Solutions to Chapter 5

1. Explain the difference between connectionless unacknowledged service and connectionless acknowledged service. How do the protocols that provide these services differ?

Solution:

In an acknowledged connectionless network, reliable delivery can be achieved through the use of ACK and NAK transmissions. Such protocols are suited for communication over networks in which higher layers are sensitive to loss and the underlying network is inherently unreliable with a significant probability of loss or error.

Unacknowledged networks provide simpler and faster communication for networks that are inherently reliable or provide service to higher layers that can tolerate information loss.

2. Explain the difference between connection-oriented acknowledged service and connectionless acknowledged service. How do the protocols that provide these services differ?

Solution:

The use of acknowledgments can provide reliable transfer over networks that are prone to error and loss. In connection oriented networks, every packet in a data flow travels on the same path through the network and the proper ordering of packets is guaranteed. In such networks, if a packet arrives out of order, the receiver immediately knows that a packet has been lost. In a connectionless network, the service needs a mechanism for dealing with unordered delivery of information. This is especially important for real-time or delay-sensitive traffic, which may require immediate retransmission and may not be able to use buffering to correct unordered packet arrivals.

3. Suppose that the two end-systems α and β in Figure 5.3 communicate over a connection-oriented packet network. Suppose that station α sends a 10-kilobyte message to station β and that all packets are restricted to be 1000 bytes (neglect headers); assume that each packet can be accommodated in a data link frame. For each of the links, let p be the probability that a frame incurs errors during transmission.

Solutions follow questions:

Suppose that the data link control just transfers frames and does not implement error control. Find the probability that the *message* arrives without errors at station β.

Let p be the probability that a frame incurs errors during transmission. We know the following:

Message length = 10,000 bytes
Maximum packet size = 1000 bytes
Number of packets for transmission = 10

The probability of a packet arriving error free at end system $P_{packet} = (1 - p)^3$. The probability that all packets arrive error free at end system β is $P_{error} = [(1 - p)^3]^{10} = (1 - p)^{30} \approx e^{-30p}$.

a. Suppose that error recovery is carried out end to end and that if there are any errors, the entire message is retransmitted. How many times does the message have to be retransmitted on average?

The average number of required transmissions = $1/P_{error} = e^{30p}$.

b. Suppose that the error recovery is carried out end to end on a packet-by-packet basis. What is the total number of packet transmissions required to transfer the entire message?

The average number of transmissions per packet = $1/P_{packet} = e^{3p}$. The total number of packet transmissions is then $10/P_{packet} = 10e^{3p}$.

As an example suppose p = .01, then the message retransmission approach requires 1.35 message transmissions. The packet transmission approach requires 1.03 message transmissions. Clearly packet-by-packet retransmission is better.

4. Suppose that two peer-to-peer processes provide a service that involves the transfer of discrete messages. Suppose that the peer processes are allowed to exchange PDUs that have a maximum size of *M* bytes including *H* bytes of header. Suppose that a PDU is not allowed to carry information from more than one message.

Solutions follow questions:

a. Develop an approach that allows the peer processes to exchange messages of arbitrary size.

To exchange messages of arbitrary size, large messages must be segmented into parts of M-H bytes each in length to be transmitted in multiple PDUs. Small messages must be placed in a single PDU.

b. What essential control information needs to be exchanged between the peer processes?

The peer processes need to communicate information that allows for the reassembly of messages at the receiver. For example, the first PDU may contain the message length. The last PDU may contain and end-of-message marker. Sequence numbers may also be useful to detect loss in connection oriented networks and to help in reconstruction of the messages in connectionless networks. Lastly, since variable size PDUs are permitted, the size of the PDU must be transmitted in the PDU header.

c. Now suppose that the message transfer service provided by the peer processes is shared by several message source-destination pairs. Is additional control information required, and if so, where should it be placed?

In this case, in addition to all of the header information mentioned in b), each PDU must be labeled with a stream ID, so that the receiver can treat each stream independently when reassembling messages.

5. Suppose that two peer-to-peer processes provide a service that involves the transfer of a stream of bytes. Suppose that the peer processes are allowed to exchange PDUs that have a maximum size of *M* bytes, including *H* bytes of header.

Solutions follow questions:

a. Develop an approach that allows the peer processes to transfer the stream of bytes in a manner that uses the transmission line efficiently. What control information is required in each PDU?

The streams should be segmented into $M - H$ size blocks and transmitted in the PDUs. Since there is no inherent message length known a priori, an end-of-stream code must be transmitted in the last PDU. If the stream bytes arrive at a sufficient rate, the PDU size will be constant and need not be transmitted in the header. The packets may still be numbered for reordering in a connectionless network (as in problem 4).

The number of PDUs in the stream can be theoretically infinite and only a finite number of bits can be used to represent sequence numbers. Thus, the size of the sequence number field in the header should be chosen small enough to incur as little overhead as possible and large enough so that two simultaneously arriving (but different) PDUs with the same sequence number is unlikely.

b. Suppose that the bytes in the stream arrive sporadically. What is a reasonable way to balance efficiency and delay at the transmitter? What control information is required in each PDU?

The bytes at the transmitter should be buffered until $M - H$ are available for transfer. Depending on how sporadic the arrivals are and the delay sensitivity of the application, this buffer delay may need to be limited by sending partially full PDUs. Thus, each PDU header must include a field to hold the size of the PDU

c. Suppose that the bytes arrive at a constant rate and that no byte is to be delayed by more that T seconds.

Does this have an impact on the efficiency?

Yes. Because the header length is constant, larger PDUs provide more efficient data transfer. If T is very small, partially filled packets must always be sent. If T is large, larger PDUs can be transmitted so more efficient transmission can be achieved. Because the bytes arrive at a constant rate, the PDUs will all be the same length. Thus, there is no need for a PDU length field in the header.

d. Suppose that the bytes arrive at a variable rate and that no byte is to be delayed by more than T seconds. Is there a way to meet this requirement?

A timer is required that counts down from T seconds. When the first byte arrives the timer starts. When a PDU is full or when the timer expires (whichever occurs first), the PDU is transmitted and the timer is restarted upon the arrival of the next byte.

6. Suppose that two peer-to-peer processes provide a service that involves the transfer of a stream of bytes. Develop an approach that allows the stream transfer service to be shared by several pairs of users in the following cases:

Solutions follow questions:

a. The bytes from each user pair arrive at the same constant rate.

All incoming bytes from the n streams should be buffered separately into n queues. Each queue should be serviced in round-robin fashion.

b. The bytes from the user pairs arrive sporadically and at different rates.

All incoming bytes from the n streams should be buffered separately into n queues. Assuming that the arrival rate of each stream is constant, each queue should be serviced at rate proportional to their arrival rate in sequential order. One implementation of this is weighted round-robin service, where the weights would be proportional to the arrival rates.

If the arrival rates are variable, the weight can be dynamically proportional to the inverse of the queue lengths. In both cases, if a queue is empty, it loses its “turn” in the round-robin service.

7. Consider the transfer of a single real-time telephone voice signal across a packet network. Suppose that each voice sample should not be delayed by more than 20 ms.

Solutions follow questions:

a. Discuss which of the following adaptation functions are relevant to meeting the requirements of this transfer: handling of arbitrary message size; reliability and sequencing; pacing and flow control; timing; addressing; and privacy, integrity and authentication.

Message size is important because in real-time signals of voice it is necessary to transfer fixed packet size of that holds no more than 20 ms of speech signal. The handling of arbitrary message size is not as important as long as the desired packet size for voice can be handled.

Sequencing is important because each packet needs to arrive in the same sequence that it was generated. *Reliability* is moderately important since voice transmission can tolerate a certain level of loss and error.

Pacing and flow control are not as important because the synchronous nature of the voice signal implies that the end systems will be matched in speed.

Timing, for real-time voice transfer is important because this adaptation function helps to control the jitter in the delivered signal.

Addressing is only during the connection setup phase if we assume some for of virtual circuit packet switching method.

Privacy, integrity, and authentication have traditionally not been as important as the other issues discussed above.

b. Compare a hop-by-hop approach to an end-to-end approach to meeting the requirements of the voice signal.

If the underlying network is reliable then the end-to-end approach is better because the probability of error is very low so processing at the edge suffices to provide acceptable performance.

If the underlying network is unreliable then the hop-by-hop approach may be required. For example if the probability of error is very high, as in a wireless channel, then error recovery at each hop may be necessary to make effective communication possible.

8. Suppose that a packet network is used to transfer *all* the voice signals that arrive at the base station of a cellular telephone network to a telephone office. Suppose that each voice sample should not be delayed by more than 20 ms.

Solutions follow questions:

a. Discuss which of the following adaptation functions are relevant to meeting the requirements of this transfer: handling of arbitrary message size; reliability and sequencing; pacing and flow control; timing; addressing; and privacy, integrity and authentication.

The following table summarizes parts (a) and (b):

Adaptation function	**Relevant in Upstream Direction (part a)**	**Relevant in Downstream Direction (part b)**
Handling of arbitrary message size	No	No
Reliability and sequencing	Yes	Yes
Pacing and flow control	No	No
Timing	Yes	Yes
Addressing	Yes	Yes
Privacy, integrity and authentication	Yes	No

Because traffic in a cellular network consists of constant rate streams of information, the message size need not be arbitrary, and the timing and sequence of packets is essential for service. Addressing is necessary to identify the two end callers, and authentication may only be necessary in the upstream direction, although if it is done at the base station, it need not be repeated. Pacing and flow control are not necessary because cellular networks service constant bit rate traffic and only admit calls that can be accommodated at that constant bit rate.

b. Are the requirements the same in the opposite direction from the telephone office to the base station?

See answer to part (a).

c. Do the answers to parts (a) and (b) change if the signals arriving at the base station include e-mail and other short messages?

If the signals include email and other short messages in addition to voice, further adaptation functions are required. The handling of arbitrary message sizes is needed, since email messages are variable in length. Although timing is still necessary for voice traffic, the message traffic has no strict timing requirements. Pacing and flow control may now be needed depending on the nature of the messages. Text email is low in bandwidth and likely would not require flow control, but very large messages, particularly those transmitted in the downstream direction to the cellular user would require flow control.

9. Suppose that streaming video information is transferred from a server to a user over a packet network.

Solutions follow questions:

a. Discuss which of the following adaptation functions are relevant to meeting the requirements of this transfer: handling of arbitrary message size; reliability and sequencing; pacing and flow control; timing; addressing; and privacy, integrity and authentication.

The following table summarizes the results of part (a).

Adaptation function	Required	Discussion
Handling of arbitrary message size	No	Stream has no inherent required message size
Reliability and sequencing	Yes	This may be relaxed in connection-oriented network
Pacing and flow control	Maybe	See note below
Timing	Yes	Video has strict timing requirements
Addressing	Maybe	Required for point-to-point or point-to multipoint transfer, but not in broadcast system
Privacy, integrity and authentication	Maybe	Required for point-to-point or point-to multipoint transfer, but *may* not be required in broadcast system

Whether explicit pacing and flow control is necessary depends on the nature of the receiver. If the receiver is dedicated for receiving video, it should able to handle the incoming signal. If other applications may be received at the same time, flow control might be needed.

If the signal is constant-rate, uncompressed video, flow control is not possible. For a compressed signal, flow control may be implemented by using MPEG, which transmits video in layers that can be selectively discarded to reduce the bandwidth to the receiver.

b. Suppose that the user has basic VCR features through control messages that are transferred from the user to the server. What are the adaptation requirements for the control messages?

The following table summarizes the results of part (b).

Adaptation function	Required	Discussion
Handling of arbitrary message size	No	Can define messages to be fixed length
Reliability and sequencing	Yes	Necessary to allow user to perform many operations in a short time period
Pacing and flow control	No	Control messages would not likely produce high bandwidth
Timing	Yes	Interactive control has timing requirements
Addressing	Yes	The control traffic must be addressed to the video server
Privacy, integrity and authentication	Yes	Authentication is important that only the users have control over the content they receive. Privacy for control information may be provided, but would likely not be a strict requirement.

10. Discuss the merits of the end-to-end vs. hop-by-hop approaches to providing a constant transfer delay for information transferred from a sending end system to a receiving end system.

Solution:

Jitter in networks has two primary sources – variation in queuing delay and variation in propagation delay. The former is caused because traffic intensity at nodes varies with time, so the buffering that is required at each node is equally variable. The latter exists primarily in connectionless networks and results because each packet in a flow can traverse a different path through the network to its destination.

Hop-by-hop approaches can be used to deal with queuing delay variation, and end-to-end approaches are required to deal with path-length variation. By setting up a path prior to transmission, a constant propagation delay can be ensured. To deal with queuing delay variation, the scheduler at a node can give delay sensitive traffic priority, ensuring that it's delay is kept below some maximum amount. This can be based on header information in the packet. If the path for a flow has been previously specified, the number of nodes in the path is known, so an overall queuing delay maximum is, thus, insured.

11. Consider the Stop-and-Wait protocol as described in the chapter. Suppose that the protocol is modified so that each time a frame is found in error at either the sender or receiver, the last transmitted frame is immediately resent.

Solutions follow questions:

a. Show that the protocol still operates correctly.

The protocol will operate correctly because there is only one variation from the protocol in the chapter, namely, the sender will retransmit before the time-out.

b. Does the state transition diagram need to be modified to describe the new operation?

The state transition diagram remains the same.

c. What is the main effect of introducing the immediate-retransmission feature?

The main effect is that the expected time for transmission is reduced because when the error is detected a NAK is send and the sender can stop the transmission and initiate the retransmission of the frame. If the error is in the ACK then the sender will not have to wait for the time out. Always when there is an error in the ACK or NAK the last frame sent has to be retransmitted because the sender does not know if the frame was received with or without errors.

12. In Stop-and-Wait ARQ why should the receiver always send an acknowledgment message each time it receives a frame with the wrong sequence number?

Solution:

The sender cannot send the next frame until it has received the ACK for the last frame so, if the receiver gets a frame with the wrong sequence it has to be a retransmission of the previous frame received. This means that the ACK was lost so the receiver has to ACK again to indicate the sender that it has received the frame.

