Solutions to Chapter 5

Version 1.01 (list of changes on last page)

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 service, reliable delivery can be achieved through the use of ACK and NAK control messages. 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. For example, HDLC provides for unnumbered acknowledgment service for connection setup and release.

Unacknowledged networks provide simpler and faster communication for networks that are inherently reliable or provide service to higher layers that can tolerate information loss or have built-in error recovery mechanisms.

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 links or networks that are prone to error, loss, and or resequencing. In a connection-oriented service, a setup phase between the sending user and receiving user establishes a context for the transfer of information. In a connection-oriented acknowledged service acknowledgments are provided to the sending user for all SDUs.

In a connectionless service, there is no prior context provided for the transfer of information between the sending user and the receiving user. The sender passes its SDU to the underlying layer without prior notice. In an acknowledged connectionless service, the sending user requires an acknowledgment of delivery of its SDU.

The protocols that provide these services are very different. Connection-oriented acknowledged service requires the use of stateful protocols that keep track of sequence numbers, acknowledgments, and timers. Connectionless services use much simpler protocols that are stateless in nature. Connectionless acknowledged service does require that the transmitting protocol track the acknowledgment of a PDU. In the simplest instance, the receiver would be required to send an ACK for correctly received PDU and the transmitter would keep a timer. If an ACK was not received in time, the transmitter would inform the user of a failure to deliver.

3. Suppose that the two end systems α and β in Figure 5.6 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:

a. 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 $P_{error} = [(1 - p)^3]^{10} = (1 - p)^{30} \approx e^{-30p}$.

b. 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 = $1/ P_{error} = e^{30p}$.

c. 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 payload 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. This stream ID may be avoided if the source and destination operate so that they handle the transfer of a single message at a time. For example, this approach is used by AAL5 in ATM.

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. The best possible efficiency in line usage occurs when every PDU is of maximum size. If the byte stream arrives at a steady rate, full PDUs can be constructed by waiting until a sufficient number of bytes to fill a PDU payload have arrived.

The control information should include sequence numbering, payload size, and error checking. If reliability is required, the control information should also allow for ACK control messages.

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?

If the bytes stream arrives sporadically, arriving bytes can be buffered while the transmitter waits to fill a PDU. A timer can be used to place a bound on the delay incurred waiting.

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:

The protocol that provides the service must be able to multiplex information from different user pairs for transfer. A basic design decision is whether the PDUs should carry information from multiple users or be constrained to carry information from a single user. The latter approach is simpler in that only a single user needs to be identified per PDU. The latter approach is complicated by the need to identify several users (actually multiplexing IDs) per PDU.

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

If the bytes arrive at a constant rate, it may be possible to arrange a PDU structure that transfers the merged byte stream without extensive multiplexing ID information. For example, the position of a byte in the payload could identify which user it belongs to. Alternatively, bytes for each user can be buffered until they can fill a minimum sized payload. Note that the first approach will involve less delay than the second approach.

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

Mixing bytes from the different streams poses a greater challenge in this case. The bytes from different streams need to be buffered separately until they can fill a minimum sized PDU in the approach where each PDU carries information from a single user. Alternatively, the bytes from a stream could fill a minimum-size sub-payload that could then be carried with sub-payloads from other users on shared PDUs. The choice of minimum size payload will tradeoff efficiency against delay.

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 voice, samples are delayed while waiting to fill a packet. A portion of the 20 ms delay must be apportioned to the packetization delay, which in turn determines the packet payload size. 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 the voice samples need to be played back in the same sequence that they were 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. Addressing the form of multiplexing ID may be required if multiplexing is involved.

Privacy, integrity, and authentication have traditionally not been as important as the other issues discussed above. However there will surely be applications were it is essential that the voice information be kept confidential (privacy), that the information be tamper free (integrity), and that imposters be detected and deterred (authentication.)

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

Sequencing and timing are the most important requirements for real-time voice. These requirements are better met in a hop-by-hop approach than in an end-to-end approach because delay performance is critical. Resequencing on an end-to-end basis may lead to excessive delay. Hop-by-hop controls on transfer delay may be critical to achieving real-time transfer.

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.

The key difference from the case of a single voice call (considered in Problem 7) is the multiplexing aspect of handling of multiple voice streams. The considerations in multiplexing multiple byte streams of Problem 6 apply. The separate voice streams could be carried in separate PDUs or they could be combined in the payload of the PDUs. The first approach is simpler but involves greater delay. The second approach can lead not only to lower delays but also to higher efficiency because larger payloads are possible using multiple voice streams.

Mobility is another aspect that needs to be considered. As users move from cell to cell, the network must track their movement and perform handoffs as cell boundaries are traversed. Changes in addresses and routing are involved in these handoffs.

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

The set of requirements are essentially the same in both directions. However, the direction from the users upstream into the network is more difficult in general because of the need to deal with multiple incoming streams.

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. Alternatively, fixed small message sizes may be specified, as in Short Message Service (SMS). Pacing and flow control may now be needed depending on the nature of the messages and applications. 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.

Streaming video involves the delivery of video information across a network. The term "streaming" usually implies a non-real-time situation where significant delay can be tolerated. The following table summarizes the results of part (a).

Adaptation function	Required	Discussion
Handling of arbitrary message size	No	A stream has no inherent required message size. However video and audio formats do have specific structures, which could be relevant to how transfer is carried out.
Reliability and sequencing	Yes	Reliability is important for video & audio quality. Sequencing is important but can be relaxed as buffering delay is increased.
Pacing and flow control	Maybe	See note below
Timing	Yes	Video has strict timing jitter requirements. Synchronization of different media is also important, but not an issue when the media are combined before packetization.
Addressing	Maybe	Required for point-to-point or point-to multipoint transfer, but not in broadcast system. Required for different media components.
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	Control messages are short so this is not an issue.
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, but a generalized control protocol for multiple devices is possible.
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 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.

End-to-end approaches can also provide constant transfer delay, for example, by attaching timestamps to packets obtained from a common clock, e.g. Global Positioning System, and buffering information at the receiving end until a target delivery time.

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 the only difference is that frames are retransmitted sooner than otherwise. The detection of errors in an arriving frame at the receiver will cause an ACK to be sent sooner, possibly causing the transmitter to retransmit sooner. The detection of errors in an arriving frame at the transmitter will cause an immediate retransmission of the current information frame.

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 a speedup in the error recovery process.

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. To setup a connection in a data link, the initiating data link entity sends a SETUP frame, such as SABM in Figure 5.47. 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 5.10 and Figure 5.11 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 5.10. 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. In the absence of connection indexing, the link should only handle one SETUP at a time until the SETUP is confirmed through an acknowledgment. Otherwise ambiguities in terms of what frame an ACK corresponds to will arise.

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 frames 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 first delayed ACK frame arrives after the initiating station sends the DISC frame and the second delayed ACK frame arrives after it sends the SETUP frame, the initiating station will assume the connection is established. Meanwhile the receiver receives the DISC frame and disconnects the connection and because the last SETUP frame is lost it remains in this state.

To resolve this problem the initiating station should not send out SETUP messages with DISC messages outstanding.

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] $= (1-p)^{n/N}$

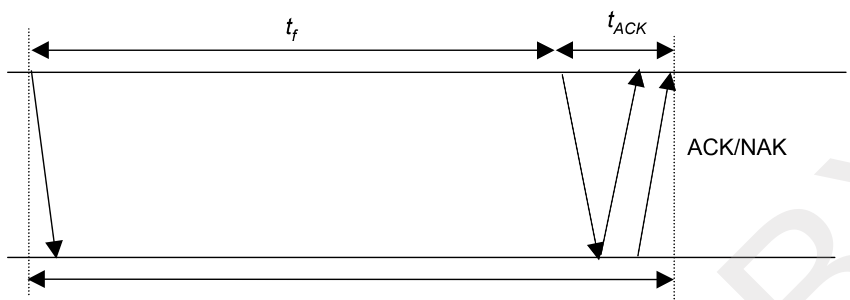
A block has no errors if all subblocks have no errors, so

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_{timeout}
- 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_{total} | 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[total] = $n/R\,(1 – P_f) = n/[Re^{–np}]$ = 8 / (3.35 x $10^{–4}$) = 23,847 seconds = 6.62 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 $n_a << n/N$: $t_0^{\,b} = n_f/R = n/NR$

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

$T = E[t_{\text{total}}] = N\, T_b = n(1 - p)^{-n/N} / R = 8\,(1 - p)^{-n/N} = 8\, e^{np/N}$ 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 5.12. 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. Each unsuccessful trial requires a timeout for retransmission. We assume that the timeout time is set to be equal to frame transmission time plus ACK transmission time.
- Y is the random variable that represents the number of trials before a successful transmission of an ACK. An ACK error will require a new successful retransmission of the frame for next ACK transmission. An ACK is not sent until a new retransmitted frame arrives at the receiver.
- X and Y follow a geometric random-variable distribution.

$T1$ = Average time to go from (0,0) to (0,1) = $(T_f + T_a)\, E(X) + T_f$

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

$T2$ = Average time to go from (0,1) to (1,1) = $T_1 E(Y) + T_a$

$$T2 = T_1\left[\frac{P_a}{1-P_a}\right] + T_a = \left[\frac{2+P_f}{1-P_f}\right]\left[\frac{P_a}{1-P_a}\right] + 1$$

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

P_f	P_a	$T1$	$T2$	Throughput
0.2	0.1	2.75	1.3055	0.4931
0.02	0.01	2.06	1.0208	0.6492
0.002	0.001	2.006	1.0020	0.6649
0	0	2.0000	1.0000	0.6667

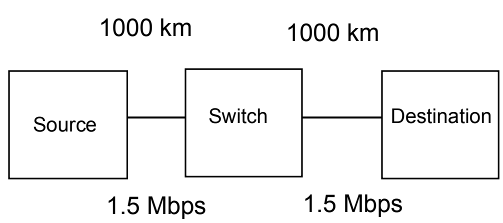
17. Write a program for the transmitter and the receiver implementing Stop-and-Wait ARQ over a data link that can introduce errors in transmission. Assume station A has an unlimited supply of frames to send to station B. Only ACK frames are sent from station B to station A. Hint: Identify each event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

Solution:

Stop and Wait Transmitter				
Current State	**Event**	**Sequence Number**	**Action**	**Next State**
Ready	Request from upper layer		Prepare frame and send; start timer	Wait
Wait	Arrival of error-free ACK frame	Correct sequence number $R_{next} = S_{last} + 1$	Increment Send Sequence Number $S_{last} = R_{next}$	Ready
Wait	Request from upper layer		New requests are blocked	Wait
Wait	Arrival of error-free ACK frame	Incorrect sequence number	Discard frame	Wait
Wait	Arrival of erroneous frame		Discard frame	Wait
Wait	Timeout Expires		Resend frame; start timer	Wait

Stop and Wait Receiver				
Current State	**Event**	**Sequence Number**	**Action**	**Next State**
Ready	Arrival of error-free frame	Expected Sequence Number $S_{last} = R_{next}$	Accept frame, increment R_{next}, $R_{next} = R_{next} + 1$, and send ACK, deliver packet to higher layer	Ready
Ready	Arrival of error-free ACK frame	Incorrect sequence number	Discard frame; send ACK with R_{next}	Ready
Ready	Arrival of erroneous frame		Discard frame	Ready

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 10^5 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 error, the ACK time, and processing time are negligible.

