + Tp)

Tt → transmission time

Tp → propagation time

C → number of collision

Problem:

Consider the Pure ALOHA, Slotted ALOHA, and Non-persistent CSMA. Which one will you use at high load? Why?

Solution:

I will use Non-persistent CSMA. At high load there would be more transmission attempts, Non-persistent CSMA senses the channel before sending and causes fewer collisions than Pure ALOHA and Slotted ALOHA. So Non-persistent CSMA has best performance at high load.

Problem:

Ten thousand airline stations are competing for the use of a single slotted ALOHA channel. The average station makes 18 requests/hour. A slot is 125 micro-sec. What is the approximate total channel load?

Solution:

Average requests for 10000 stations = 10^4 x 18 / (60 x 60) = 50 requests/sec

Average slots number = 1 / (125 x 10^-6) = 8000 slots/sec.

Total channel load = average requests / average slots number

= 50 / 8000 = 0.0625

Hence, the total channel load is 0.0625 request/slot.

Problem:

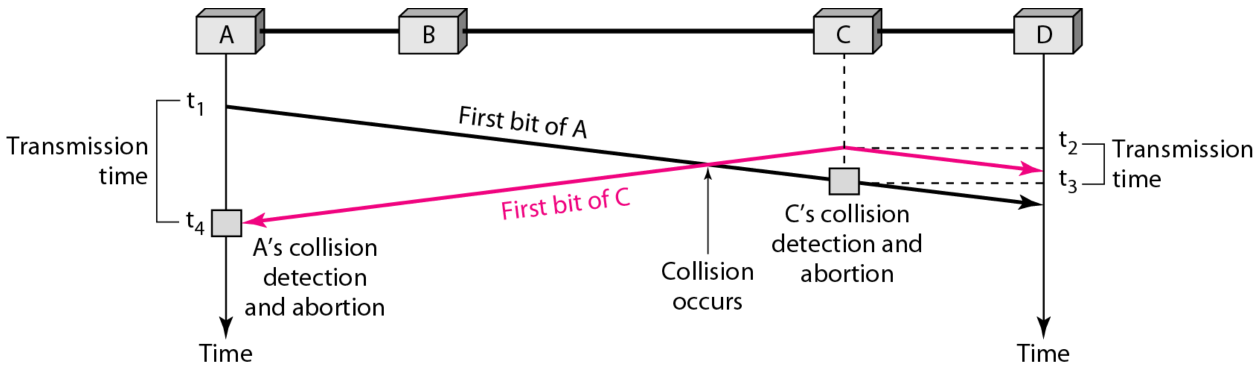
In Following Figure , the data rate is 10 Mbps, the distance between station A and C is 2000 m, and the propagation speed is 2 x 10^8 *m/s*. Station A starts sending a long frame at time *t1* =0; station C starts sending a long frame at time *t2* = 3 µs. The size of the frame is long enough to guarantee the detection of collision by both stations. Find:

a. The time when station C hears the collision *(t3)'*

b. The time when station A hears the collision *(t4)'*

c. The number of bits station A has sent before detecting the collision.

d. The number of bits station C has sent before detecting the collision.



Solution:

We have t1 = 0 and t2 = 3 µs

a. t3 − t1= (2000 m) / (2 × 108 m/s) =10 µs ⟶ t3 = 10 µs + t1 = **10 µs**

b. t4 − t2 = (2000 m) / (2 × 108 m/s) =10 µs ⟶ t4 = 10 µs + t2 = **13 µs**

c. Tfr(A) = t4 − t1 = 13 − 0 = 13 µs ⟶ BitsA = 10 Mbps × 13 µs = **130 bits**

d. Tfr(C) = t3 − t2 = 10 − 3 = 07µs ⟶ BitsC = 10 Mbps × 07 µs = **70 bits**

If the bandwidth of the line is 1.5 Mbps, RTT is 45 msec and packet sizes is 1 KB, then find the link utilization in stop and wait with proper explanation.

Utilization = transmission time / (transmission time + RTT)

--> Transmission time = Length of packet / Bandwidth

= 1KB / 1.5 Mbps = 1*10^(3)*8 / 1.5*10^(6) = 5.33 msec.

--> Utilization = 5.33 / (5.33 + 45) = 0.1059 = 10.59% . [Rtt = 45