13. Discuss the factors that should be considered in deciding whether an ARQ protocol should act on a frame in which errors are detected.

Solution:

If a frame is in error, then all of the information contained in it is unreliable. Hence any action taken as a result of receiving an erroneous frame should not use the information inside the frame. A viable option when an erroneous frame is received is to do nothing, and instead to rely on a timeout mechanism to initiate retransmission. However error recovery will be faster if we use a NACK message to prompt the sender to retransmit. The inherent tradeoff is between the bandwidth consumed by the NACK message and the faster recovery.

14. Suppose that a network layer entity requests its data link layer to set up a connection to another network layer entity. In order to setup a connection in a data link, the initiating data link entity sends a SETUP frame, such as SABM in Figure 5.37. Upon receiving such a frame, the receiving data link entity sends an acknowledgment frame confirming receipt of the SETUP frame. Upon receiving this acknowledgment the initiating entity can inform its network layer that the connection has been setup and is ready to transfer information. This situation provides an example of how *unnumbered acknowledgments* can arise for confirmed services.

Solutions follow questions:

a. Reexamine Figure 5.9 and Figure 5.10 with respect to error events that can take place and explain how these events are handled so that connection setup can take place reliably.

Fundamentally, the problem involves getting the receiver state to change from an idle state to a connected state. This is the same as getting the state to go from state (0,0) to state (1,1) in Figure 5.11. Therefore transmissions of the SETUP message must be accompanied by the start of an associated timer to trigger retransmissions. Also, the receiver must acknowledge every SETUP frame that it receives. The sender should ignore redundant SETUP acknowledgments and the receiver should ignore redundant SETUP frames. In order to allow multiple connections, each flow's SETUP and acknowledgment frames should be indexed.

b. To terminate the connection, either data link layer can send a DISC frame that is then acknowledged by an unnumbered acknowledgment. Discuss the effect of the above error events and how they can be dealt with.

This problem is very similar to problem a. except that in the SETUP problem the actual transfer of frames cannot begin until a SETUP acknowledgment is received. In the disconnect case, the sender can stop transmitting after a certain number of retransmissions of the DISC message.

c. Suppose that an initiating station sends a SETUP frame twice but that the corresponding ACK times are delayed a long time. Just as the ACK frames from the original transmissions are about to arrive, the initiating station gives up and sends a DISC frame followed by another SETUP frame. What goes wrong if the SETUP frame is lost?

If the DISC acknowledgment is lost, the original DISC message will remain unacknowledged. By conventional protocol rules, the initiating station will retransmit the DISC message. The receiver will receive this message last (likely after acknowledging the last SETUP message and the connection will be terminated.

To solve this, the initiating station should not send out SETUP messages with DISC messages outstanding. Alternatively, the ARQ could be altered to let the initiating station ignore outstanding DISC messages if another SETUP has been sent.

15. A 1 Mbyte file is to be transmitted over a 1 Mbps communication line that has a bit error rate of $p = 10^{-6}$.

Solutions follow questions:

The file length $n = 8 \times 10^6$ bits, the transmission rate $R = 1$ Mbps, and $p = 10^{-6}$.

a. What is the probability that the entire file is transmitted without errors? Note for n large and p very small, $(1 - p)^n \approx e^{-np}$.

P[no error in the entire file] = $(1 - p)^n \approx e^{-np}$, for $n >> 1$, $p << 1$

$= e^{-8} = 3.35 \times 10^{-4}$

We conclude that it is extremely unlikely that the file will arrive error free.

b. The file is broken up into N equal-sized blocks that are transmitted separately. What is the probability that all the blocks arrive correctly without error? Does dividing the file into blocks help?

A subblock of length n/N is received without error with probability:

P[no error in subblock] = $(1 - p)^{n/N}$

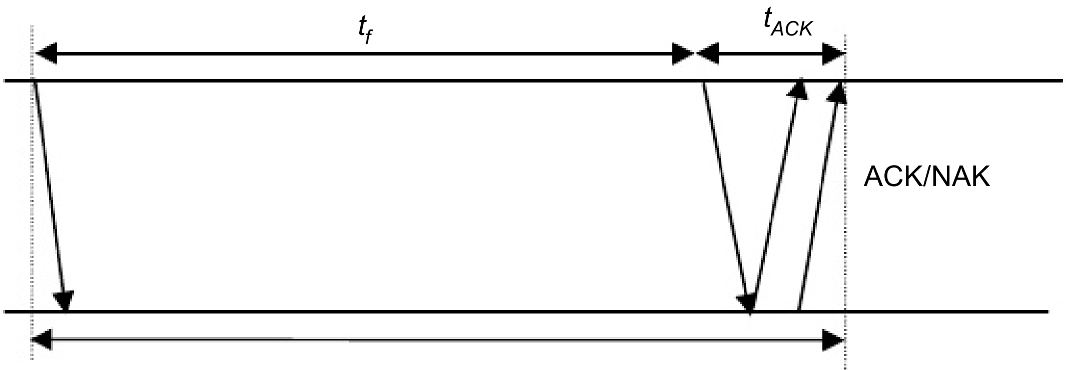
A block has no errors if all subblocks have no errors, so

P[no error in block] = P[no errors in subblock]^N $=((1 - p)^{n/N})^N = (1 - p)^n$

So simply dividing the blocks does not help.

c. Suppose the propagation delay is negligible, explain how Stop-and-Wait ARQ can help deliver the file in error-free form. On the average how long does it take to deliver the file if the ARQ transmits the entire file each time?

Refer to the following figure for the discussion.



We assume the following:

- t_0 = basic time to send a frame and receive the ACK/NAK
- t_{total} = total transmission time until success
- n_f = number of bits/frame
- n_a = number of bits per ACK
- n_t = number of transmissions
- P_f = probability of frame transmission error

$t_0 = t_f + t_{ACK} = n_f/R + n_a/R \qquad (t_{\text{prop}} \approx 0).$

$P[n_t = i] = P$[one success after $i - 1$ failure] $= (1 - P_f)\, P_f^{\,i-1}$

$t_{\text{total}} \mid i$ transmissions $= i.t_0$

$$E[t_{\text{total}}] = \sum_{i=1}^{\infty} i t_0 P[n_t = i] = t_0 (1 - P_f) \sum_{i=1}^{\infty} i P_f^{\,i-1} = t_0 (1 - P_f)/(1 - P_f)^2 = t_0/(1 - P_f)$$

Here, $n_f = n >> n_a$ thus $t_0 \approx t_f = n/R$; and $P^f = 1 - P$[no error] $= 1 - e^{-np}$

$$E[\text{total}] = n/R\,(1 - P_f) = n/[Re^{-np}] = 8 / (3.35 \times 10^{-4}) = 23{,}847 \text{ seconds} = 6.62 \text{ hours!}$$

The file gets through, but only after many retransmissions.

d. Now consider breaking up the file into N blocks. (Neglect the overhead for the header and CRC bits.) On the average how long does it take to deliver the file if the ARQ transmits the blocks one at a time? Evaluate your answer for N = 80, 800, and 8000.

For 1 block $P_f = 1 - P_b = 1 - (1 - p)\ n/N$ and $n_f = n/N$

if $t_{\text{prop}} \approx 0$ and $n_a << n/N$: $t_0^{\ b} = n_f/R = n/NR$

$$T_b = E[t_{\text{total}}^{\ b}] = t_0^{\ b} / (1 - P_f) = n(1 - p)^{-n/N} / NR \quad \text{average time to transmit one block}$$

$$T = E[t_{\text{total}}] = N\,T_b = n(1 - p)^{-n/N} / R = 8\,(1 - p)^{-n/N} = 8\,e^{np/N} \quad \text{if } n/N >> 1,\ p << 1$$

- $N = 80 \Rightarrow T \approx 8\,e^{0.1} = 8.84$ sec
- $N = 800 \Rightarrow T \approx 8\,e^{0.01} = 8.08$ sec
- $N = 8000 \Rightarrow T \approx 8\,e^{0.001} = 8.008$ sec

Each subblock has a higher probability of arriving without errors, and so requires fewer retransmissions to deliver error free. The overall delay is reduced dramatically.

e. Explain qualitatively what happens to the answer in part (d) when the overhead is taken into account.

As N increases, the effect of overhead becomes more significant because the headers constitute a bigger fraction of each subblock.

16. Consider the state transition diagram for Stop-and-Wait ARQ in Figure 5.11. Let P_f be the probability of frame error in going from station A to station B and let P_a be probability of ACK error in going from B to A. Suppose that information frames are two units long, ACK frames are one unit long, and propagation and processing delays are negligible. What is the average time that it takes to go from state (0,0) to state (0,1)? What is the average time that it then takes to go from state (0,1) to state (1,1)? What is the throughput of the system in information frames/second?

Solution:

We know that P_f is the probability of frame error and P_a is the probability of ACK error. We assume that:

- X is the random variable that represents the number of trials before a successful transmission of a frame.
- Y is the random variable that represents the number of trials before a successful transmission of an ACK.
- X and Y follow a geometric random-variable distribution.

  $T1$ = Average time to go from (0,0) to (0,1) = 2 [$E(X)$ + 1]

$$T1 = 2\left[\frac{P_f}{1-P_f} + 1\right] = 2\left[\frac{P_f + 1 - P_f}{1-P_f}\right] = \frac{2}{1-P_f}$$

$T2$ = Average time to go from (0,1) to (1,1) = 1 [$E(Y)$ + 1]

$$T2 = \left[\frac{P_a}{1-P_a} + 1\right] = \left[\frac{P_a + 1 - P_a}{1-P_a}\right] = \frac{1}{1-P_a}$$

Throughput = Frame Time / Expected Total Transmission Time = 2 / ($T1$ + $T2$)

P_f	P_a	$T1$	$T2$	Throughput
0.2	0.1	2.5000	1.1111	0.5538
0.02	0.01	2.0408	1.0101	0.6555
0.002	0.001	2.0040	1.0010	0.6656
0	0	2.0000	1.0000	0.6667

17. Write a program for the transmitter and the receiver implementing Stop-and-Wait ARQ over a data link that can introduce errors in transmission. Assume station A has an unlimited supply of frames to send to station B. Only ACK frames are sent from station B to station A. Hint: Identify each event that can take place at the transmitter and receiver and specify the required action.

Solution:

In the table below, each possible event, the current state of the transmitter and its corresponding action are described.

Event		**State**			**Action**
	Packet error	*Buffer Empty*	*Correct Sequence Number*	*ACK outstanding*	
Data arrives	NA	N	NA	DC	Add the data to the buffer Start timeout
Data arrives	NA	Y	NA	N	Send the data, set the timeout timer
Data arrives	NA	Y	NA	Y	Buffer the data
ACK arrives	N	Y	Y	-	Reset the timeout timer
ACK arrives	N	N	Y	-	Send the next frame Set the timeout timer
ACK arrives	N	DC	N	-	Do nothing
ACK arrives	Y	DC	NA	-	Do nothing

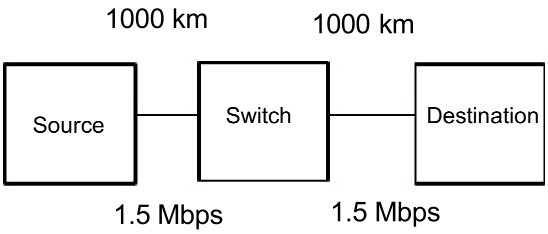
Note: DC – don’t care, NA – not applicable

In the table below, each possible event, the current state of the receiver and its corresponding action are described.

Event			**Action**
Data arrives	R_{next} is correct	Error in packet	
Y	Y	Y	Do nothing
Y	Y	N	Send next ACK
Y	N	N	Send Previous ACK
N	NA	NA	Do Nothing (idle state)

Note: NA – not applicable

18. A 64-kilobyte message is to be transmitted from the source to the destination. The network limits packets to a maximum size of two kilobytes, and each packet has a 32-byte header. The transmission lines in the network have a bit error rate of 10–6, and Stop-and-Wait ARQ is used in each transmission line. How long does it take on the average to get the message from the source to the destination? Assume that the signal propagates at a speed of 2 x 105 km/second.



Solution:

Message Size	65536 bytes
Max Packet Size	2048 bytes
Packet Header	32 bytes
Available for info	2016 bytes
# of packets needed	32.51 packets
Total	33 packets

bit error rate	1E-06		
bits/packet	16384		
Probability of error in packet	0.016251		1 – (1– bit_error_rate) ^ (bits/packet)
Propagation speed	2E+05	Km/s	
Distance	1000	Km	
Bandwidth	1.5	Mb/s	

We assume that the ACK time and processing time are negligible.

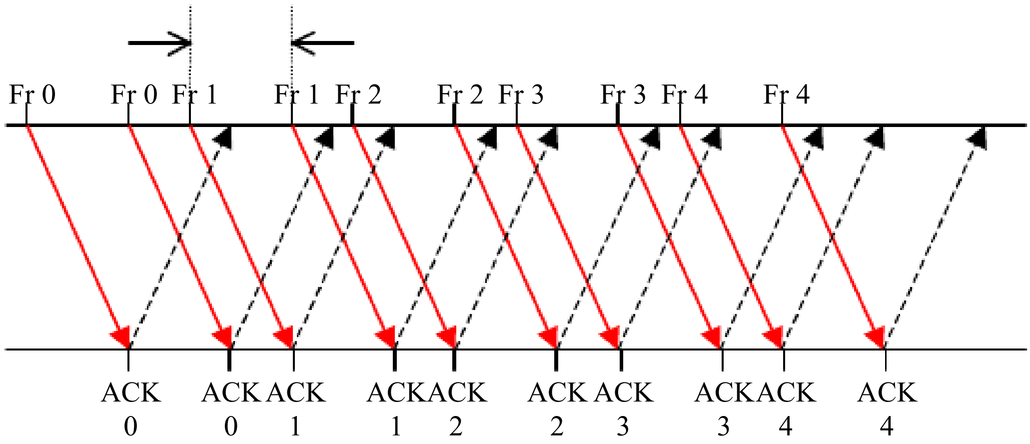
T_{prop} = distance / propagation speed = 0.0050 s
T_f = packet size / bandwidth = 0.0109 s
T_0 = $T_{prop} + T_f$ = 0.0159 s
P_f = probability of error in packet = 0.016251

$E[T_{total}] = T_0 / (1 - P_f) = 0.0162$

Since we have to send every packet over two links and we have 33 packets, then the average time is $E[T_{total}]$ * 2 * 33 = 1.068255711.