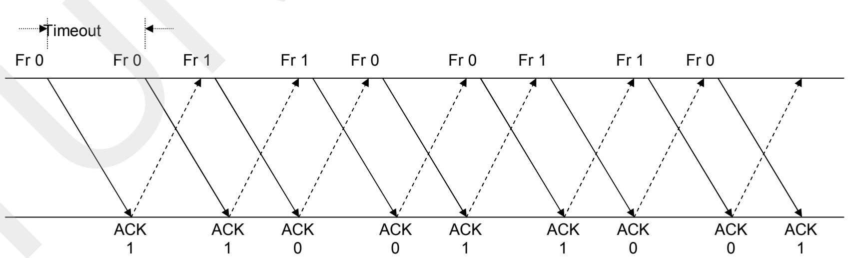
T_{prop} = distance / propagation speed = 0.0050 s
T_f = packet size / bandwidth = 0.0109 s
T_0 = $T_{prop} + T_f$ = 0.0159 s
P_f = probability of error in packet = 0.016251

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

There is pipelining effect that occurs as follows: After the first packet arrives at switch 1, two transmissions take place in parallel. The first packet undergoes stop-and-wait on the second link while the second packet undergoes stop-and-wait in the first link. The packet arriving at the switch cannot begin transmission on the next link until the previous packet has been delivered, so there is an interaction between the transmission times of the two packets. We will neglect this effect. The time to send every packet over two links is then the initial packet transmission time + 33 additional packet timess, and so the average time is $E[T_{total}]$ * 34 = 0.522 seconds.

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:

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

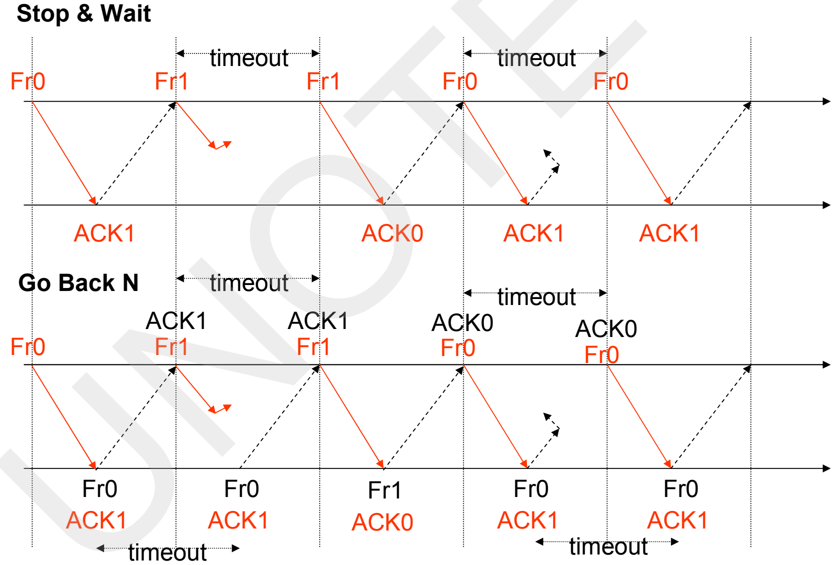
$$\eta_o = \frac{R^0_{eff}}{R} = \frac{\frac{n_f - n_o}{t_o}}{R} = \frac{\frac{8\times512}{4.64\times10^{-4}}}{10\times10^6} = 0.8828 = 88.3\%$$

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:

The figure below shows that the bidirectional Go-Back-N ARQ recovers from errors in the same time that Stop-and-Wait:



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 (Megabits)

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

Time for the sequence number to wrap around (sec)
Frame Time * 2^{32}=4294967296*8*1000/R seconds

Bit Rate Mbps	Round Trip Time (msec)		
	100	10	1
1	34359738.37	34359738.37	34359738.37
10	3435973.84	3435973.84	3435973.84
100	343597.38	343597.38	343597.38
1000	34359.74	34359.74	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?

Time for the sequence number to wrap around (sec)
Byte Time * 2^{32}=4294967296*8/R seconds

Bit Rate Mbps	Round Trip Time (msec)		
	100	10	1
1	34359.74	34359.74	34359.74
10	3435.97	3435.97	3435.97
100	343.60	343.60	343.60
1000	34.36	34.36	34.36

The sequence number wraps around in much shorter time. At 1 Gbps the sequence number wraps around in only 34.36 seconds.

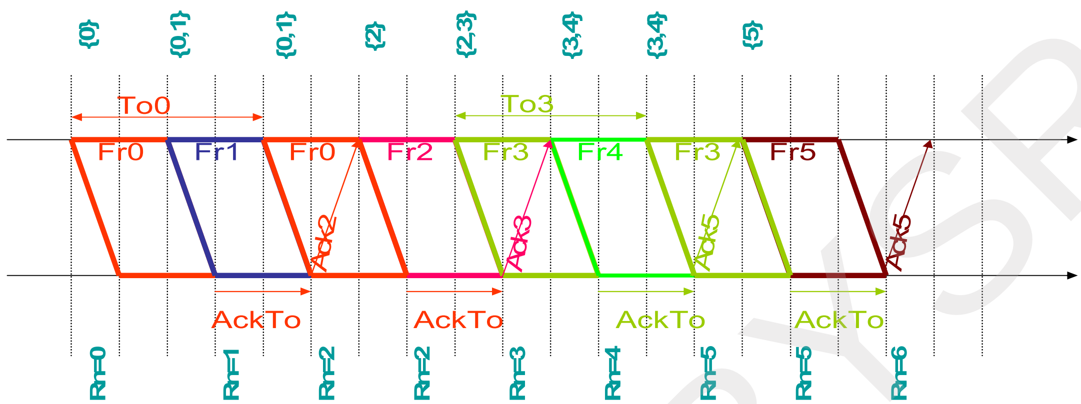
23. Consider a bidirectional link that uses Go-Back-N with $N = 7$. Suppose that all frames are one unit long and that they use a time-out value of 2. Assume the propagation is 0.5 unit and the processing time is negligible. Assume the ACK timer is one unit long. Assuming stations A and B begin with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

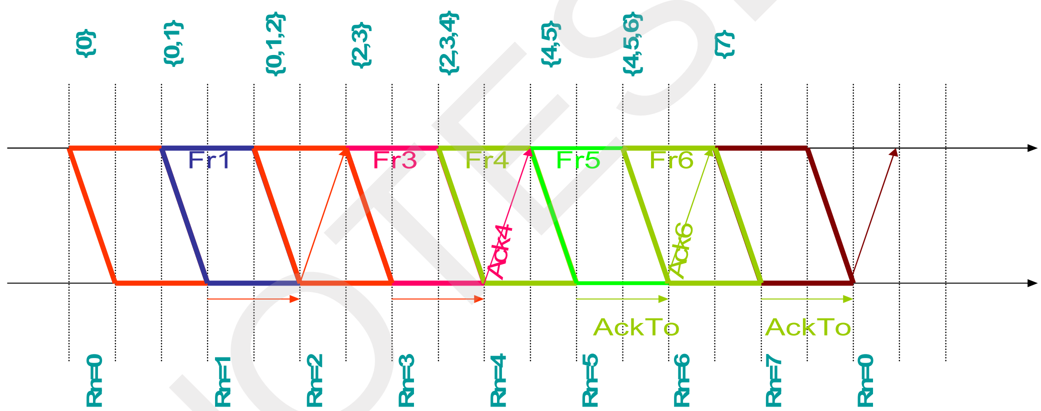
Solutions follow questions:

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. The ACK timer delays the sending of an ACK to provide an opportunity for piggybacking.

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

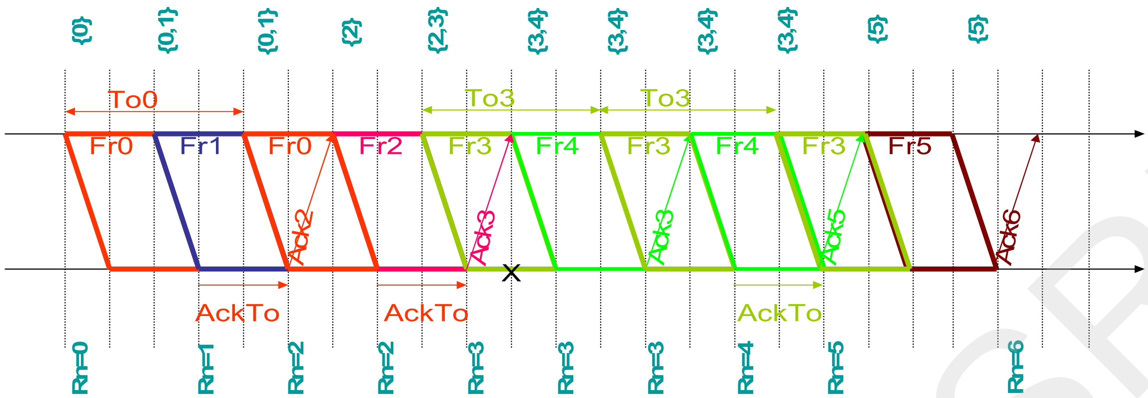
The first figure below shows that the timeout value is too short and cause needless retransmissions. The second figure below shows that by increasing the timeout value to 3 units, transmissions occur smoothly and efficiently.