19. Suppose that a Stop-and-Wait ARQ system has a time-out value that is less than the time required to receive an acknowledgment. Sketch the sequence of frame exchanges that transpire between two stations when station A sends five frames to station B and no errors occur during transmission.

Solution:



20. The Trivial File Transfer Protocol (RFC 1350) is an application layer protocol that uses the Stop-and-Wait protocol. To transfer a file from a server to a client, the server breaks the file into blocks of 512 bytes and sends these blocks to the client using Stop-and-Wait ARQ. Find the efficiency in transmitting a 1 MB file over a 10 Mbps Ethernet LAN that has a diameter of 300 meters. Assume the transmissions are error free and that each packet has 60 bytes of header attached.

Solution:

The propagation delay in an Ethernet LAN is negligible compared to the total transmission time of a packet from start to finish. Ignoring processing time and using the terminology in the chapter, we have:

$$t_o = t_f + t_{ack} = \frac{8(512+60)}{10\times10^6} + \frac{64}{10\times10^6} = 4.64\times10^{-4}$$

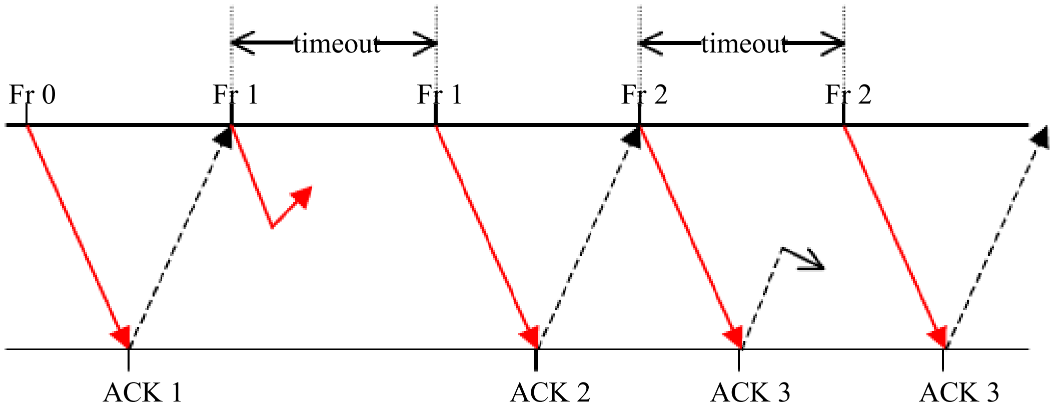
$$\eta_o = \frac{R^0_{eff}}{R} = \frac{\dfrac{n_f - n_o}{t_o}}{R} = \frac{\dfrac{8\times512}{4.64\times10^{-4}}}{10\times10^6} = 0.8828 = 88.3\%$$

One more source of overhead occurs because the last packet is not full. However, this additional overhead accounts for a very small fraction of the total overhead and does not affect the above result.

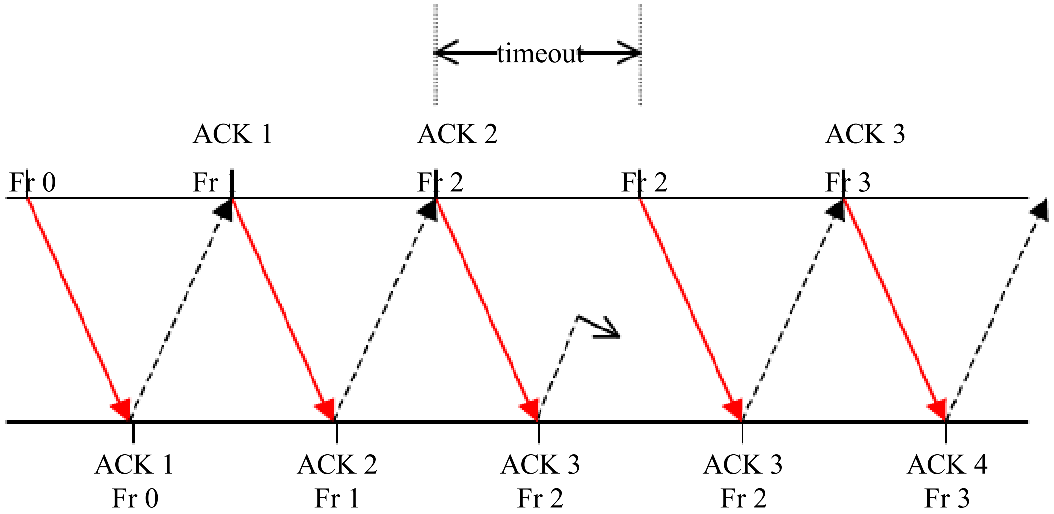
21. Compare the operation of Stop-and-Wait ARQ with bidirectional Go-Back-N ARQ with a window size of 1. Sketch out a sequence of frame exchanges using each of these protocols and observe how the protocols react to the loss of an information frame and to the loss of an acknowledgment frame.

Solution:

Stop-and-Wait:



Go-Back-N:



The performance of both protocols is identical in terms of recovery time. Each one takes on full timeout, which is equal to one full propagation delay.

22. Consider the various combinations of communication channels with bit rates of 1 Mbps, 10 Mbps, 100 Mbps, and 1 Gbps over links that have roundtrip times of 10 msec, 1 msec, and 100 msec.

Solutions follow questions:

a. Find the delay-bandwidth product for each of the 12 combinations of speed and distance.

Delay-bandwidth (Mb)

Bit Rate Mbps	Round Trip Time (msec)		
	100	10	1
1	0.1	0.01	0.001
10	1.0	0.10	0.010
100	10.0	1.00	0.100
1000	100.0	10.00	1.000

b. Suppose 32-bit sequence numbers are used to transmit blocks of 1000 bytes over the above channels. How long does it take for the sequence numbers to wrap around, that is, to go from 0 up to 2^m?

Block 1000 bytes
Sequence 32 bits 4294967296

assuming no errors

Time for the sequence number to wrap around (sec)
Prop Delay + Frame Time * 2^{32}

Bit Rate Mbps	Round Trip Time (msec)		
	100	10	1
1	34359738.47	34359738.38	34359738.37
10	3435973.94	3435973.85	3435973.84
100	343597.48	343597.39	343597.38
1000	34359.84	34359.75	34359.74

c. Now suppose the 32-bit sequence numbers are used to count individual transmitted bytes. How long does it take for the sequence numbers to wrap around?

Same analysis as (b) changing the block from 1000 bytes to 1 byte

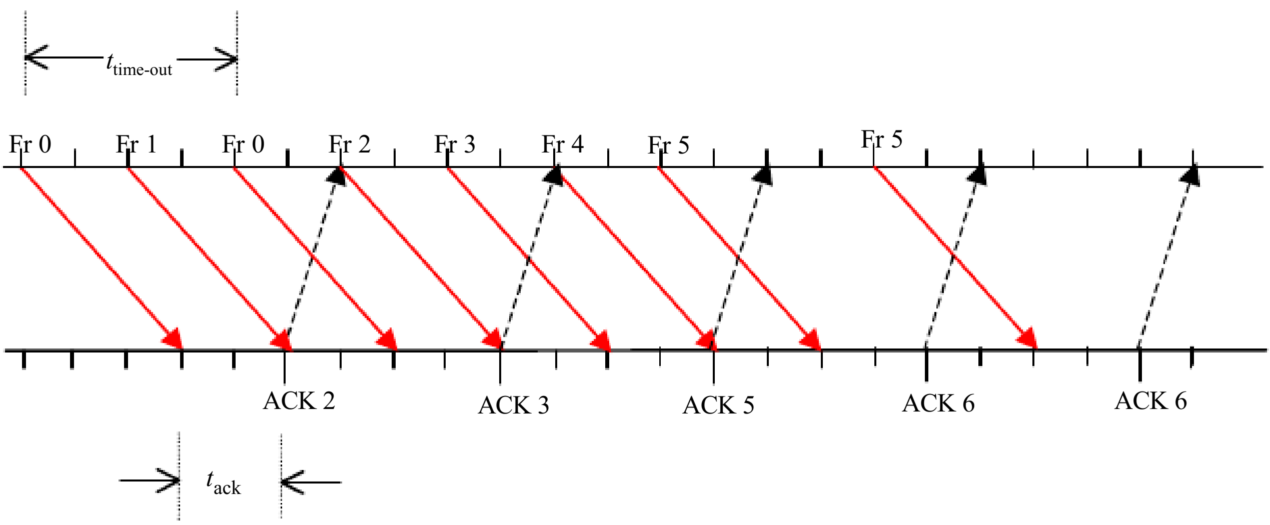
Bit Rate Mbps	Round Trip Time (msec)		
	100	10	1
1	34359.84	34359.75	34359.74
10	3436.07	3435.98	3435.97
100	343.70	343.61	343.60
1000	34.46	34.37	34.36

23. Consider a bidirectional link that uses Go-Back-N with $N = 7$. Suppose that all frames are one unit long and that they use a time-out value of 2. Assume the propagation is 0.5 unit and the processing time is negligible. Assume the ACK timer is one unit long. Assuming stations A and B begin with their sequence numbers set to zero, show the pattern of transmissions and associated state transitions for the following sequences of events:

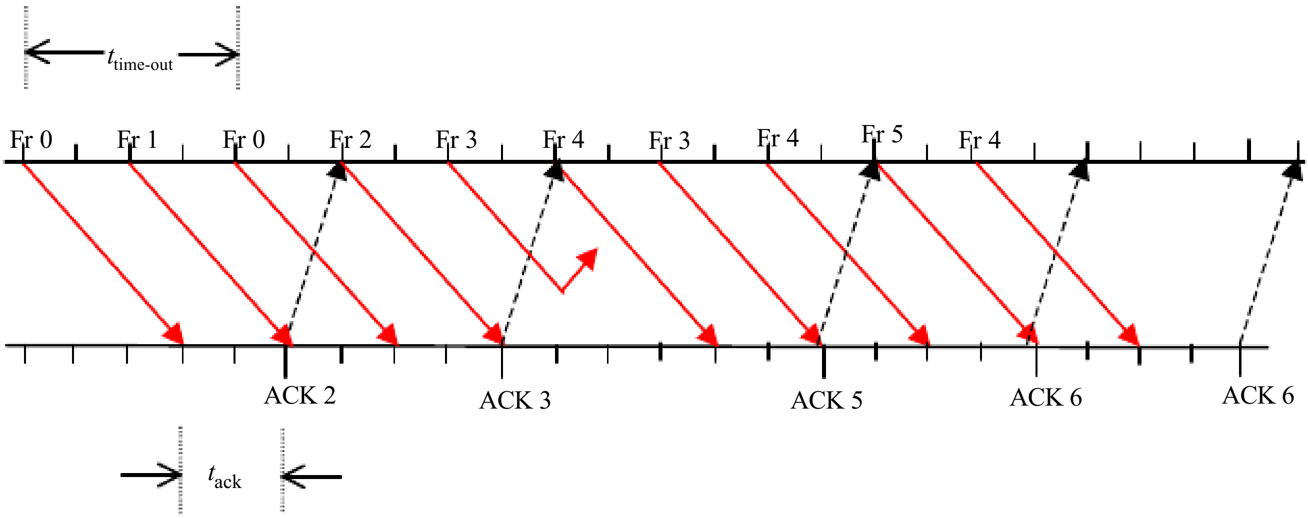
Solutions follow questions:

Go-Back-N with $N = 7$, $t_{timeout} = 2$, $t_{prop} = 0.5$, $t_{ACK} = 1$, $t_f = 1$. Each tick represents one half of a unit of time.

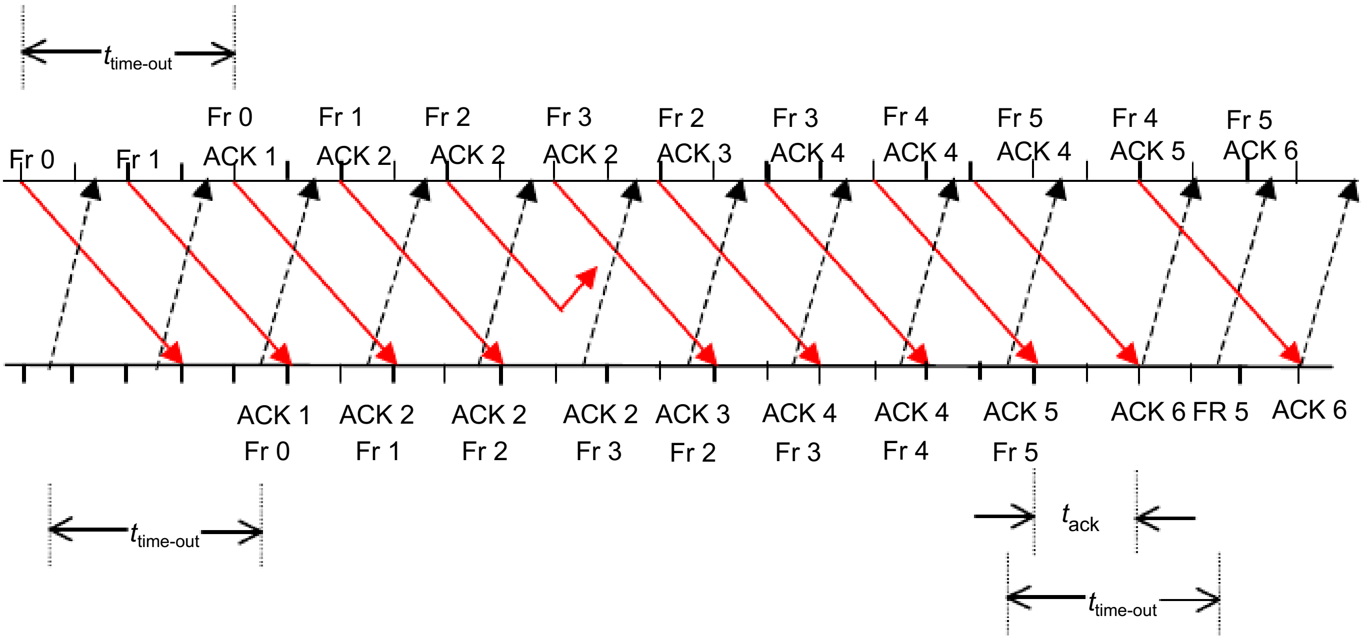
a. Station A sends six frames in a row, starting at $t = 0$. All frames are received correctly.



b. Station A sends six frame in a row, starting at $t = 0$. All frames are received correctly, except frame 3 is lost.



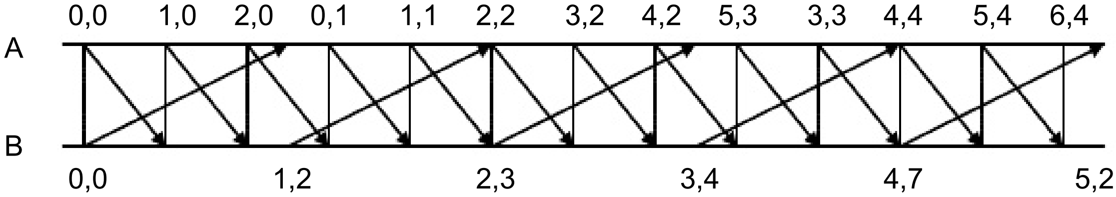
c. Station A sends six frames in a row, starting at $t = 0$. Station B sends six frames in a row starting at $t = 0.25$. All frames are received correctly.



24. Consider a bidirectional link that uses Go-Back-N with $N = 3$. Suppose that frames from station A to station B are one unit long and that they use a time-out value of 2, and that frames in the opposite directions are 2.5 units long and that they use a time-out value of 4. Assume propagation and processing times are negligible, that the stations have an unlimited number of frames ready for transmission, and that all ACKs are piggybacked onto information frames. Assuming stations A and B begin with their sequence numbers set to zero, show the pattern of transmissions and associated state transitions that result if there are no transmission errors.

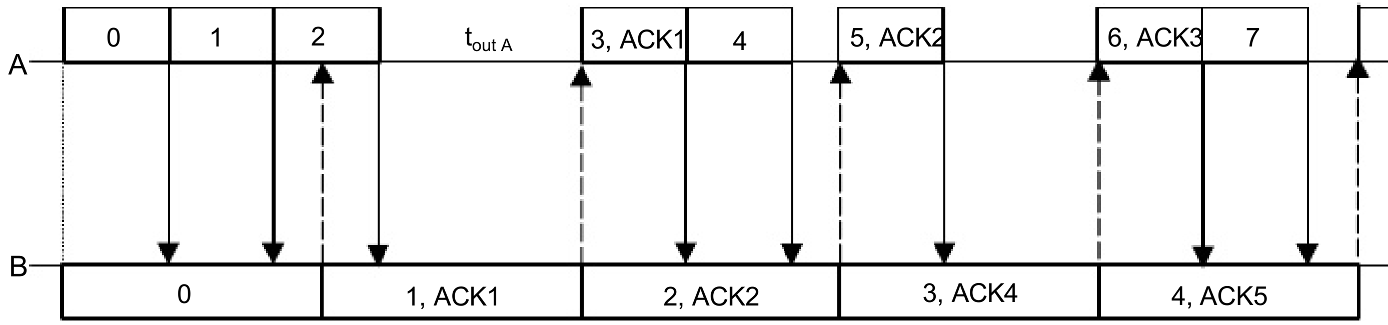
Solution:

The following figure shows a bidirectional link using Go-Back-N ARQ, where $N = 3$, and the ACKs are piggybacked. In addition, $n_A = 1$, $n_B = 2.5$, $t_{\text{out A}} = 2$, $t_{\text{out B}} = 4$, and $t_{prop} = 0$.



The state for each node is composed of the ordered pair (S_{last}, R_{next}). The global state of the system is composed by [A(S_{last}, R_{next}), B(S_{last}, R_{next})]

Node A retransmits because the ACK of node B arrives late.
Tail of arrow: Initiates transmission at sender
Head of arrow: arrival of last bit to the receiver



25. Consider the Go-Back-N ARQ protocol.

Solutions follow questions:

a. What can go wrong if the ACK timer is not used?

When no traffic arrives at a receiver during bidirectional Go-Back-N ARQ, and the receiver has to send an ACK, it usually sends the ACK after the ACK timer expires. If the ACK timer is not used, there are only two options remaining:

1) The ACK must be sent immediately (that is, never use piggybacking)

Although this will function correctly, it is an inefficient use of bandwidth in the general case.

2) The ACK must only be sent if it can be piggybacked

This is problematic if traffic arrives sporadically. The sender will wait a long time until a piggyback opportunity arises.

b. Show how the frame timers can be maintained as an ordered list where the time-out instant of each frame is stated relative to the time-out value of the previous frame.

Assume that the timer counts down from $t_{timeout}$. In order to have a separate timer for each frame, we need not implement *N* timers. Only the oldest frame can timeout. The system can save, for each frame, an arrival offset time that is related to the frame that preceded it and place these offsets in an ordered list based on the frame sequence numbers. If an ACK for the oldest frame arrives, the system simply increments the timer by the offset of the following frame in the list. If an ACK for any other frame arrives, the timer is incremented by the sum of all of the offsets in the list that are up to this newly acknowledged frame.

c. What changes if each frame is acknowledged individually instead of by using a cumulative acknowledgment (R_{next} acknowledges all frames up to $R_{next} - 1$)?

If each frame needs to be acknowledged individually, then the number of ACK messages will increase and the rate at which the transmission window can be increased will be reduced.

26. Suppose that instead of Go-Back-N ARQ, *N* simultaneous Stop-and-Wait ARQ processes are run in parallel over the same transmission channel. Each SDU is assigned to one of the *N* processes that is currently idle. The processes that have frames to send take turns transmitting in round-robin fashion. The frames carry the binary send sequence number as well as an ID identifying which ARQ process it belongs to. Acknowledgments for *all* ARQ processes are piggybacked onto *every* frame.

Solutions follow questions:

a. Qualitatively, compare the relative performance of this protocol with Go-Back-N ARQ and with Stop-and-Wait ARQ.

For simplicity assume that the time between consecutive frame transmissions in Stop-and-Wait corresponds to N consecutive transmissions without stopping. The parallel Stop-and-Wait procedure described above is an effective way to fill the transmission pipe without the additional complexity of Go-Back-N ARQ.

Vs. Go-Back N. Go-Back-N delivers frames in order. The parallel Stop-and-Wait protocol does not deliver frames in order, so additional processing is required if frames must be delivered in sequence. Because all the processes are independent, this protocol retransmits erroneous frames individually. In contrast, the Go-Back-N protocol retransmits a group of *N* frames. In this sense, the parallel protocol seems to perform similar to a Selective Repeat process.

Vs. Stop-And-Wait. If Stop-and-Wait is used, the effective bandwidth, without errors, will be *N* times less than the protocol described here. In fact, the larger *N*, the more efficient the protocol described is. At its worst case, where *N* = 1, it reduces to Stop-And-Wait.

b. How does the service offered by this protocol differ from that of Go-Back-N ARQ?

Assuming *N* connections are serviced in both cases, in Go-Back-N *N* outstanding frames are permitted no matter how many of the connections are actually sending traffic. In this protocol, when only *i* connections have traffic, the effective total send window is reduced to *i* frames.

27. Write a program for the transmitter and the receiver implementing Go-Back-N ARQ over a data link that can introduce errors in transmission.

Solutions follow questions:

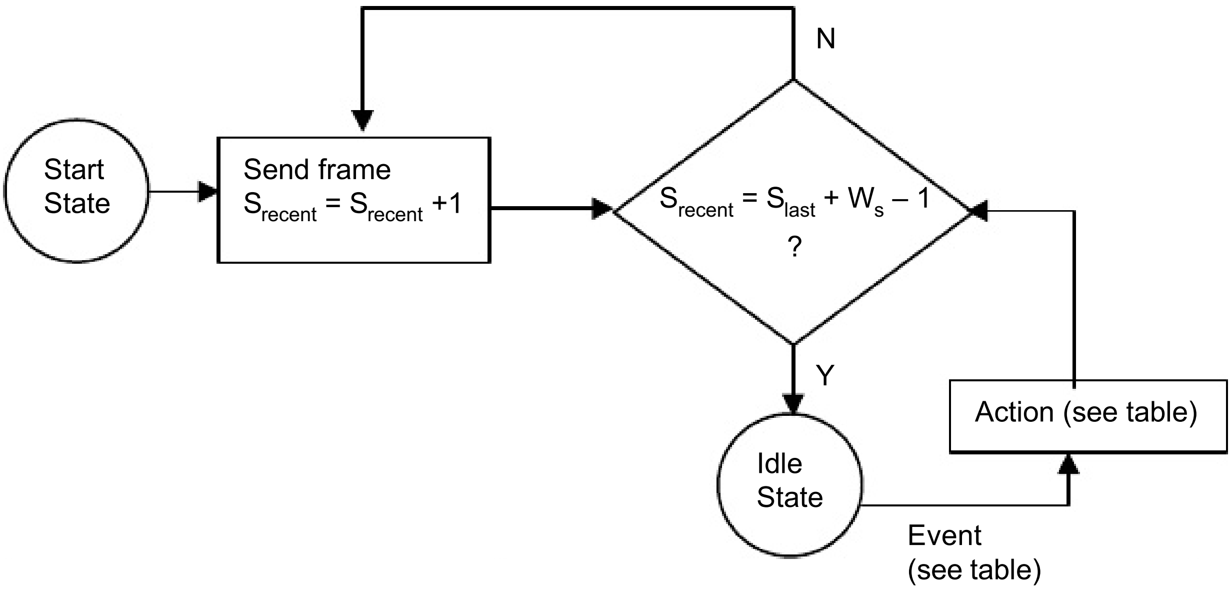
a. Identify what variables need to be maintained.

R_{next} : frame expected by receiver
S_{last} : oldest outstanding frame (back of window)
S_{recent} : most recently transmitted frame

b. The program loops continuously waiting for an event to occur that requires some action to take place. Identify the main events that can occur in the transmitter. Identify the main events that can occur in the receiver.

<u>Transmitter</u>

The following state diagram describes the transmitter's behavior. The table after it describes the events and their corresponding actions.



Event	Action
Receives an ACK with R_{next}	update S_{last} to R_{next}, update the timeout timer
Timeout expires	S_{recent} is updated to S_{last}

Receiver

The receiver is very simple. It remains idle until it receives a frame with R_{next}. At that point it increments R_{next} and sends the new R_{next} in an ACK frame.

28. Modify the program in problem 27 to implement Selective Repeat ARQ.

Solution:

Receiver

Event	Action
Receives a frame with R_{next}	R_{next} incremented as many frames as possible such that all frames with sequence number < R_{next} have already been successfully received. ACK sent with new R_{next}
Receives a frame with a sequence number not equal to R_{next}	Add this sequence number to the list of received frames Send ACK with R_{next}

Transmitter

The flow chart is identical to that of Problem 5.27. The only differences lie in the response to different events, as summarized in the following table:

Event	Action
Receives an ACK with R_{next}	Update S_{last} to R_{next} , Update each timer that belongs to a frame with a sequence number < R_{next}
Timeout expires on frame i	Resend frame *i* only
Receives an NAK with R_{next}	Resend frame R_{next}

29. Three possible strategies for sending ACK frames in a Go-Back-N setting are as follows: send an ACK frame immediately after each frame is received, send an ACK frame after every other frame is received, and send an ACK frame when the next piggyback opportunity arises. Which of these strategies are appropriate for the following situations?

Solutions follow questions:

a. An interactive application produces a packet to send each keystroke from the client; the server echoes each keystroke that it receives from the client.

Since each keystroke is echoed, there will always be a piggyback opportunity. Thus, the piggyback method should be used.

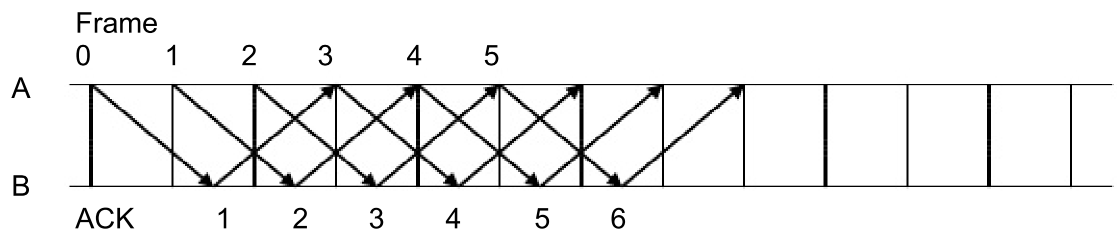
b. A bulk data transfer application where a server sends a large file that is segmented in a number of full-size packets that are to be transferred to the client.

The upstream traffic *to* a server is generally much less than the downstream traffic. Thus, the piggybacking method is non-ideal in this case. If the channel has a low probability of error, the alternating ACK method is better, as it saves bandwidth. However, if the connection causes frequent errors, every frame should be acknowledged. Additional overhead traffic is caused by the ACK frames, but will compensated by the bandwidth savings that will arise when the errors are discovered more quickly.