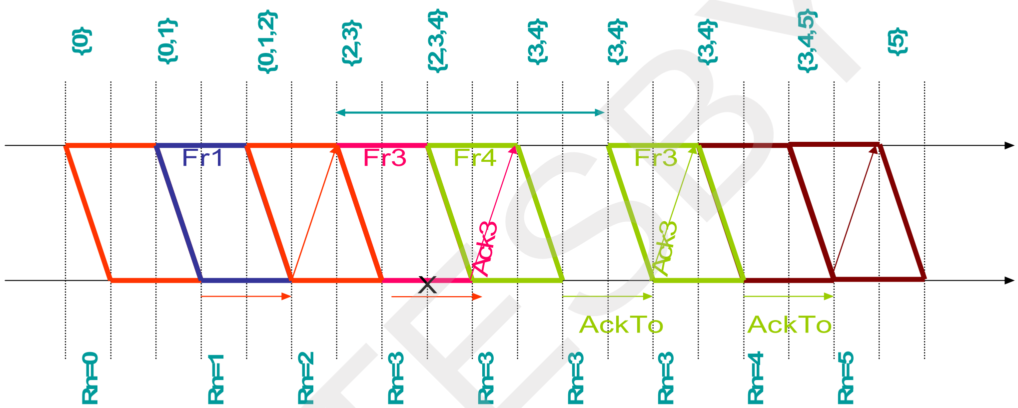




b. Station A sends six frame in a row, starting at $t = 0$. All frames are received correctly, except frame 3 is lost.

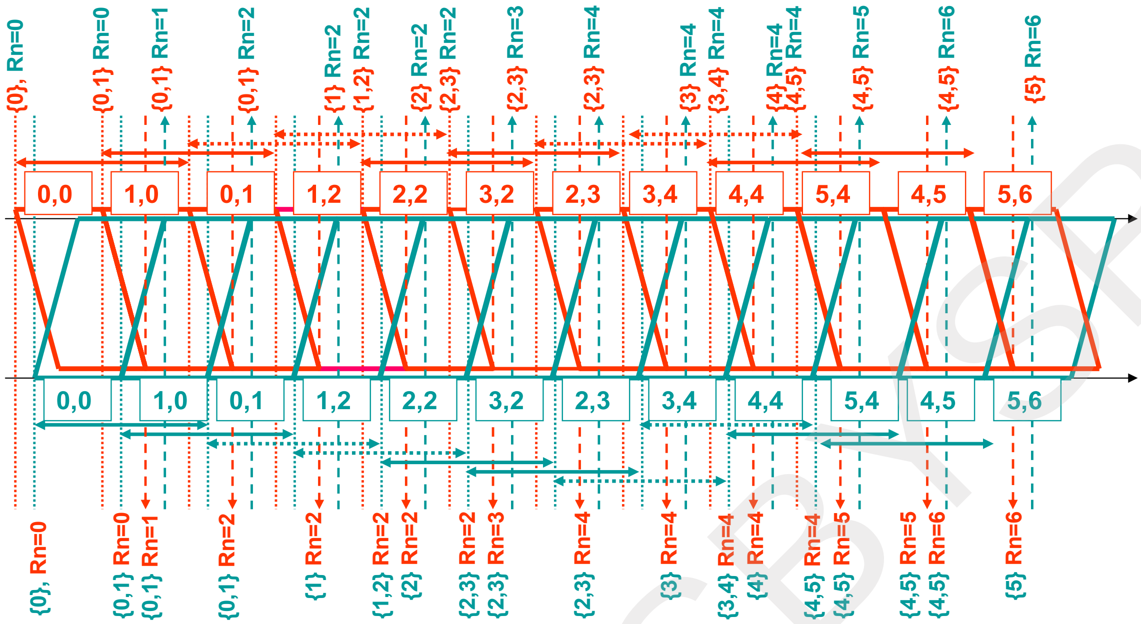
The following two figures show the sequence of frame exchanges with timeout values of 2 and 3 units. The system with the shorter timeout value initiates the recovery from error sooner, but cannot advance the window fast enough due to the short timeout value.





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.

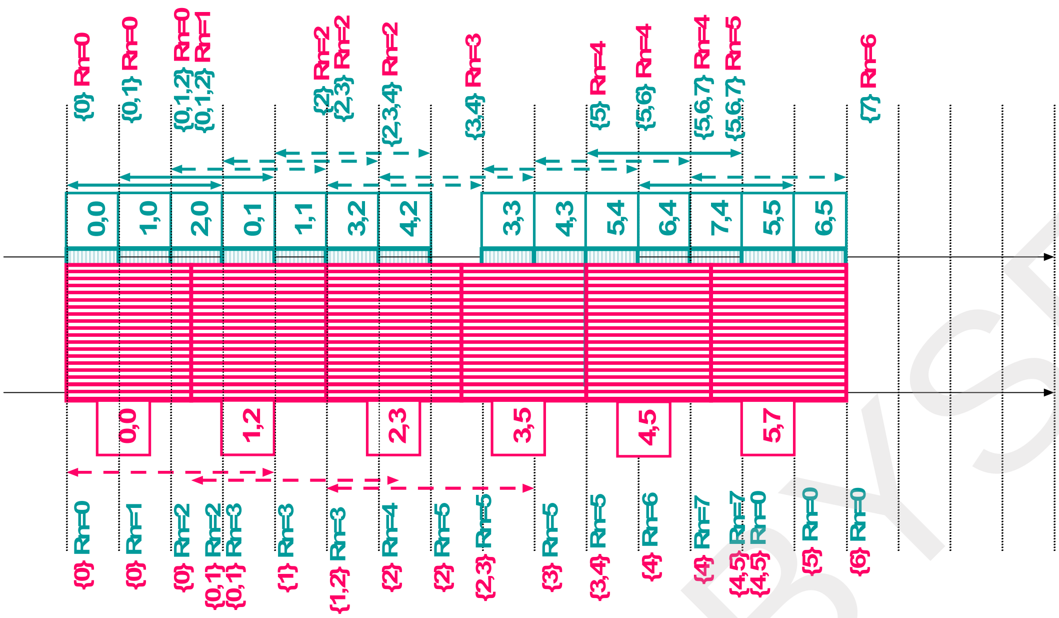
This problem shows that when piggybacking is used the timeout has too be somewhat longer. In the following figure we see that timeouts occur repeatedly causing needless retransmission of frames that have already arrived at the receiver. A timeout value of 4 is required to assure a smooth sliding forward of the transmitter's send window.

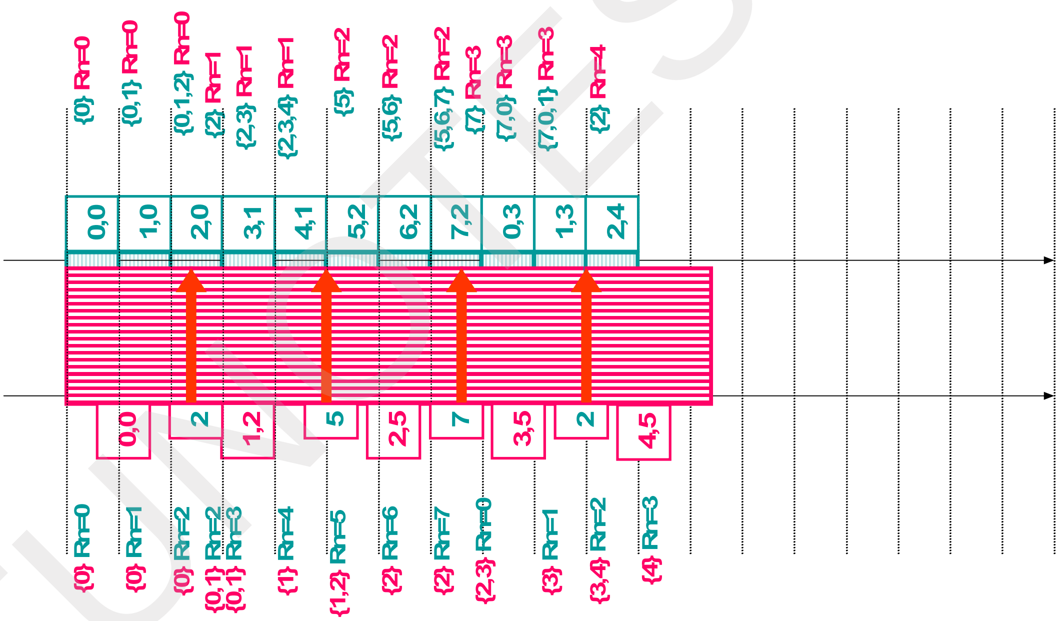


24. Consider a bidirectional link that uses Go-Back-N with $N = 3$. Suppose that frames from station A to station B are one unit long and that they use a time-out value of 2, and that frames in the opposite directions are 2.5 units long and that they use a time-out value of 4. Assume propagation and 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 shows the sequence of frames exchanged by stations A and B, as well as the set of unacknowledged frames, and the next expected frame at each station. Note that acknowledgments from station A to station B are frequent because of the short frame length and so station B sends frames in a continuous efficient manner. This is not the case for station A which receives acknowledgments after large delay and so frequently spends its time carrying out needless retransmissions. The solution to this problem is to insert ACK messages from station B to station A in between the long message transmissions. The second figure below shows that transmissions from A to B then flow smoothly.





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, use piggybacking only if frame already available)

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 similarly to a Selective Repeat process.

Vs. Stop-And-Wait. If Stop-and-Wait is used, the effective bit rate, 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?

The parallel Stop-and-Wait protocol does not deliver frames in order, unless augmented by a frame resequencing scheme.

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) at the transmitter
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.