30. Consider a bidirectional link that uses Selective Repeat ARQ with a window size of $N = 4$. Suppose that all frames are one unit long and use a time-out value of 2. Assume that the one-way propagation delay is 0.5 time unit, the processing times are negligible, and the ACK timer is one unit long. Assuming station A and B begin with their sequence numbers set to zero, show the pattern of transmissions and associated state transitions for the following sequences of events:

Solutions follow questions:

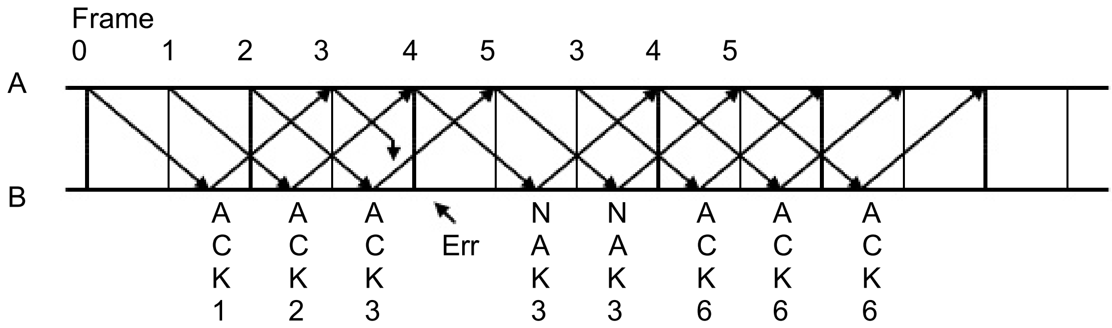
a. Station A sends six frames in a row, starting at $t = 0$. All frames are received correctly.



Tail of arrow: Initiates transmission at sender
Head of arrow: arrival of last bit at the receiver

Propagation delay = 0.5; Frame Length = 1; ACK Length = 1

b. Station A sends six frame in a row, starting at $t = 0$. All frames are received correctly, except frame 3 is lost.



Tail of arrow: Initiates transmission at sender
Head of arrow: arrival of last bit at the receiver
Propagation delay = 0.5; Frame Length = 1; ACK Length = 1

c. Station A sends six frames in a row, starting at $t = 0$. Station B sends six frames in a row starting at $t = 0.25$. All frames are received correctly.

31. In the chapter we showed that if the transmit and receive maximum window sizes are both equal to the available sequence number space, then Selective Repeat ARQ will work correctly. Rework the arguments presented in the chapter to show that if the sum of the transmit and receive maximum window sizes equals the available sequence number space, then Selective Repeat ARQ will work correctly.

Solution:

Assume $W_s + W_R = 2^m$ and assume that the current send window is 0 to $W_S - 1$. Suppose also that the receive window is 0 to $W_R - 1$. Now suppose that frame 0 is received correctly but that the acknowledgment for frame 0 is lost. The transmitter can transmit new frames only up to frame $W_S - 1$. Depending on which transmission arrive without error, R_{next} will be in the range between 1 and W_s while $R_{next} + W_R - 1$ will be in the range of 1 to $W_R + W_S - 1$. The maximum value of R_{next} occurs when frames 0 through $W_S - 1$ are received correctly, in which case the value of R_{next} is W_S and the value of R $R_{next} + W_R - 1$ increases to $W_R + W_S - 1$. Crucially, the receiver will not receive frame $W_R + W_S$ until the acknowledgment for frame 0 has been received at the transmitter. Any receipt of frame 0 prior to frame $W_R + W_S$ indicates a duplicate transmission of frame 0. Therefore, the sum of the maximum window sizes is 2^m.

32. Suppose that Selective Repeat ARQ is modified so that ACK messages contain a list of the next *m* frames that it expects to receive.

Solutions follow questions:

a. How does the protocol need to be modified to accommodate this change?

First, the frame header needs to be modified to accommodate the list of frames to receive. It can be a fixed or a variable number of slots. NAK won't be necessary because the receiver explicitly indicates which frames need to be transmitted.

b. What is the effect of the change on protocol performance?

The performance will increase in cases of multiple errors or in cases where the delay is high. A single frame can ask for the retransmission of several frames. The drawback is the overhead in the header and the increased protocol complexity relative to pure Selective-Repeat ARQ.

33. A telephone modem is used to connect a personal computer to a host computer. The speed of the modem is 56 kbps and the one-way propagation delay is 100 ms.

Solutions follow questions:

First we assume the following:

$P_f = 1 - (1 - 10^{-4})^{nf}$
$n_f = 256 \times 8 = 2048$ or $n_f = 512 \times 8 = 4096$
$t_{prop} = 100$ ms

$n_o = 0$
$n_a = 64$ bits
$t_{proc} = 0$

a. Find the efficiency for Stop-and-Wait ARQ if the frame size is 256 bytes; 512 bytes. Assume a bit error rate of $10^{-4.}$

Using the results in Table 5.2,

$$\eta = (1 - P_f)\frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})}{n_f}R}$$

$= 0.125$ ($n_f = 2048$)

$= 0.177$ ($n_f = 4096$)

b. Find the efficiency of Go-Back-N if three-bit sequence numbering is used with frame sizes of 256 bytes; 512 bytes. Assume a bit error rate of $10^{-4.}$

Given that $W_S = 2^3 = 8$, we can calculate that the window size is:

$$\frac{n_f \times W_S}{R} = 290ms$$

Since this is greater than the round trip propagation delay, we can calculate the efficiency by using the results in Table 5.2.

$$\eta = (1 - P_f)\frac{1 - \frac{n_o}{n_f}}{1 + (W^S - 1)P^f}$$

$= 0.355$ ($n_f = 2048$)

$= 0.198$ ($n_f = 4096$)

34. A communications link provides 1 Mbps for communications between the earth and the moon. The link is used to send color images from the moon. Each image consists of 10,000 x 10,000 pixels, and 16 bits are used for each of the three-color components of each pixel.

Solutions follow questions:

a. How many images per second can be transmitted over the link?

The number of images that can be transmitted per second is:

$$1\times 10^{6}\frac{bits}{\sec}\Big/(10000^{2}\times 16\times 3\frac{bits}{image}) = 2.1\times 10^{-4}\,bits/image$$

b. If each image is transmitted as a single block, how long does it take to get an acknowledgment back from earth? The distance between earth and the moon is approximately 375,000 km.

The total time to get an acknowledgment from earth, assuming that $t_{ACK} << t_f$ is:

$$t_O = t_f + 2t_{prop} = \frac{(10000^{2}\times 16\times 3\frac{bits}{image})}{1\times 10^{6}\frac{bits}{\sec}} + 2\frac{375000\times 10^{3}\,m}{3\times 10^{8}\frac{m}{\sec}}$$

$$= 4800 + 1.25 = 4800\,\sec/image$$

Note that if each image is transmitted in a single block, t_{prop} becomes insignificant compared to t_f.

c. Suppose that the bit error rate is 10^{-5}, compare Go-Back-N and Selective Repeat ARQ in terms of their ability to provide reliable transfer of these images from the moon to earth. Optimize your frame size for each ARQ protocol.

We know that in order to optimize efficiency, W_S should be chosen such that, without error, packets can be continuously transmitted.

For Go-Back-N, ignoring t_{proc}, we have:

$$W_S = \frac{2t_{prop} + t_{ack} + t_f + 2t_{proc}}{t_f} = \frac{2t_{prop} + t_{ack}}{t_f} + 1$$

$$= \frac{2(1.25) + \frac{64}{1\times 10^{6}}}{\frac{n_f}{1\times 10^{6}}} + 1 = \frac{3\times 10^{6}}{n_f} + 1$$

$$P_f = 1 - (1 - 10^{-5})^{n_f}$$

We have assumed that n_{ACK} = 64 bits.

To calculate the transmission efficiency, we can now use the result from Table 5.2.

$$\eta = (1-10^{-5})^{n_f} \frac{1-\frac{n_o}{n_f}}{1+(\frac{3\times10^6}{n_f})[1-(1-10^{-5})^{n_f}]}$$

$$\eta = (1-10^{-5})^{n_f} \frac{n_f - n_o}{n_f + (3x10^6)[1-(1-10^{-5})^{n_f}]}$$

This optimum value of n_f can be found by taking the derivative with respect to n_f and equating the result to zero. (See problem 5.36).

In this case, assuming that n_0 = 20 bytes we get that

η_{max} = 3.04 (when n_f = 5619bits)

For the case of Selective Repeat, assuming an ideal window size, we have:

$$\eta = (1-10^{-5})^{n_f}(1-\frac{n_0}{n_f})$$

In this case, assuming again that n_0 = 20 bytes we get that

η_{max} = 92.24% (when n_f = 4080 bits)

35. Two computers are connected by an intercontinental link with a one-way propagation delay of 100 ms. The computers exchange 1-Megabyte files that they need delivered in 250 ms or less. The transmission lines have a speed of R Mbps and the bit error rate is 10^{-8}. Design a transmission system by selecting the bit rate R, the ARQ protocol, and the frame size.

Solution:

Using Selective Repeat ARQ for efficiency's sake and assuming that the overhead n_o = 64 bits, we need to select n_f and R to ensure delivery in 250 ms or less.

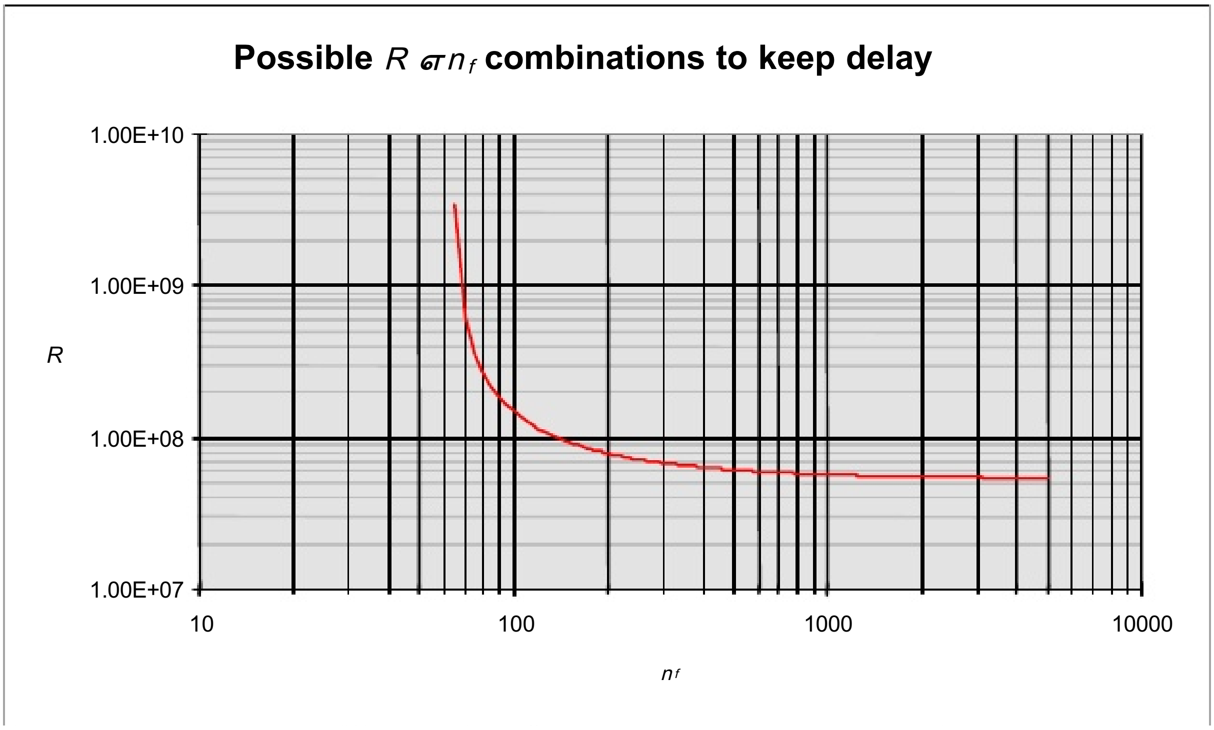
$$\text{Total delay } = 2t_{prop} + \frac{8\times10^6}{R^o_{eff}} = 200\times10^{-3} + \frac{8\times10^6}{\eta R} < 250\times10^{-3}$$

$$\eta R > 6.25\times10^{-9}$$

Using the results of Table 5.2, we see that

$$(1-10^{-8})^{n_f}(1-\frac{64}{n_f})R > 53.33\times10^6$$

The allowable region is shown below as the area under the graph.



36. Find the optimum frame length that maximizes transmission efficiency by taking the derivative and setting it to zero for the following protocols:

Solutions follow questions:

a. Stop-and-Wait ARQ.

$$\eta = (1-P_b)^{n_f} \frac{n_f - n_o}{n^f + n^a + 2(t^{proc} + t^{prop})R}$$

Let $a = (1 - P_b)$
Let $b = 2\,(t_{proc} + t_{prop})\,R + n_a$

$$\frac{d}{dn_f}[a^{n_f}\frac{n_f - n_o}{n_f + b}] = \frac{a^{n_f}[\ln a(n_f - n_o) + 1](n_f + b) - a^{n_f}(n_f - n_o)}{(n_f + b)^2} = 0$$

$$a^{n_f}\{\ln a[n_f^2 + (b - n_o)n_f - n_o b] + n_f + b - n_f + n_o = 0$$

$$n_f^2 + (b - n_o)n_f - n_o b] + n_f + \frac{b - n_f + n_o b}{\ln a} = 0$$

$$n_f = \frac{(n_o - b) \pm \sqrt{(b - n_o)^2 - 4\frac{(b - n_o - n_o b)}{\ln a}}}{2}$$

b. Go-Back-N ARQ.

Borrowing the result from problem 34(c)

$$\eta = (1-P_b)^{n_f} \frac{n_f - n_o}{n_f + n_f[1-(1-P_b)^{n_f}][\frac{2R(t_{proc}+t_{prop})}{n_f}]}$$

Let a = (1 – P_b)
Let b = 2R (t_{proc} + t_{prop}) and use the approximation $1-(1-p)^n \approx np$, then

$$\eta = \frac{a^{n_f}(n_f - n_o)}{n_f + n_f P_b b} = \frac{a^{n_f}(n_f - n_o)}{n_f[1+P_b b]}$$

Taking the derivative and equating the numerator to zero, we find that:

$$0 = \frac{d\eta}{dn^f} = \frac{a^{n_f}(n_f - n_o)}{n^f} = \frac{(a^{n_f}\ln a(n_f - n_0) + a^{n_f})n_f - a^{n_f}(n_f - n_0)}{n^{f\,2}}$$

which leads to the quadratic equation

$$n_f^2 - n_o n_f + n_o / \ln a = 0$$

which gives the solution:

$$n_f = \frac{n_o \pm \sqrt{n_o^{\,2} - 4\frac{n_o}{\ln a}}}{2}$$

c. Selective Repeat ARQ.

Borrowing the result from problem 34c:

$$\eta = (1-P_b)^{n_f}(1 - \frac{n_0}{n_f})$$

Except for a scale factor, this equation is the same as the approximation for η found in part (b). The solution for the optimum frame size has the same form.

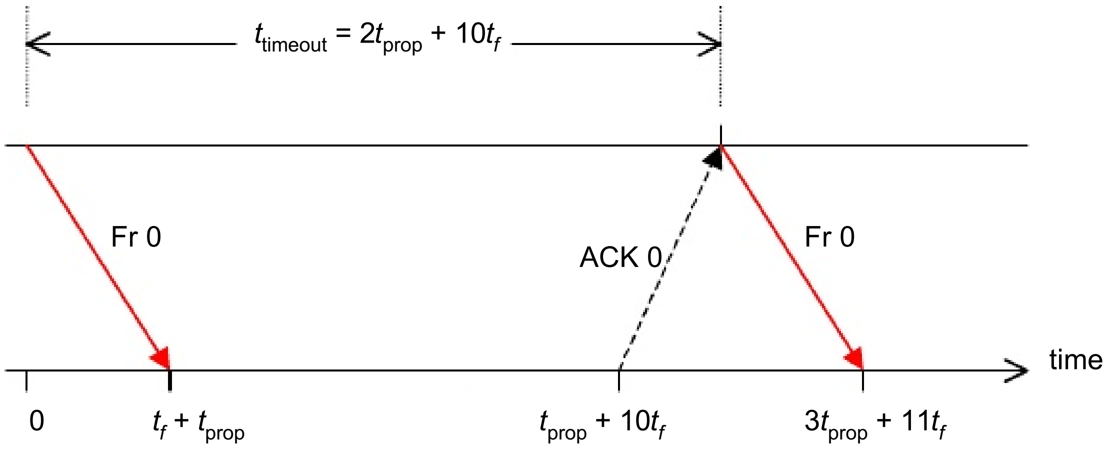
37. Suppose station A sends information to station B on a data link that operates at a speed of 10 Mbps, and that that station B has a 1-Megabit buffer to receive information from A. Suppose that the application at station B reads information from the receive buffer at a rate of 1 Mbps. Assuming that station A has an unlimited amount of information to send, sketch the sequence of transfers on the data link if Stop-and-Wait ARQ is used to prevent buffer overflow at station B. Consider the following cases:

a. One-way propagation delay is 1 microsecond.

b. One-way propagation delay is 1 ms.

c. One-way propagation delay is 100 ms.

Solution:

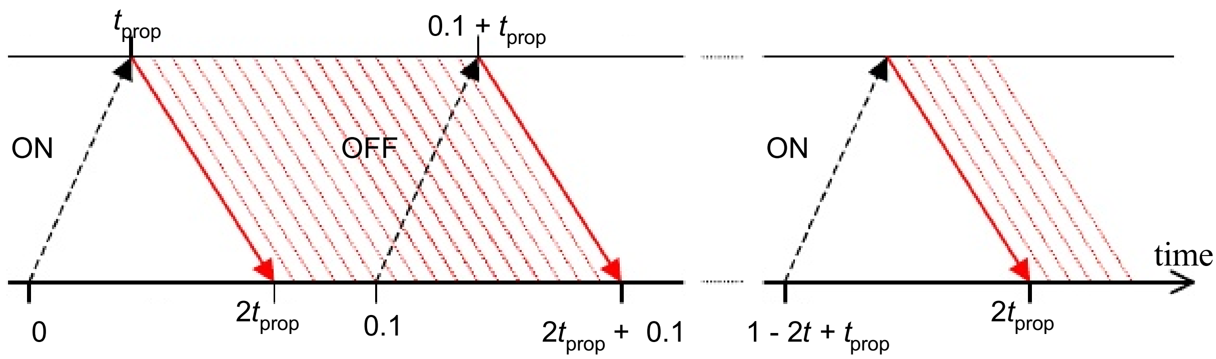


t_f depends on the size of the packets. However, assuming that the packets are 1 Mbit in order to maximize efficiency, t_f = 100 ms. For parts (a), (b) and (c), the sequence of events will look the same as above.

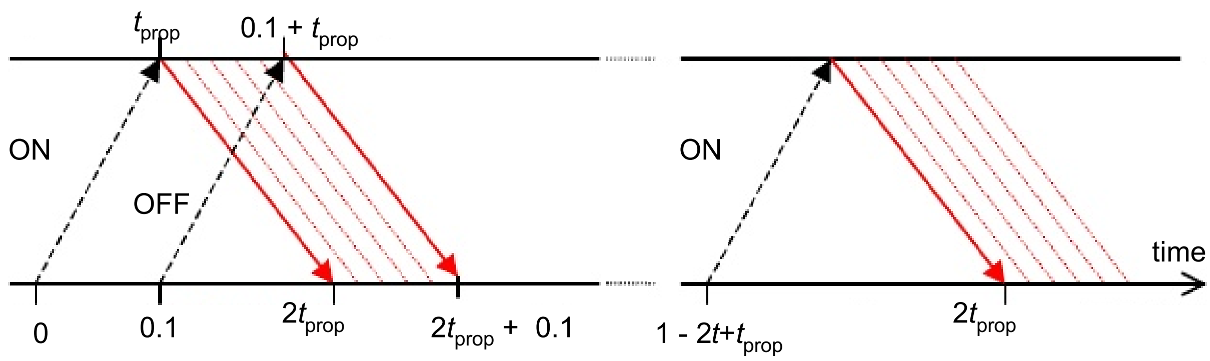
38. Redo problem 37 using Xon/Xoff flow control.

Solution:

If t_{prop} is less than 50ms (parts (a) and (b)), the following picture holds.



If t_{prop} is greater than 50ms, the following picture is more indicative of the sequence of events:

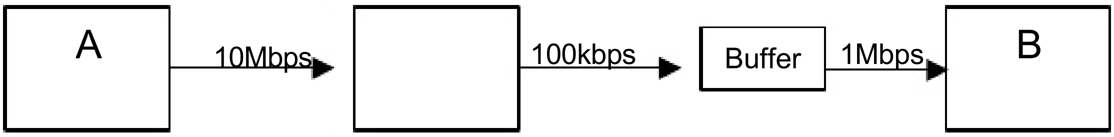


39. Suppose station A sends information to station B over a two-hop path. The data link in the first hop operates at a speed of 10 Mbps, and the data link in the second hop operates at a speed of 100 kbps. Station B has a 1-Megabit buffer to receive information from A, and the application at station B reads information from the receive buffer at a rate of 1 Mbps. Assuming that station A has an unlimited amount of information to send, sketch the sequence of transfers on the data link if Stop-and-Wait ARQ is used on an end-to-end basis to prevent buffer overflow at station B.

a. One-way propagation delay in data link 1 and in data link 2 are 1 ms.

b. One-way propagation delay in data link 1 and in data link 2 are 100 ms.

Solution:

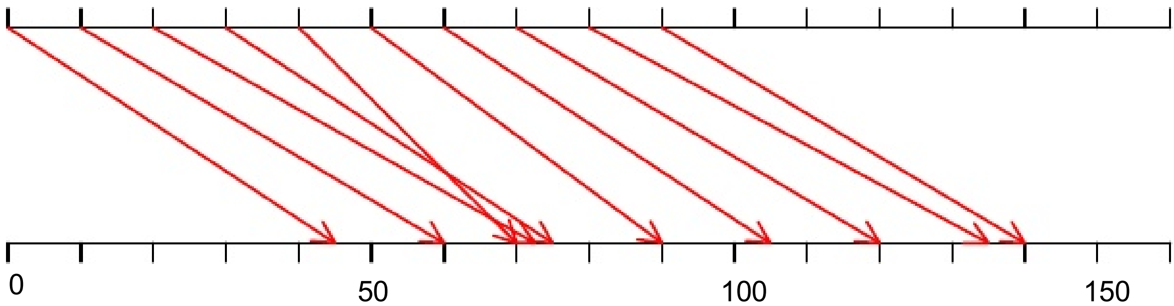


Assume that a frame is n_f bits long, then the frame transmission time in link 1 is $t_1 = n_f/10^7$ seconds, and $t_2 = n_f/10^5$ seconds on the second link. Therefore the time to transmit a frame over the second link is 100 times greater than the time to transmit over the first link. For example if n_f = 10000 bits, then t_1 = 1 ms and t_2 =100 ms. Consequently, the node between link 1 and link 2 must be capable of buffering the bits that arrive in link 1 but which must wait for transmission over link 2. If we assume that the frame is buffered before node until it has completely arrived, then it will take $t_3 = n_f/10^6$ seconds to read out. In the above example, we have t^3 = 10 ms. The ACK message for Stop-and-Wait can then be sent. Let $t_f=t_1+t_2+t_3$. The sequence of events is then as follows: a frame that begins transmission at time 0 will be received in its entirety at node B at time $t_f + t_{prop}$; an ACK message is send thereafter, and the message arrives at the transmitter at approximately time t_f+2t_{prop}.

40. A sequence of fixed-length packets carrying digital audio signal is transmitted over a packet network. A packet is produced every 10 ms. The transfer delays incurred by the first 10 packets are 45 ms, 50 ms, 53 ms, 46 ms, 30 ms, 40 ms, 46 ms, 49 ms, 55 ms, 51 ms.

Solutions follow questions:

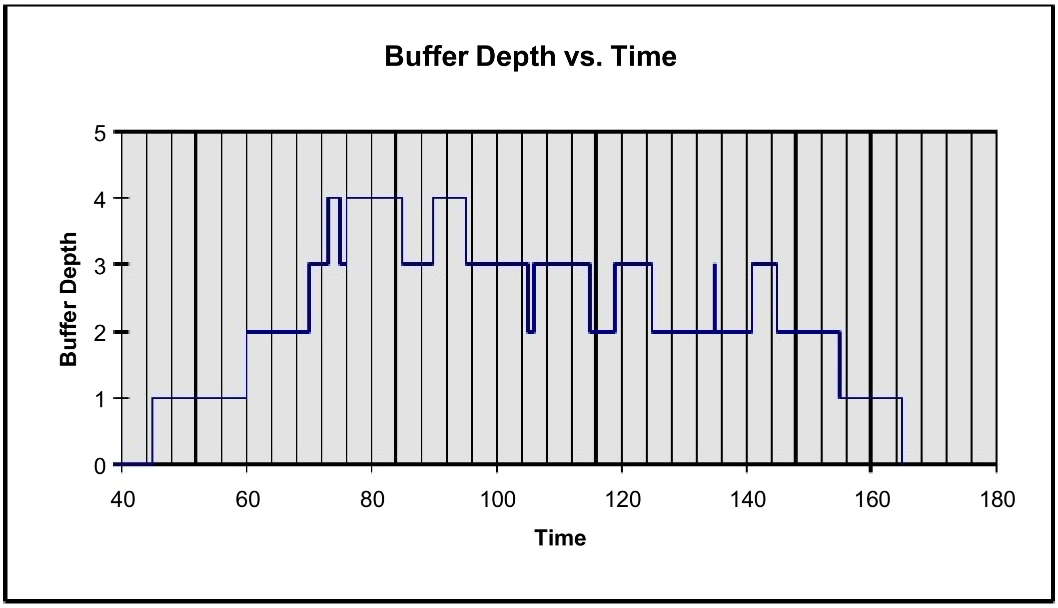
a. Sketch the sequence of packet transmission times and packet arrival times.



b. Find the delay that is inserted at the receiver to produce a fixed end-to-end delay of 75 ms.

Packet #	0	1	2	3	4	5	6	7	8	9
Transmission Time	0	10	20	30	40	50	60	70	80	90
Delay	45	50	53	46	30	40	46	49	55	51
Arrival Time	45	60	73	76	70	90	106	119	135	141
Desired playout time	75	85	95	105	115	125	135	145	155	165
Delay Inserted	**30**	**25**	**22**	**29**	**45**	**35**	**29**	**26**	**20**	**24**

c. Sketch the contents of the buffer at the receiver as a function of time.



41. Consider an application in which information that is generated at a constant rate is transferred over a packet network so timing recovery is required at the receiver.

Solutions follow questions:

a. What is the relationship between the maximum acceptable delay and the playout buffer?

The length of the playout buffer determines the maximum playout delay that will occur (i.e. when the buffer is full). This maximum buffer length is therefore determined by the maximum acceptable playout delay.

b. What is the impact of the bit rate of the application information stream on the buffer requirements?

The maximum delay incurred by the buffering is explicitly equal to

Max buffer delay = buffer size (in bits) / playout rate

Since the playout rate is equal to the bit rate of the application information stream, we see that for a given acceptable maximum playout delay, the buffer size is directly determined by the playout rate.

c. What is the effect of jitter on the buffer size design?

In general, to reduce playback delay, the buffer length should be kept as small as possible. However, the buffer must be able to accommodate the jitter of the incoming traffic stream.

Jitter has the effect of making the incoming data rate variable. If the stream has high jitter (i.e. high delay variations), then the buffer must be made large to accommodate times when the incoming data rate may be higher than the playback rate. From another perspective, if the jitter is large, there is a large chance that many packets will arrive after a propagation delay that is significantly smaller than that of the other packets in the stream. These packets must be buffered until it is their turn to be played out.