Go Back N Transmitter			
Event	**Condition**	**Action**	**Next State**
Request from upper layer	Not the last sequence number in nonempty send window	Prepare frame with S_{recent} and send, increment send sequence number; start timer	Ready
Request from upper layer	Last sequence number in nonempty send window	Prepare frame and send, increment send sequence number; start timer	Blocking
Arrival of error-free ACK frame	Correct sequence number, R_{next} is between S_{last} and S_{recent}	Slide window forward, $S_{last} = R_{next}$, max Sequence number = $S_{last} + W_S - 1$	Ready
Arrival of error-free ACK frame	Sequence number R_{next} is NOT between S_{last} and S_{recent}	Discard frame	Ready
Timeout Expires		Resend all frames from S_{last} onwards, reset timer	Ready
Arrival of erroneous frame		Discard frame	Ready
Request from upper layer		Requests are blocked	Blocking
Arrival of error-free ACK frame	Correct sequence number, R_{next} is between S_{last} and $S_{last} + W_S - 1$	Slide window forward, $S_{last} = R_{next}$, max Sequence number = $S_{last} + W_S - 1$	Ready
Arrival of error-free ACK frame	Sequence number R_{next} is NOT between S_{last} and S_{recent}	Discard frame	Blocking
Timeout Expires		Resend all frames from S_{last} onwards, reset timer	Blocking
Arrival of erroneous frame		Discard frame	Blocking

Go Back N Receiver				
Current State	**Event**	**Sequence Number**	**Action**	**Next State**
Ready	Arrival of error-free frame	Expected sequence number $S_{last} = R_{next}$	Accept frame, increment R_{next}, $R_{next} = R_{next} + 1$, and send ACK, deliver packet to higher layer	Ready
Ready	Arrival of error-free frame	Incorrect sequence number	Discard frame; send ACK with R_{next}	Ready
Ready	Arrival of erroneous frame		Discard frame	Ready

28. Modify the program in problem 5.27 to implement Selective Repeat ARQ.

Solution:

Selective Repeat Transmitter

Current State	Event	Sequence Number	Action	Next State
Ready	Request from upper layer	Not the last sequence number in nonempty send window	Prepare frame with S_{recent} and send, increment send sequence number; start timer	Ready
Ready	Request from upper layer	Last sequence number in nonempty send window	Prepare frame and send, increment send sequence number; start timer	Blocking
Ready	Arrival of error-free ACK frame	Correct sequence number, R_{next} is between S_{last} and S_{recent}	Slide window forward, $S_{last} = R_{next}$, max sequence number = $S_{last} + W_S - 1$	Ready
Ready	Arrival of error-free NAK frame	Correct sequence number, R_{next} is between S_{last} and S_{recent}	Slide window forward, $S_{last}=R_{next}$, max Sequence number = $S_{last} + W_S - 1$, retransmit frame with R_{next}	Ready
Ready	Arrival of error-free ACK/NAK frame	Sequence number R_{next} is NOT between S_{last} and S_{recent}	Discard frame	Ready
Ready	Timeout Expires		Resend frame corresponding to timer, reset associated timer	Ready
Ready	Arrival of erroneous frame		Discard frame	Ready
Blocking	Request from upper layer		Requests are blocked	Blocking
Blocking	Arrival of error-free ACK frame	Correct sequence number, R_{next} is between S_{last} and $S_{last} + W_S - 1$	Slide window forward, $S_{last}=R_{next}$, max sequence number = $S_{last} + W_S - 1$	Ready
Blocking	Arrival of error-free NAK frame	Correct sequence number, R_{next} is between S_{last} and S_{recent}	Slide window forward, $S_{last}=R_{next}$, max sequence number = $S_{last} + W_S - 1$, retransmit frame with R_{next}	Ready
Blocking	Arrival of error-free NAK frame	Correct sequence number, $R_{next} = S_{last}$	retransmit frame with R_{next}	Blocking
Blocking	Arrival of error-free ACK/NAK frame	Sequence number R_{next} is NOT between S_{last} and S_{recent}	Discard frame	Blocking
Blocking	Timeout expires		Resend frame corresponding to timer, reset associated timer	Blocking
Blocking	Arrival of erroneous frame		Discard frame	Blocking

Selective Repeat Receiver

Current State	Event	Sequence Number	Action	Next State
Ready	Arrival of error-free frame	Sequence number is R_{next}	Accept frame; slide window forward $R_{next} = R_{next} + k$, and send ACK, deliver k packets to higher layer	Ready
Ready	Arrival of error-free frame	Sequence number is in the receive window, that is, between $R_{next}+1$ and $R_{next} + W_R - 1$	Accept and buffer frame; and send ACK	Ready
Ready	Arrival of error-free frame	Sequence number is outside the receive window,	Discard frame, send ACK with R_{next}	Ready
Ready	Arrival of erroneous frame		Discard frame	Ready

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. Indeed, the echo packet constitutes an acknowledgment.

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.

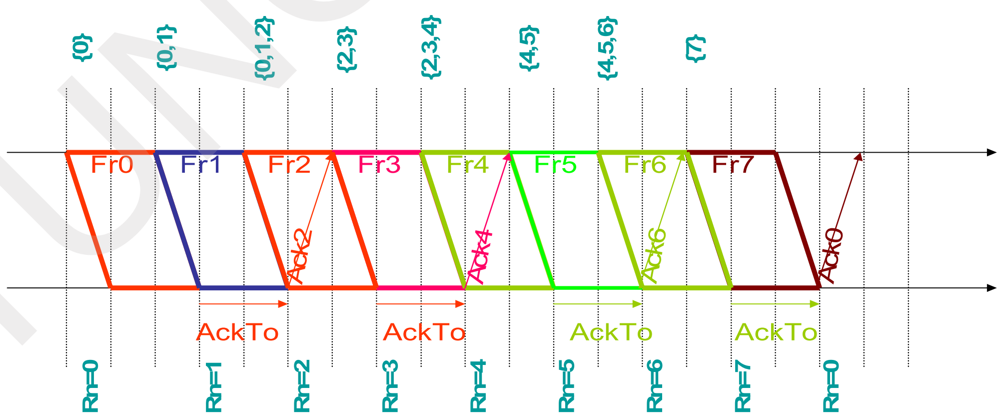
The upstream traffic *to* a server is generally much less than the downstream traffic. Thus, the piggybacking method is non-ideal in this case. If the channel has a low probability of error, the alternating ACK method is better, as it saves bandwidth. However, if the connection causes frequent errors, 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:

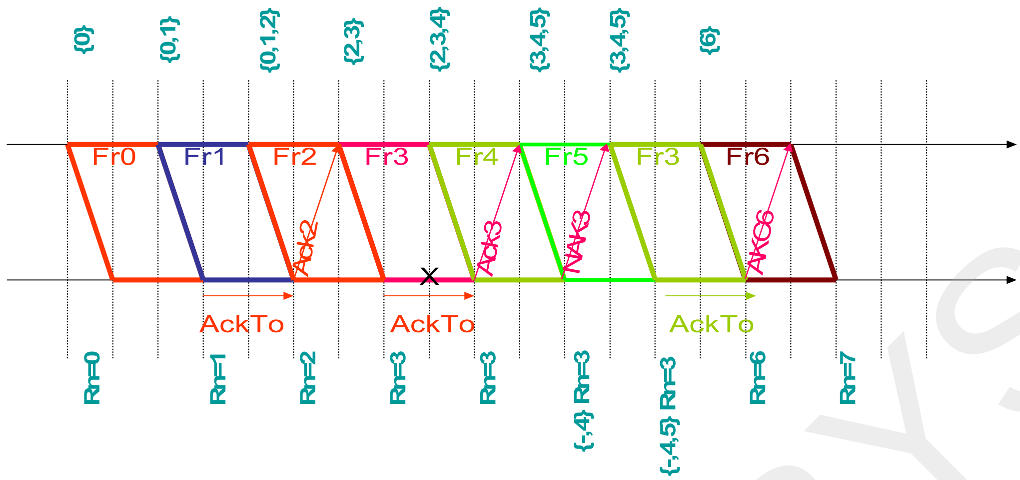
Solutions follow questions:

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

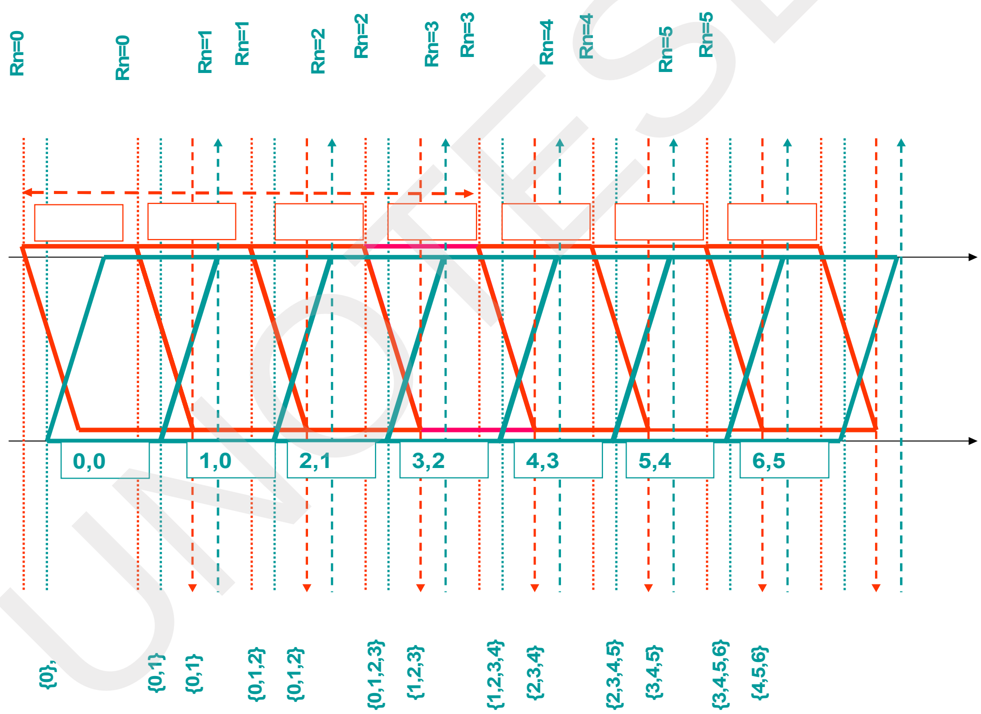
A timeout value of 2 causes the transmitter to resend frames before the window size of 4 is exhausted. We will assume a timeout value of 4 in the following solutions.



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.



31. In the chapter we showed that if the transmit and receive maximum window sizes are both equal to half 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 transmissions 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 $R_{next} + W_R - 1$ 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. Second, the transmitter operation must change to retransmit frames according to the received list. If the list contains the m oldest frames that are yet to be received, then the list can be used to skip retransmissions of frames that have already been received.

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

The performance will increase in cases with high error rate 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:

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}.

First we have the following:

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

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

Using the results in Equation 5.4,

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

$= 0.125$ ($n_f = 2048$)
$= 0.177$ ($n_f = 4096$)

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 $W_S = 2^3 - 1 = 7$, we can calculate that the window size is:

$$\frac{n_f \times W_S}{R} = 256ms$$

Since this is greater than the round trip propagation delay, we can calculate the efficiency by using the results in Equation 5.8.

$$\eta = (1 - P_f)\frac{1 - \dfrac{n_o}{n_f}}{1 + (W_S - 1)P_f}$$

$= 0.385 \quad (n_f = 2048)$

$= 0.220 \quad (n_f = 4096)$

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:

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

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:

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

$$= 4800 + 2.50 = 4802.5 \sec/image$$

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.