42. A speech signal is sampled at a rate of 8000 samples/second. Packets of speech are formed by packing 10 ms worth of samples into each payload. Timestamps are attached to the packets prior to transmission and used to perform error recovery at the receiver. Suppose that the time stamp is obtained by sampling a clock that advances every Δ seconds. Is there a minimum value that is required for Δ? If so what is it?

Solution:

The clock is sampled every 10 ms to produce a timestamp. Clearly, we require that Δ < 10 ms in order to ensure a unique timestamp for each packet. However, if the timestamps are to be used for error detection, a Δ of just under 10 ms is not sufficient. To illustrate, suppose that the clock advances every Δ = 9 ms. Then the times at which the clock advances will be as follows:

Clock tick:	0	9	18	27	36	45	54	63	72	81	90
Count:	0	1	2	3	4	5	6	7	8	9	10

The time stamps that result from sampling every 10 ms will be:

Sample:	0	10	20	30	40	50	60	70	80	90	100
Stamp:	0	1	2	3	4	5	6	7	8	10	11

We see that the sample at time 90 will have a time stamp of 10, which is 2 units larger than the time stamp at time 80. If a packet is lost, a gap in the timestamps will be detected by the receiver. However for the example, the receiver may not be able to decide whether a packet was lost or whether the phenomenon above occurred.

While it is true that this problem can be overcome because of the periodic nature of the above phenomenon, a more reliable solution is to place the constraint that Δ < 5ms. With this constraint in place each timestamp will be offset by 2Δ or possibly 3Δ (in the case of an extra skipped clock pulse). If a packet is lost, a timestamp offset of 4Δ or possibly 5Δ will be detected at the receiver, which is completely unambiguous.

43. Suppose that PDUs contain both timestamps and sequence numbers. Can the timestamps and sequence numbers be combined to provide a larger sequence number space? If so, should the timestamps or the sequence numbers occupy the most significant bit locations?

Solution:

Yes. Two variations are possible. The first is to transmit groups of consecutive packets with a common sequence number and different time stamps. The second is to transmit groups of consecutive packets with a common time stamp and different sequence numbers.

The former is preferable for the reasons described in Problem 42. Since time stamps periodically skip over one clock increment, by having the timestamps occupy the least significant bit locations, the loss to the sequence number space is minimized.

44. Consider the timestamp method for timing recovery discussed in the chapter.

Solutions follow questions:

a. Find an expression that related the difference frequency Δf to the number of cycles M and N.

$$\Delta f = f_n - f_s = f_n\left(1 - \frac{M}{N}\right)$$

b. Explain why only M needs to be sent.

f_n is globally known and N is agreed on beforehand (as a standard or else during connection setup). Thus, only M needs to be sent.

c. Explain how the receiver uses this value of M to control the playout procedure.

The procedure plays out frames at a rate:

$$f_r = f_n - \Delta f = f_n \left(1 - \frac{M}{N}\right) = \frac{M}{N} f_n$$

45. A 1.5 Mbps communications link is to use HDLC to transmit information to the moon. What is the smallest possible frame size that allows continuous transmission? The distance between earth and the moon is approximately 375,000 km, and the speed of light is 3×10^8 meters/second.

Solution:

The round trip propagation delay is:

$$2t_{prop} = 2\,\frac{(375 \times 10^6\, m)}{3 \times 10^8\, m/s} = 2.50 \sec$$

To allow for continuous transmission, we must use Go-Back-N or Selective Repeat.

Go-Back-N:

If $N = 7$ → $\dfrac{7 n_f}{1.5 Mbps} = 2.5s$ → n_f = 535 715 bits

If $N = 127$ → $\dfrac{127 n_f}{1.5 Mbps} = 2.5s$ → n_f = 29 528 bits

Selective Repeat:

If $N = 4$ → $\dfrac{4 n_f}{1.5 Mbps} = 2.5s$ → n_f = 973 500 bits

If $N = 64$ → $\dfrac{64 n_f}{1.5 Mbps} = 2.5s$ → n_f = 58 594 bits

46. Perform the bit stuffing procedure for the following binary sequence: 1101111111011111110101.

Solution:

The inserted stuff bits are underlined.

1101111111011111110101 → 11011111<u>0</u>11011111<u>0</u>110101

47. Perform bit destuffing for the following sequence: 11101111101111101111110.

Solution:

The removed stuff bits are indicated by a '-'.

11101111101111101111110 → 111011111-11111-1111110

48. Suppose HDLC is used over a 1.5 Mbps geostationary satellite link. Suppose that 250-byte frames are used in the data link control. What is the maximum rate at which information can be transmitted over the link?

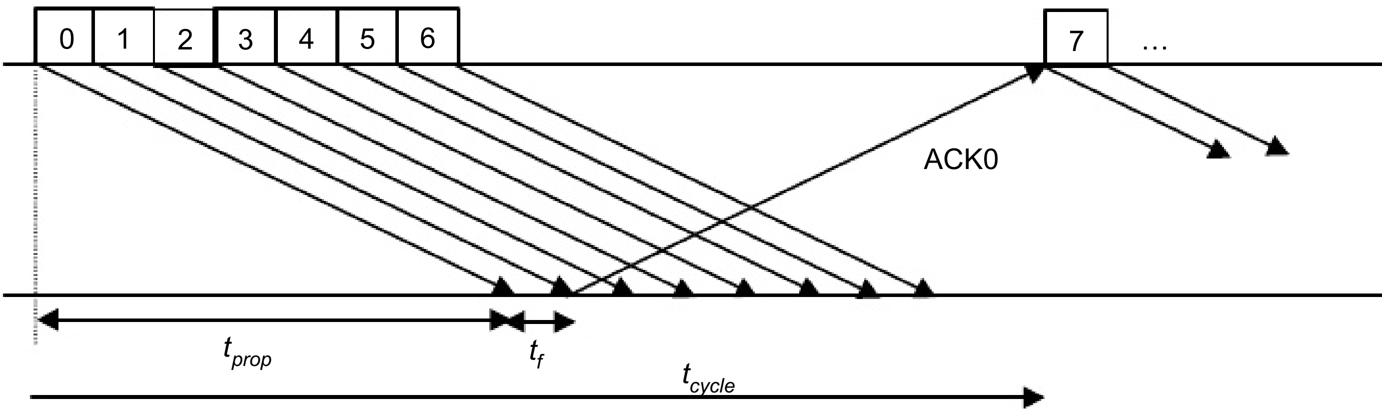
Solution:

R = 1.5 Mbps, and n_f =250 bytes or 2000 bits (250 x 8). The distance that the information must travel is the earth-to-satellite distance, or $d \approx$ 36,000 km. The speed of light c is 3 x 10^8. We can calculate the propagation delay and processing rate as follows:

$t_{prop} = d/c = 36 \times 10^6 / 3 \times 10^8 = 120$ ms

$t_f = n_f/R = 2000/1.5 \times 10^6 = 1.33$ ms

We can use either Go-Back-N or Selective Repeat ARQ. The default window size is N = 7 (with a 3-bit sequence number).



The maximum information rate is achieved with no error, and hence, no retransmission.

t_{cycle} = minimum time to transmit a group of N packets

$= t_f + 2\ t_{prop} = 1.33 + 2\text{x}2120 = 241.33$ ms

n = no. of bits transmitted in a cycle = $N.n_f$ = 7x2000 = 14,000 bits

R_{max} = no. of bits sent in a cycle / minimum cycle time = n/t_{cycle} = 58 kbps

If the extended sequence numbering option (7-bit) is used, the maximum send window size would be $N = 2^7 - 1 = 127$, and hence, the maximum information rate is:

$R_{max} = N.n_f / t_{cycle} = 127\text{x}2000/(241.33\text{x}10^{-3}) = 1.052$ Mbps

49. In HDLC how does a station know if a received frame with the fifth bit set to 1 is a P or an F bit?

Solution:

If the station is a secondary station, the bit is a 'P' (and the station is being *Polled* for more frames). If it is a primary station, the bit is a 'F' bit (indicating the *Final* frame of the current transmission).

50. Which of the following statements are incorrect?

Solutions follow questions:

For this question, one must just keep in mind that all frames always contain the address of the secondary station.

a. A transmitting station puts its own address in command frames.

Incorrect. Command frames are destined for secondary stations, so this transmitter would put the address of the secondary station in the frame.

b. A receiving station sees its own address in a response frame.

Incorrect. Responses come from secondary stations to primary stations. Thus the address in the frame is that of the sender in this case.

c. A response frame contains the address of the sending station.

TRUE. Response frame originate in secondary stations, so they will have contain the address of the sender.

51. In HDLC suppose that a frame with P = 1 has been sent. Explain why the sender should not send another frame with P = 1. What should be done to deal with the case where the frame is lost during transmission?

Solution:

The poll bit is like a token that is passed to a secondary so it can transmit. At the end of transmission the secondary must return it in the form of final bit.

If the poll frame is lost then the token no longer exists and no station can transmit. The server must create a new token (poll frame) only after it is sure that the previous poll frame is lost and not just delayed. Consequently it must wait for certain period of time.

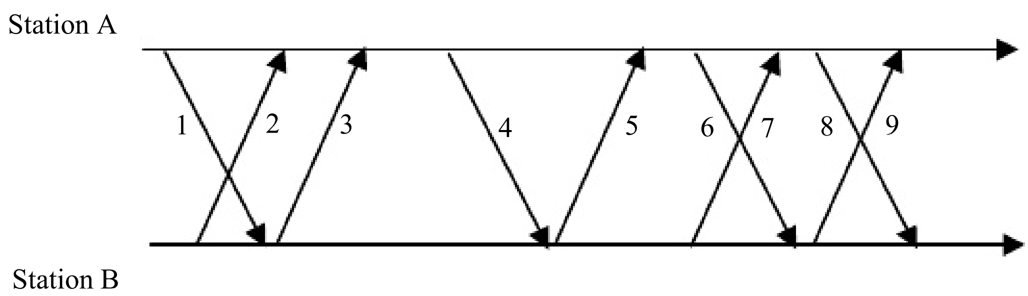
52. HDLC specifies that the N(R) in a SREJ frame requests the retransmission of frame N(R) and also acknowledges all frames up to N(R) – 1. Explain why only one SREJ frame can be outstanding at a given time.

Solution:

Suppose two outstanding SREJ frames exist. Let frame A have N(R) = m and frame B have N(R) = n. Without loss of generality, suppose $n > m$.

Since each SREJ frame with value N(R) implicitly acknowledges all previous frames up to N(R) – 1, frame A indicates that frame m has not yet been received and frame B indicates that frame m has been received. Thus, if two SREJ are allowed to be outstanding at the same time, contradictory information will be sent to the receiver.

53. The following corresponds to an HDLC ABM frame exchange with no errors.



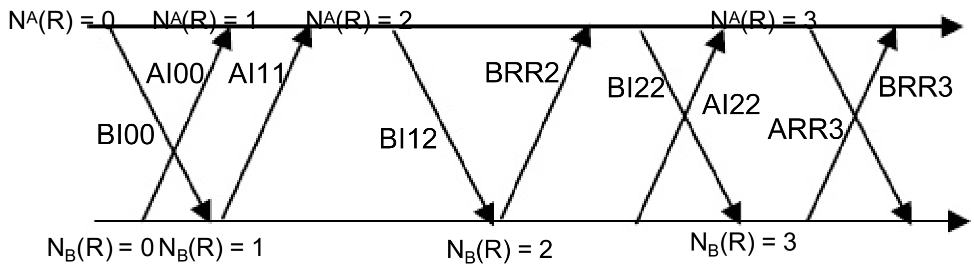
1. BI00 2. AI00 3. xIxx 4. xIxx 5. xRRxx

6. xIxx 7. xIxx 8. xRRxx 9. xRRxx

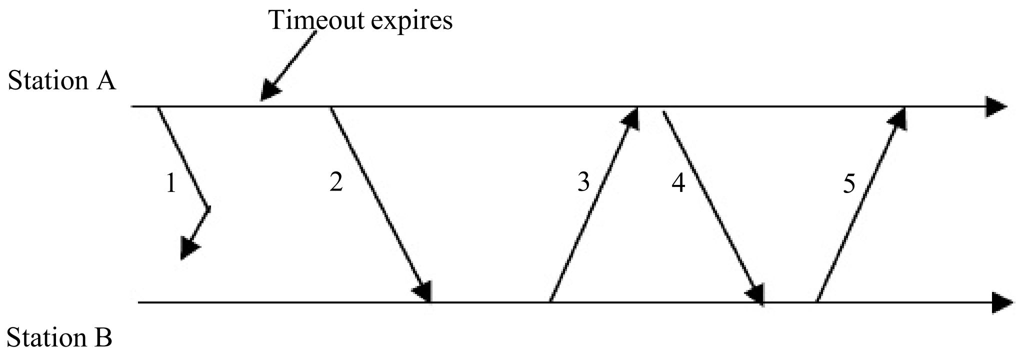
a. Complete the diagram by completing the labeling of the frame exchanges.

b. Write the sequence of state variables at the two stations as each event takes place.

Solution:



54. Assume station B is awaiting frame 2 from station A.



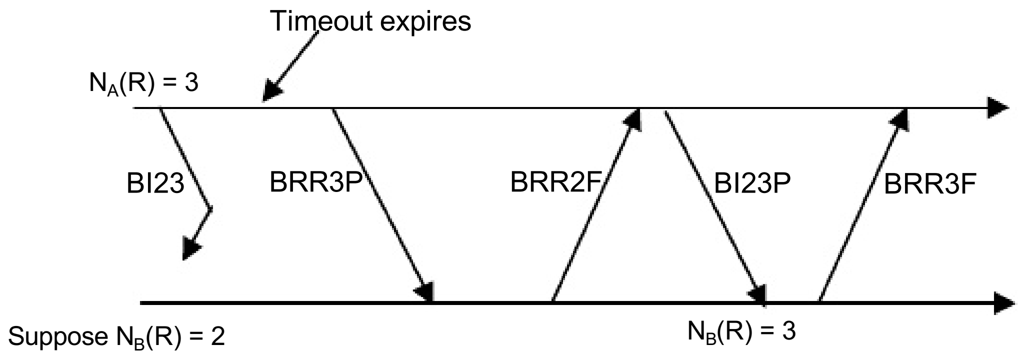
1. BI23 2. xRRxP 3. xRRxy 4. xIxx 5. xRRx

a. Complete the diagram in HDLC ABM by completing the labeling of the frame exchanges.