The difficulty here is that if we transmit an entire image as a single block, then the average number of errors in a block is 48 x 10^8 x 10^{-5} = 48000. Thus every transmission fails. Clearly we need to transmit using a smaller frame size, say n. Qualitatively, the following happens as we vary n. If n is very small, then the probability of frame error is small, but the overhead due to the header n_o will become significant. If n becomes too large, then the efficiency drops because of frequent retransmissions. Thus there must be an intermediate value of n that optimizes efficiency. We will find this value by trial and error. We will assume a header overhead of 64 bits in a frame of size n greater than 64 bits.

Each frame carries n bits of information, so the required window size for a propagation delay of 2.5 seconds is then W_s=2.5x10^6/n+1 The probability of frame error is then

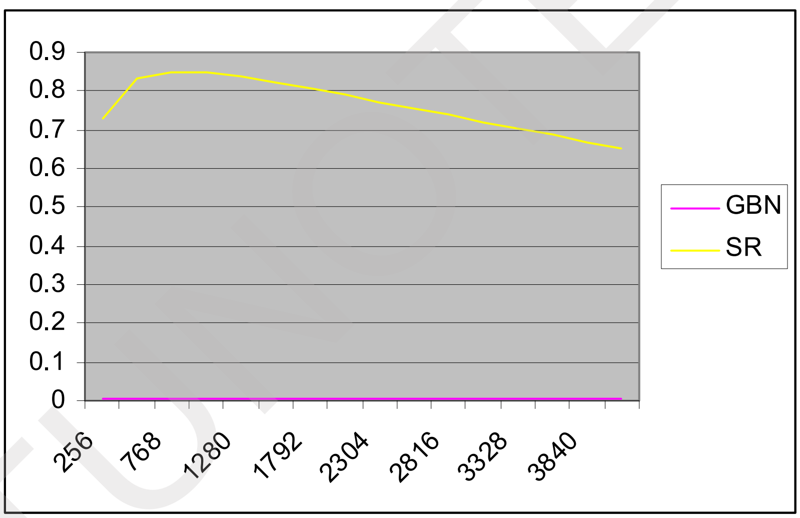
$$P_f = 1-(1-10^{-5})^n \approx 1-e^{-n10^{-5}}$$

The efficiency for Go-Back-N is given by:

$$\eta_{GBN} = \frac{(1-P_f)(1-\frac{64}{n})}{1+(W_s-1)P_f} = \frac{(1-\frac{64}{n})e^{-n10^{-5}}}{1+(\frac{2.5x10^6}{n})(1-e^{n10^{-5}})}$$

Using similar assumptions, the efficiency of Selective Repeat ARQ is given by

$$\eta_{SR} = (1-P_f)(1-\frac{64}{n}) = (1-\frac{64}{n})e^{-n10^{-5}}$$



The figure above shows the efficiency of Go-Back-N and Selective Repeat ARQ as a function of frame size. Go-Back-N has very low efficiency (always below 1%) for all values of n. Selective Repeat achieves a maximum of about 85% efficiency at around *n*=768 bits.

Note that the optimum value of *n* is dependent on the value of n_o.

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.

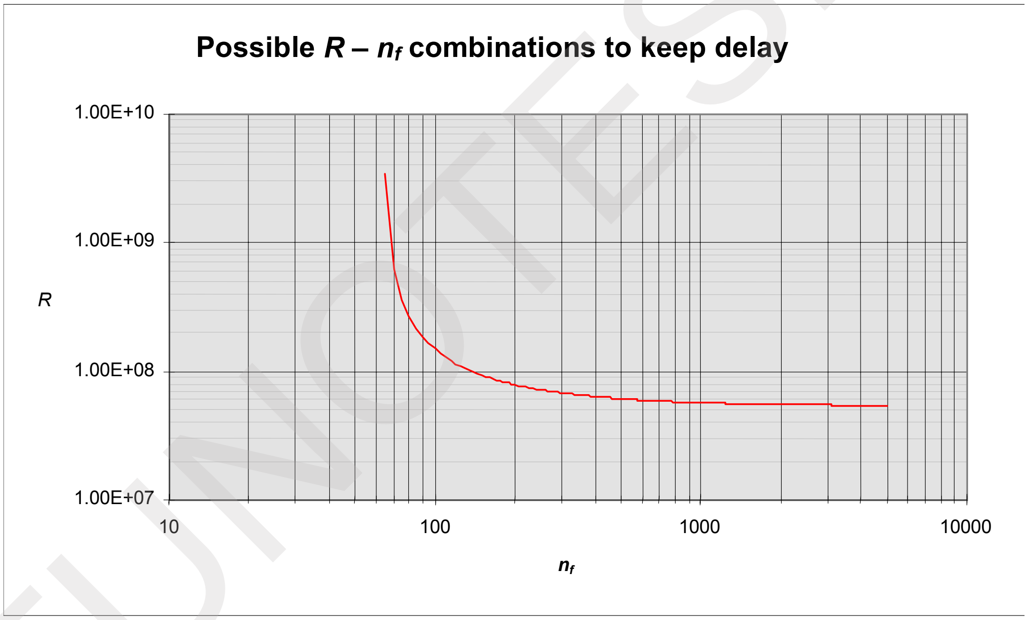
Total delay $= 2t_{prop} + \dfrac{8\times10^6}{R_{eff}^o} = 200\times10^{-3} + \dfrac{8\times10^6}{\eta R} < 250\times10^{-3}$

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

From Equation 5.11 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 n_f that maximizes transmission efficiency for a channel with random bit errors 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 $b = 2\,(t_{proc} + t_{prop})\,R + n_a$

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

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

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

$$n_f = \frac{(n_o - b) \pm \sqrt{(b - n_o)^2 - 4\left[\frac{b + n_o}{\ln a} - bn_o\right]}}{2}$$

b. Go-Back-N ARQ.

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

Let $a = (1 - P_b)$
Let $b = 2R\,(t_{proc} + t_{prop})$ and use the approximation $1 - (1 - p)^n \approx np$, (which is valid when np <<1), 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:

$$0 = \frac{d\eta}{dn_f} = \frac{d}{dn_f}\left[\frac{a^{n_f}(n_f - n_o)}{n_f}\right] = \frac{(a^{n_f}\ln a(n_f - n_0) + a^{n_f})n_f - a^{n_f}(n_f - n_0)}{n_f^{\,2}}$$

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 5.34c:

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

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.

d. Find the optimum length for a 1 Mbps channel with 10 ms reaction time, 25-byte overhead, 25-byte ACK frame, and $p = 10^{-4}$, 10^{-5}, and 10^{-6}.

$R = 1\text{x}10^6$,

$n_o = 8 \text{ x } 25 = 200$,

$n_a = 8 \text{ x } 25 = 200$,

$P_b = 10^{-4}, 10^{-5}, 10^{-6}$

$a = 1 - P_b$, $\ln a = -1\text{x}10^{-3}, -1\text{x}10^{-4}, -1\text{x}10^{-5}$

$t_{prop} + t_{proc} = 10$ ms, $b = 2\,(t_{proc} + t_{prop})\,R + n_a = 2 \text{ x } 10^{-2} \text{ x } 1\text{x}10^6 + 200 = 20200$

P	10^{-4}	10^{-5}	10^{-6}
Stop and Wait ARQ	1156	7552	36303
Go back N ARQ	558	1517	4573
Selective Repeat	558	1517	4573

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.

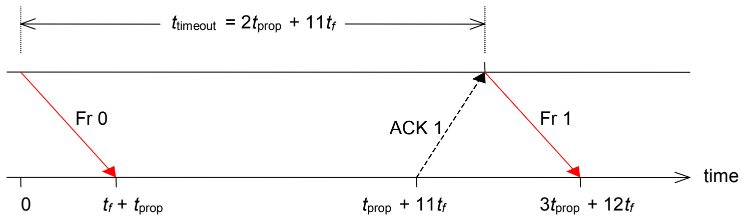
b. One-way propagation delay is 1 ms.

c. One-way propagation delay is 100 ms.

Note: The solution to this problem is currently being reviewed and may be revised.

Solution:

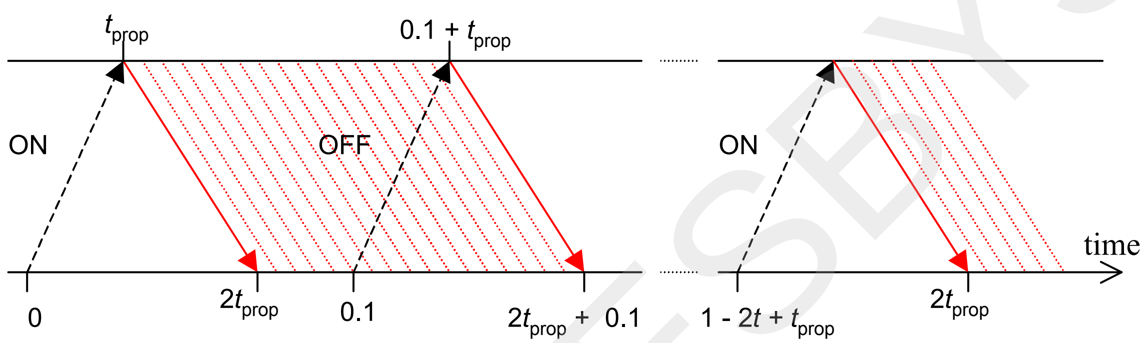
The issue in this problem is that the receiver clears its buffer at 1/10 the rate that the channel can put information into the buffer. Consequently, acknowledgments must be withheld from the transmitter to allow the receiver time to clear the buffer. For the sake of simplicity assume a frame is 1 Mbit long, then it takes 1 Mbit/10 Mbps = 100 ms to transmit the frame, which arrives in its entirety at the receiver at time 100 ms + t_{prop} as shown in the figure below.



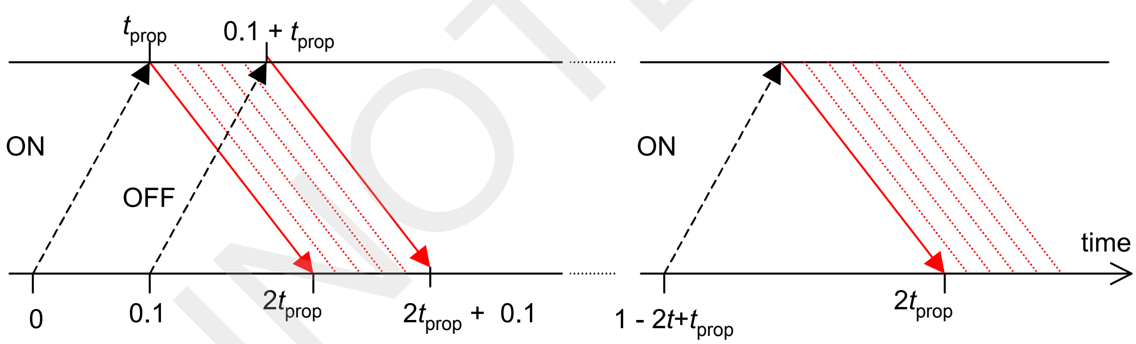
38. Redo problem 5.37 using Xon/Xoff flow control.

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:



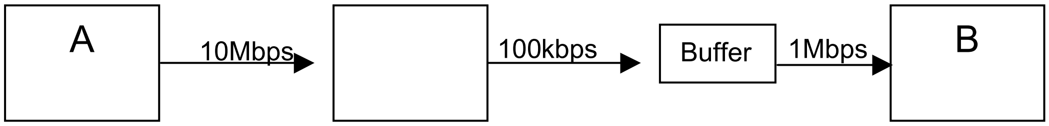
39. Suppose station A sends information to station B over a two-hop path. The data link in the first hop operates at a speed of 10 Mbps, and the data link in the second hop operates at a speed of 100 kbps. Station B has a 1-Megabit buffer to receive information from A, and the application at station 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.

Note: The solution to this problem is currently being reviewed and may be revised.

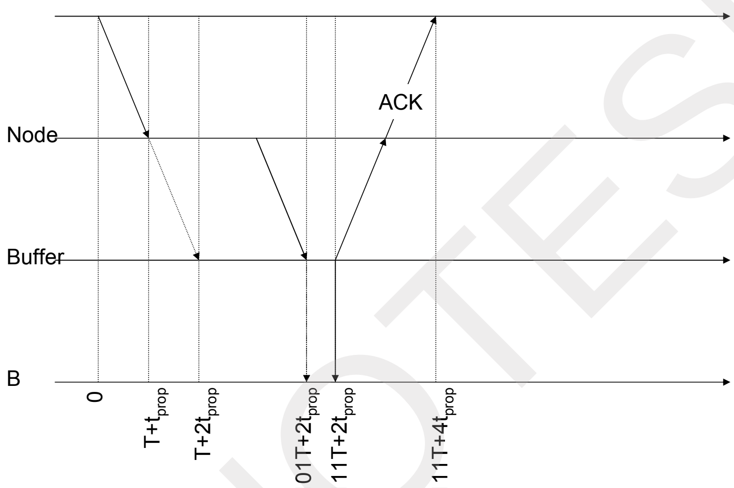
a. One-way propagation delay in data link 1 and in data link 2 are 1 ms.

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

Solution:



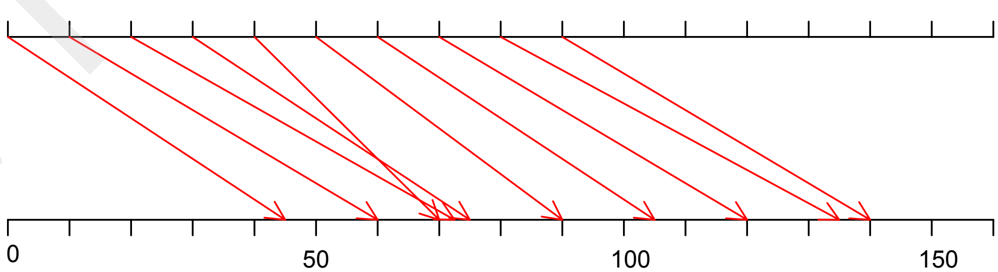
Assume that a frame is n_f bits long, then the frame transmission time in link 1 is $t_1 = n_f/10^7$ seconds, and $t_2 = n_f/10^5$ seconds on the second link as shown in the figure below. Therefore the time to transmit a frame over the second link is 100 times greater than the time to transmit over the first link. For example if n_f = 10000 bits, then T=t_1 = 1 ms and t_2 =100 ms=100T, and similarly if n_f = 1000000 bits, then t_1 = 100 ms and t_2 =10 seconds. 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 until it has completely arrived, then it will take $t_3 = n_f/10^6$ =10T seconds to read out. In the above examples, we have t_3 = 10 ms and 1 second respectively. The ACK message for Stop-and-Wait can then be sent. Let $t_f=t_1+t_2+t_3$=111T. 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 $t_f + 2\, t_{prop}$; an ACK message is send thereafter, and the message arrives at the transmitter at approximately time t_f+4t_{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:

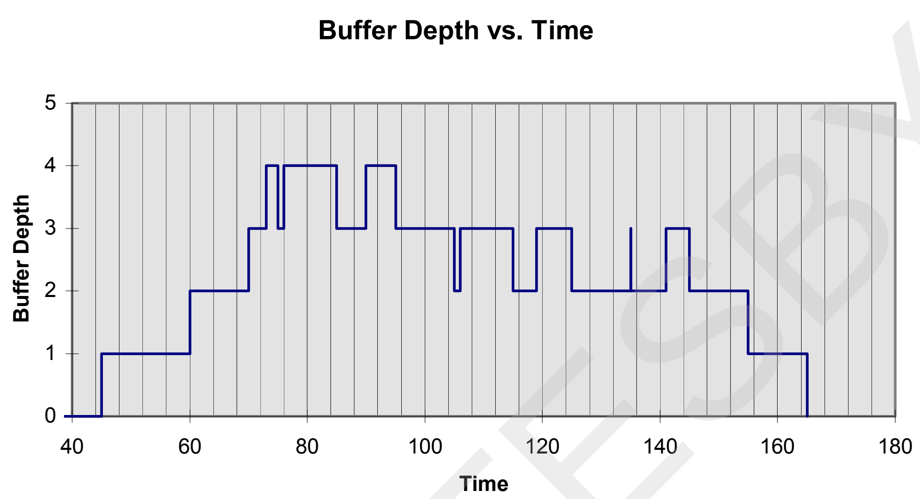
a. Sketch the sequence of packet transmission times and packet arrival times.



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?

There are several components in the delay experienced by the information: (1) packetization delay, the time the first byte in a packet payload waits until the packet is full; (2) network transfer delay, the time to traverse the network; and (3) playout delay, the additional delay inserted at the receiver. The sum of these three components should not exceed the maximum acceptable delay. In particular, the length of the playout buffer cannot be longer than the product of the maximum delay and the bit rate of the information.

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. In addition the playout delay should ensure that enough packets are buffered so at no time will the buffer become empty and cause "starvation" of the playout system.

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 the clock advance at least once between samples, that is, Δ < 10 ms, in order to ensure a unique timestamp for each packet. In addition if timestamps consist of m bits, we do not want the clock to advance so quickly that the timestamp counter wraps around in one sample time, that is, $2^m \times \Delta >> 10$ ms.

The timestamps can be used to resequence packets at the receiver. 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 UDP is used to carry the speech signal in the previous problem. What is the total bit rate consumed by the sequence of UDP PDUs? What is the percent overhead?

Solution:

Number of data bits in each PDU = $64 \times 10^3 \times 10 \times 10^{-3}$ = 640 bits

UDP header size = 8 x 8 bytes = 64 bits

Total bit rate = (640 + 64) / (10×10^{-3}) = 704 x 10^2 bps = 70.4 Kbps

Overhead = [64 / (640 + 64)] x100= 9.1%

44. 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:

The combination of timestamps and sequence numbers help protect a receiver against sequence number wraparound in very high speed networks. If the transmission speed is sufficiently high, it is possible for a transmitter to re-use a sequence number in a relatively short period of time. The timestamp can be used to distinguish between frames (segments) with the same sequence number. Here is how: the sequence numbers can be viewed as being obtained by looking at the time in a wall clock; the timestamp can correspond to the date. The combination of date and clock time allow the receiver to distinguish between a new segment (with a given sequence number) and an old retransmitted segment (with the same sequence number). In order for this to work, the date should advance at least once in the time it takes for the sequence to wrap around, i.e. the date should advance every 24 hours. This combination of timestamps and sequence numbers is called the PAWS algorithm for "protection against wrapped sequence numbers."

45. 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:

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

46. In Figure 5.32, determine the number of bytes exchanged between client and server in each direction.

Solution:

12 bytes from server to client in frame 4, and 3 bytes from client to server in frame 5. All other exchanges involve only control information in the header.

47. Suppose that the delays experienced by segments traversing the network is equally likely to be any value in the interval [50 ms, 75 ms].

Solutions follow questions:

a. Find the mean and standard deviation of the delay.

The delay lies between interval [50ms, 75ms] and is a uniform random variable. The mean is:

$E[X] = (50 + 75) / 2 = 62.5$ ms.

The standard deviation of delay is:

$STD[x] = VAR[x]^{1/2} = [(75 – 50)^2 / 12]^{1/2} = 7.217$

b. Most computer languages have a function for generating uniformly distributed random variables. Use this function in a short program to generate random times in the above interval. Also, calculate t_{RTT} and d_{RTT} and compare to part (a).

See below for sample program written in C.

```
/* Communication Networks - Chapter 5    */
/* Question 47 (b)                       */
/* Description - Generate a random value */
/* between 50 to 75 ms.  Calculate t_RTT */
/* and d_RTT                             */
/* The min, max and avg value of t_RTT   */
/* and d_RTT are also recorded           */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main (void)
{
  int i;
  float temp, t_n, t_rtt_new, t_rtt_old;
  float d_rtt_new, d_rtt_old;
  float t_rtt_min, t_rtt_sum, t_rtt_max;
  float d_rtt_min, d_rtt_sum, d_rtt_max;

  const float alpha = 0.875;
  const float beta = 0.25;

  srand (time(NULL));
  t_rtt_old = 0;
  d_rtt_old = 0;

  t_rtt_sum = t_rtt_max = 0;
  d_rtt_sum = d_rtt_max = 0;

  t_rtt_min = d_rtt_min = 500;

  for (i = 0; i < 500; i++)
  {

    /* Generate a random value between 0 to 1 */
    temp = (float) rand() / RAND_MAX;

    /* Scale the random value to fit between 50 to 75 */
    t_n = temp * 25 + 50;
```

```
    /* Calculate t_RTT and d_RTT */
    t_rtt_new = (alpha * t_rtt_old) + ((1 - alpha) * t_n);
    d_rtt_new = (beta * d_rtt_old) + ((1 - beta) * fabs (t_n - t_rtt_old));

    if (t_rtt_new < t_rtt_min)
      t_rtt_min = t_rtt_new;
    if (t_rtt_new > t_rtt_max)
      t_rtt_max = t_rtt_new;

    if (d_rtt_new < d_rtt_min)
      d_rtt_min = d_rtt_new;
    if (d_rtt_new > d_rtt_max)
      d_rtt_max = d_rtt_new;

    t_rtt_sum += t_rtt_new;
    d_rtt_sum += d_rtt_new;

    printf ("t_RTT: %f d_RTT: %f\n", t_rtt_new, d_rtt_new);
    t_rtt_old = t_rtt_new;
    d_rtt_old = d_rtt_new;
  }
  printf ("t_RTT min: %f t_RTT max: %f t_RTT avg: %f\n",
          t_rtt_min, t_rtt_max, (t_rtt_sum / 500.0));
  printf ("d_RTT min: %f d_RTT max: %f d_RTT avg: %f\n",
          d_rtt_min, d_rtt_max, (d_rtt_sum / 500.0));
}
```

According to the preceding program the average value of t_{RTT} = 61.6924 and d_{RTT} = 7.1139. These values are averaged from a sample of over 500 values.