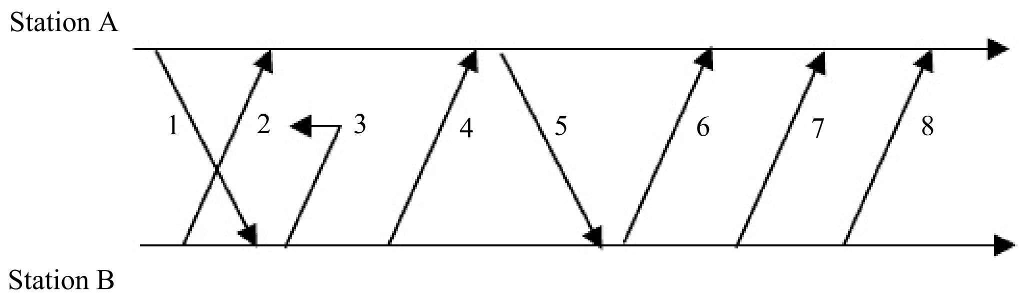
b. Write the sequence of state variables at the two stations as each event takes place.

Solution:

This sequence of events appears to be Stop-And-Wait.



55. The following corresponds to an HDLC ABM frame exchange.



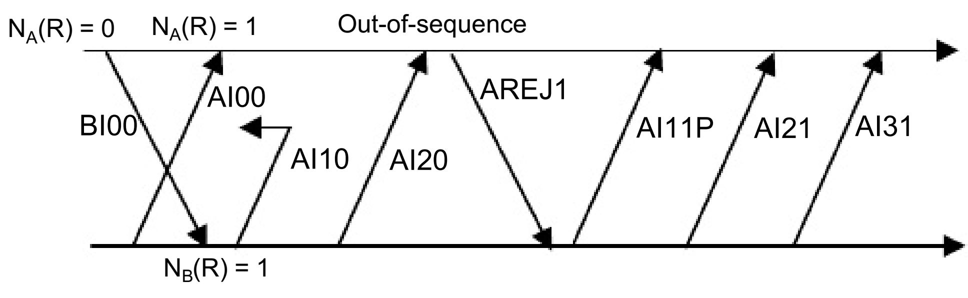
1. BI00   2. AI00   3. xIxx   4. xIxx

5. xREJx   6. xIyx   7. xyxx   8. xyxx

a. Complete the diagram by completing the labeling of the frame exchanges.

b. Write the sequence of state variables at the two stations as each event takes place.

Solution:



56. The PPP byte stuffing method uses an escape character defined by 0x7D (01111101). When the flag is observed inside the frame, the escape character is placed in front of it and the flag is exclusive-ORed with 0x20. That is, 0x7E is encoded as 0x7D 0x5E. An escape character itself (0x7D) is encoded as 0x7D 0x5D. What are the contents of the following received sequence of bytes after byte destuffing:

0x7D 0x5E 0xFE 0x24 0x7D 0x5D 0x7D 0x5D 0x62 0x7D 0x5E

Solution:

0x7D 0x5E 0xFE 0x24 0x7D 0x5D 0x7D 0x5D 0x62 0x7D 0x5E

→ 0x7E 0xFE 0x24 0x7D 0x7D 0x62 0x7E

57. Suppose that a 1-Megabyte message is sent over a serial link using TCP over IP over PPP. If the speed of the line is 56 kbps and the maximum PPP payload is 500 bytes, how long does it take to send the message?

Solution:

Assuming the overhead in one packet is equal to 8 bytes for the PPP header plus 20 bytes for the IPv4 header and 20 bytes for the TCP header. Thus, the total overhead in bits is 8 x (8 + 20 + 20) = 384 bits. Thus,

Time to send 8×10^{16} bits = (time for 1 packet)(# of packets needed)

$$= \left(\frac{n_f}{R}\right)\left(\frac{8\times10^6}{n_f - n_o}\right)$$

$$= \left(\frac{8\times 500 bits}{56\times10^3 bps}\right)\left[\frac{10^6 bytes}{(500-70) bytes}\right]$$

$$= 0.7143\frac{\text{sec}}{packet}\times 2326\, packets$$

= 166.14 seconds

58. Suppose that packets arrive from various sources to a statistical multiplexer that transmits the packets over a 64 kbps PPP link. Suppose that the PPP frames have lengths that follow an exponential distribution with mean 1000 bytes and that the multiplexer can hold up to 100 packets at a time. Plot the average packet delay as a function of the packet arrival rate.

Solution:

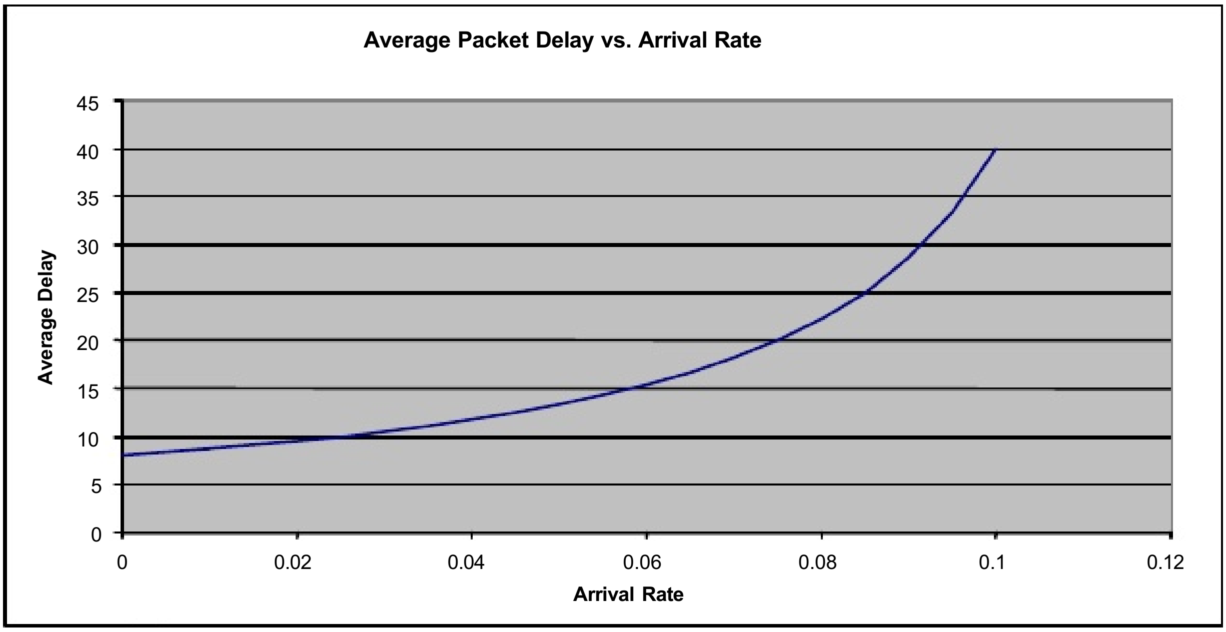
This is an M/M/1/K system in which:

Service rate μ = 6400 bps / 8000 bits = 0.125 packets/sec
Packet length $E[L]$ = 8000 bits
Buffer size K = 100 packets

Thus:

$$E[T] = \frac{E[N]}{\lambda(1-P_L)}$$

where $P_L = \dfrac{(1-\frac{\lambda}{\mu})(\frac{\lambda}{\mu})^K}{1-(\frac{\lambda}{\mu})^{K+1}}$ and $E[N] = \dfrac{\lambda}{\mu-\lambda} - \dfrac{(K+1)\lambda^{K+1}}{\mu^{K+1}-\lambda^{K+1}}$



59. Suppose that the traffic that is directed to a statistical multiplexer is controlled so that ρ is always less than 80%. Suppose that packet arrivals are modeled by a Poisson process and that packet lengths are modeled by an exponential distribution. Find the minimum number of packet buffers required to attain a packet loss probability of 10^{-3} or less.

Solution:

$$P_{loss} = \frac{(1-\rho)\,\rho^k}{1-\rho^{k+1}}$$

with $k = 24$ $P_{loss} = 0.000948$

60. Suppose that packets arrive from various sources to a statistical multiplexer that transmits the packets over a 1 Mbps PPP link. Suppose that the PPP frames have constant length of *L* bytes and that the multiplexer can hold a very large number of packets at a time. Assume that each PPP frame contains a PPP, IP, and TCP header in addition to the user data. Plot the average packet delay as a function of the rate at which user information is transmitted for *L* = 250 bytes, 500 bytes, and 1000 bytes.

Solution:

Assuming that the overhead in each packet is 48 bytes (as described in Problem 57), we have:

$R = 10^6$ bps
L = constant (250, 500, and 1000 bytes)
$K \rightarrow \infty$
n_o = 48 bytes
$\mu = R/L$ packets/sec

Because of the constant length packets, this is an M/D/1 system.

Average delay, $E[T_D] = 1 + \frac{\lambda}{2(\mu - \lambda)} \frac{1}{\mu}$

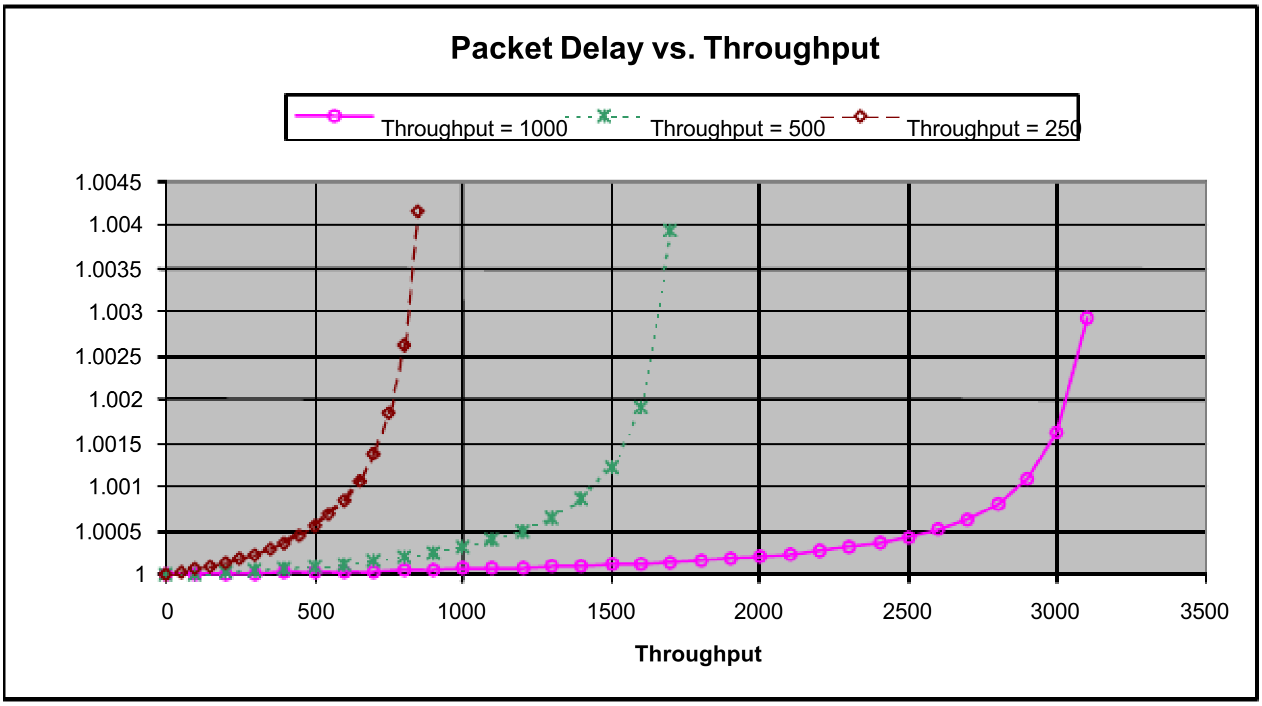
Let γ be the rate in packets per second at which user information transferred.

$$\gamma = \rho \frac{R}{L}\frac{(L-n_o)}{L} = \frac{\lambda}{\mu}\frac{R}{L}\frac{(L-48)}{L} = \lambda\frac{(L-48)}{L}$$

Thus,

$\lambda = \gamma L / (L-48)$

$$E[T_D] = 1 + \frac{\frac{\gamma L}{L-48}}{2(\frac{R}{L} - \frac{\gamma L}{L-48})}\frac{L}{R} = 1 + \frac{\gamma L^2}{2R[R(\frac{L-48}{L}) - \gamma L]} = 1 + \frac{\gamma L^3}{2R[R(L-48) - \gamma L^2]}$$



61. Suppose that a multiplexer receives constant-length packet from $N = 60$ data sources. Each data source has a probability $p = 0.1$ of having a packet in a given T-second period. Suppose that the multiplexer has one line in which it can transmit eight packets every T seconds. It also has a second line where it directs any packets that cannot be transmitted in the first line in a T-second period. Find the average number of packets that are transmitted on the first line and the average number of packets that are transmitted in the second line.

Solution:

The probability that there are k packet arrivals in a T-second period is given by the binomial distribution with parameters $N = 60$ and $p = 0.1$. The average number of arrivals is $Np = 6$. The average number of arrivals that get transferred to the first line is given by:

$$\sum_{k=0}^{8} k \binom{60}{k} (0.1)^k (0.9)^{60-k} = 4.59$$

The remainder of the packet arrivals are sent to the second line, so the average number sent to line 2 is 6 – 4.59 = 1.41 packets per *T*-second period.

62. Discuss the importance of queueing delays in multiplexers that operate at bit rates of 1 Gbps or higher.

Solution:

By aggregating traffic, better delay performance is achieved, as illustrated in section 5.5. In multiplexers that operate at bit rates of 1 Gbps or higher, the queuing delay experienced by packets is much shorter than it would be if many lower bit rate systems were used.