48. Suppose that the advertised window is 1 Mbyte long. If a sequence number is selected at random from the entire sequence number space, what is the probability that the sequence number falls inside the advertised window?

Solution:

If the sequence number field is 32 bits in length and the advertised window is 1Mbyte long, the probability that the sequence number falls inside the advertised window is:

$P = (1 \times 10^6) / 2^{32} = 2.33 \times 10^{-4}$

49. Explain the relationship between advertised window size, RTT, delay-bandwidth product, and the maximum achievable throughput in TCP.

Solutions follow questions:

a. Plot the maximum achievable throughput versus delay-bandwidth product for an advertised window size of 65,535 bytes.

First consider delay-bandwidth product, $DBP = R*2t_p$. Here delay $2t_p$ is the propagation time that elapses from when a bit is sent by a source to the destination to when the bit is returned back to the source. This is the minimum time that elapses from when a packet leaves a source to when the acknowledgment is received. The delay-bandwidth product *DBP* is then the number of bits (or bytes) that are in the network from when the source transmits continuously at the maximum rate to when the bits return immediately back to the source.

The round-trip time *RTT* is the time that actually elapses from when a packet is sent to when its acknowledgment is received. *RTT* includes not only the propagation delay, but also queueing and processing delays. The advertised window W places a limit on the amount of information that a source can have outstanding in the network.

Consider the time from when a byte leaves the source to when its acknowledgment is received (that is, consider a *RTT*). In that time, the source will have transmitted at most a window-full of bytes into the network. Therefore the window size divided by the *RTT* places a limit on the throughput *r*, that is, the rate at which information can be transmitted into the network: $r < W/RTT$.

The throughput cannot exceed the maximum bit rate $R = DBP/2t_p$ that is available for the source to transmit into the network. Therefore, the throughput increases as the window size is increased, but cannot exceed the bit rate *R*:

Throughput = $r = \min\{R, W/RTT\} = \min\{DBP/2*t_p, W/RTT\}$

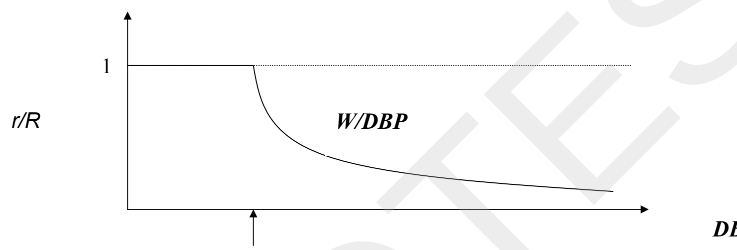
Suppose that the window size is less than the delay-bandwidth product. We then expect that the source cannot transmit at the maximum bit rate *R*. Indeed, we have that:

$r < W/RTT < W/2t_p$.

Therefore we have that:

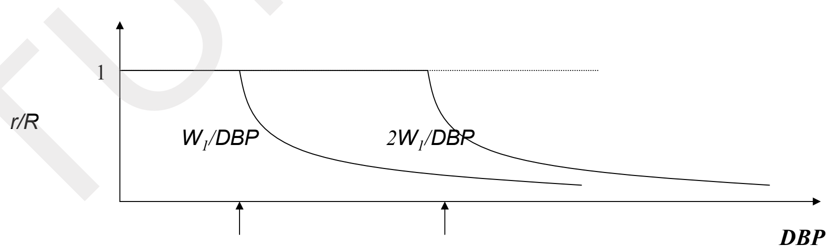
$r/R < W/(R*2t_p) = W/DBP$.

We conclude that the ratio of the maximum achievable throughput to *R* is less than the ratio of the window size to the *DBP*, as shown in the figure below.



b. In the above plot include the maximum achievable throughput when the above window size is scaled up by a factor of 2^K, where K = 4, 8, 12.

The following figure shows the case where the window size is doubled.



c. Place the following scenarios in the plot obtained in part (b): Ethernet with 1 Gbps and distance 100 meters; 2.4 Gbps and distance of 6000 km; satellite link with speed of 45 Mbps and RTT of 500 ms; 40 Gbps link with distance of 6000 km.

The window sizes (in bytes) of interest are: a. 65,536=2^{16}; b. 2^{20}=10^6 ; c. 2^{24}=16.7x10^6 ; d. 2^{28}=2.68x10^8.

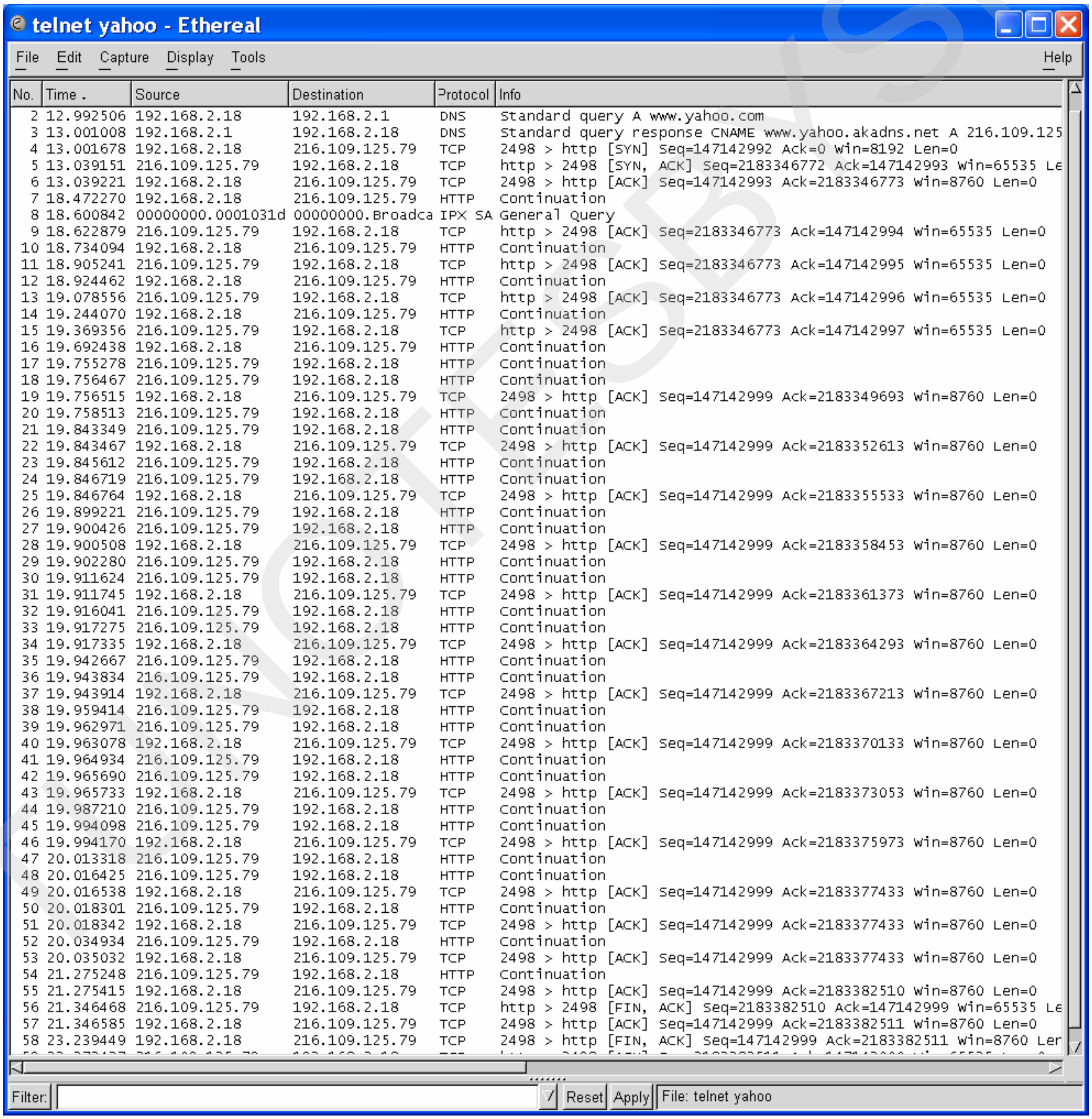
Case	*DBP* implied
Ethernet *R* = 1 Gbps *D* = 100 m	T_p = 100 / 2.5x10^8 = 4x10^{-7} *DBP* = 2*T_p*R = 8x10^{-7} * 1x10^6 *DBP* = 8 x 10^{-1} ***DBP* < 1 bit** ***Scenario a***
OC-48 Link *R* = 2.4 Gbps *D* = 6000 km	T_p = 6 x 10^3 / 2.5x10^5 = 2.4 x 10^{-1} *DBP* = 2.4 x10^{-1} * 2.4x10^9 *DBP* = 2.4 x 10^8 ***DBP* = 2400 Mbits (300 Mbytes)** ***Scenario slightly beyond d***
Satellite link *R* = 45 Mbps *RTT* = 500 ms (5x10^{-1} sec)	*DBP* = *RTT***R* *DBP* = 5x10^{-1} * 45x10^6 *DBP* = 225 x 10^5 bits ***DBP* = 21 Mbits (< 2.6 Mbytes)** ***Scenario c***
OC-768 link *R* = 40 Gbps *D* = 6000 km (6x10^6 m)	T_p = 6 x 10^3 / 2.5x10^5 T_p = 2.4 x 10^{-1} *DBP* = 2*T_p*R *DBP* = 2*2.4 x10^{-1} * 40x10^9 *DBP* = 192 x 10^8 ***DBP* = 18000 Mbits (2288 Mbytes)** ***Scenario beyond d***

50. Use a network analyzer to capture the sequence of packets in a TCP connection. Analyze the contents of the segments that open and close the TCP connection. Estimate the rate at which information is transferred by examining the frame times and the TCP sequence numbers. Do the advertised windows change during the course of the connection?

Solution:

The packet capture below is from a session with the HTTP server for yahoo.com. The initial sequence number for segments from the server can be seen from frame 5 to be 2183346772, so the first data byte has sequence number 2183346773. The FIN sent from the server in frame 56has sequence number 2183382510. Thus the server sent 35, 737 bytes. These bytes were received in the period between frame 7 (time=18.47 seconds) and frame 54 (time =21.27 seconds), so the elapsed time was about 2.8 seconds. The bit rate is then about 101 kbps.

The client window size remains fixed at 8760 bytes and the server window size remains fixed at 65535 bytes.



No.	Time	Source	Destination	Protocol	Info
2	12.992506	192.168.2.18	192.168.2.1	DNS	Standard query A www.yahoo.com
3	13.001008	192.168.2.1	192.168.2.18	DNS	Standard query response CNAME www.yahoo.akadns.net A 216.109.125
4	13.001678	192.168.2.18	216.109.125.79	TCP	2498 > http [SYN] Seq=147142992 Ack=0 Win=8192 Len=0
5	13.039151	216.109.125.79	192.168.2.18	TCP	http > 2498 [SYN, ACK] Seq=2183346772 Ack=147142993 Win=65535 Le
6	13.039221	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142993 Ack=2183346773 Win=8760 Len=0
7	18.472270	192.168.2.18	216.109.125.79	HTTP	Continuation
8	18.600842	00000000.0001031d	00000000.Broadca	IPX SA	General Query
9	18.622879	216.109.125.79	192.168.2.18	TCP	http > 2498 [ACK] Seq=2183346773 Ack=147142994 Win=65535 Len=0
10	18.734094	192.168.2.18	216.109.125.79	HTTP	Continuation
11	18.905241	216.109.125.79	192.168.2.18	TCP	http > 2498 [ACK] Seq=2183346773 Ack=147142995 Win=65535 Len=0
12	18.924462	192.168.2.18	216.109.125.79	HTTP	Continuation
13	19.078556	216.109.125.79	192.168.2.18	TCP	http > 2498 [ACK] Seq=2183346773 Ack=147142996 Win=65535 Len=0
14	19.244070	192.168.2.18	216.109.125.79	HTTP	Continuation
15	19.369356	216.109.125.79	192.168.2.18	TCP	http > 2498 [ACK] Seq=2183346773 Ack=147142997 Win=65535 Len=0
16	19.692438	192.168.2.18	216.109.125.79	HTTP	Continuation
17	19.755278	216.109.125.79	192.168.2.18	HTTP	Continuation
18	19.756467	216.109.125.79	192.168.2.18	HTTP	Continuation
19	19.756515	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183349693 Win=8760 Len=0
20	19.758513	216.109.125.79	192.168.2.18	HTTP	Continuation
21	19.843349	216.109.125.79	192.168.2.18	HTTP	Continuation
22	19.843467	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183352613 Win=8760 Len=0
23	19.845612	216.109.125.79	192.168.2.18	HTTP	Continuation
24	19.846719	216.109.125.79	192.168.2.18	HTTP	Continuation
25	19.846764	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183355533 Win=8760 Len=0
26	19.899221	216.109.125.79	192.168.2.18	HTTP	Continuation
27	19.900426	216.109.125.79	192.168.2.18	HTTP	Continuation
28	19.900508	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183358453 Win=8760 Len=0
29	19.902280	216.109.125.79	192.168.2.18	HTTP	Continuation
30	19.911624	216.109.125.79	192.168.2.18	HTTP	Continuation
31	19.911745	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183361373 Win=8760 Len=0
32	19.916041	216.109.125.79	192.168.2.18	HTTP	Continuation
33	19.917275	216.109.125.79	192.168.2.18	HTTP	Continuation
34	19.917335	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183364293 Win=8760 Len=0
35	19.942667	216.109.125.79	192.168.2.18	HTTP	Continuation
36	19.943834	216.109.125.79	192.168.2.18	HTTP	Continuation
37	19.943914	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183367213 Win=8760 Len=0
38	19.959414	216.109.125.79	192.168.2.18	HTTP	Continuation
39	19.962971	216.109.125.79	192.168.2.18	HTTP	Continuation
40	19.963078	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183370133 Win=8760 Len=0
41	19.964934	216.109.125.79	192.168.2.18	HTTP	Continuation
42	19.965690	216.109.125.79	192.168.2.18	HTTP	Continuation
43	19.965733	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183373053 Win=8760 Len=0
44	19.987210	216.109.125.79	192.168.2.18	HTTP	Continuation
45	19.994098	216.109.125.79	192.168.2.18	HTTP	Continuation
46	19.994170	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183375973 Win=8760 Len=0
47	20.013318	216.109.125.79	192.168.2.18	HTTP	Continuation
48	20.016425	216.109.125.79	192.168.2.18	HTTP	Continuation
49	20.016538	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
50	20.018301	216.109.125.79	192.168.2.18	HTTP	Continuation
51	20.018342	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
52	20.034934	216.109.125.79	192.168.2.18	HTTP	Continuation
53	20.035032	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
54	21.275248	216.109.125.79	192.168.2.18	HTTP	Continuation
55	21.275415	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183382510 Win=8760 Len=0
56	21.346468	216.109.125.79	192.168.2.18	TCP	http > 2498 [FIN, ACK] Seq=2183382510 Ack=147142999 Win=65535 Le
57	21.346585	192.168.2.18	216.109.125.79	TCP	2498 > http [ACK] Seq=147142999 Ack=2183382511 Win=8760 Len=0
58	23.239449	192.168.2.18	216.109.125.79	TCP	2498 > http [FIN, ACK] Seq=147142999 Ack=2183382511 Win=8760 Ler

51. Explain why framing information is required even in the case where frames are of constant length.

Solution:

To detect frame boundaries based on the frame length, the first frame must be detected correctly, which requires synchronization information. Also the receiver must have exact timing of the transmitter which is not practical and as a result the bit synchronization may be lost. The synchronization may be lost because of bit errors or loss of data, or loss of signal in the link.

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

Solution:

The inserted stuff bits are underlined.

1101111111011111110101 → 11011111$\underline{0}$11011111$\underline{0}$110101

53. Perform bit de-stuffing for the following sequence: 11101111101111100111110.

Solution:

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

11101111101111100111110 → 111011111-11111-011111-

54. Consider the PPP byte stuffing method. 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:

$\underline{\text{0x7D 0x5E}}$ 0xFE 0x24 $\underline{\text{0x7D 0x5D}}$ $\underline{\text{0x7D 0x5D}}$ 0x62 $\underline{\text{0x7D 0x5E}}$

→ 0x7E 0xFE 0x24 0x7D 0x7D 0x62 0x7E

55. Suppose that GFP operates over an error-free octet-synchronous physical layer. Find the average time the GFP receiver spends in the hunt state.

Solution:

At the beginning of its operation, the GFP receiver takes 4 bytes and checks to see if the second two bytes correspond to the check sum of the first pair of bytes. There is exactly one 16-bit pattern that corresponds to this checksum and so the probability that this pattern occurs at random is $2^{-16} = 1.5 \times 10^{-5}$, so most of the time the receiver immediately determines a given phase is incorrect. If the frame size is F, then on average F/2 bytes will be examined until the beginning of the frame is identified.

56. For GFP framing find the probability that the PLI and cHEC are not consistent. Assume that bit errors occur at random with probability p. Find the probability that the PLI and cHEC are not consistent in two consecutive frames.

Solution:

Considering that single bit errors are corrected:

$$P_{error} \approx P[errors \geq 2] \approx \binom{32}{2} p^2 (1-p)^{30} \approx 496 p^2$$

Probability of error in two consecutive frames = $(P_{error})^2 \approx 246016 p^2$

For example, if $p = 10^{-6}$, then probability of error in two consecutive frames is approximately 2×10^{-7}.

57. What is the maximum efficiency of GFP with respect to header overhead?

Solution:

The header is at least 4 bytes and the payload is at most 2^{16} bytes.

Maximum efficiency = $2^{16} / (2^{16} + 4) = 0.99994$

58. Can GFP be used to carry video signals? If so, how would you handle the case where each video image produces a variable length of compressed information? How would you handle the case where each video frame produces a fixed length of compressed information?

Solution:

GFP can be used to carry video signals. In the case of variable-length information, frame-mapped mode is used in which variable-length GFP frames are sent. In the case of fixed-length information, transparent mode is used in which fixed-length GFP frames are sent.

59. Suppose a server has a 1 Gbps Ethernet link to a GFP transmitter. The GFP has an STS-1 SONET connection to another site. The GFP transmitter has a limited amount of buffering to accommodate frames prior to transmission. Explain how the GFP transmitter may use flow control on the Ethernet link to prevent buffer overflow. What type of flow control is preferred if the server and GFP transmitter are co-located? If the server and transmitter are 5 km apart?

Solution:

The GFP transmitter sets two threshold levels in the buffer for low and high buffer fill levels. Once the buffer fill level exceeds the high buffer fill threshold, the flow control mechanism stops the server. Once the buffer fill level crosses the low buffer fill threshold, the flow control mechanism signals the server to start transmission.

If the server and GFP transmitter are co-located X-ON X-OFF is preferred. If the server and GFP transmitter are not co-located sliding-window flow control is preferred.

60. An inverse multiplexer combines n digital transmission lines of bit rate R bps and makes them appear as a single transmission line of n x R bps. Consider an inverse multiplexer that combine multiple STS-1 SONET lines. Find an appropriate value of n to carry a 1 Gbps Ethernet stream. What is the improvement in efficiency relative to carrying the Ethernet stream on an OC-48 link?

Solution:

$1 \times 10^9 / 50 \times 10^6 = 20$, Efficiency = 100%

Efficiency if OC-48 is used: $1x10^9$ / $2.4x10^9$ = 42%

Improvement = 100 - 42 = 58%

61. 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:

Assume 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 8×10^5 bits = (time for 1 packet)(# of packets needed)

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

$$= (\frac{8\times500 bits}{56\times10^3 bps})[\frac{10^6 bytes}{(500-70) bytes}]$$

$$= 0.07143\frac{\sec}{packet}\times 2326\, packets$$

= 166.14 seconds

62. Compare PPP and GFP. Identify situations where one is preferable to the other.

Solution:

PPP uses character-based framing which requires byte stuffing. As a result the frame size varies depending on the character pattern. GFP resolves this problem by using frame length based framing. GFP can support synchronous services, while PPP cannot. However PPP provides useful capabilities through link and network control protocols.

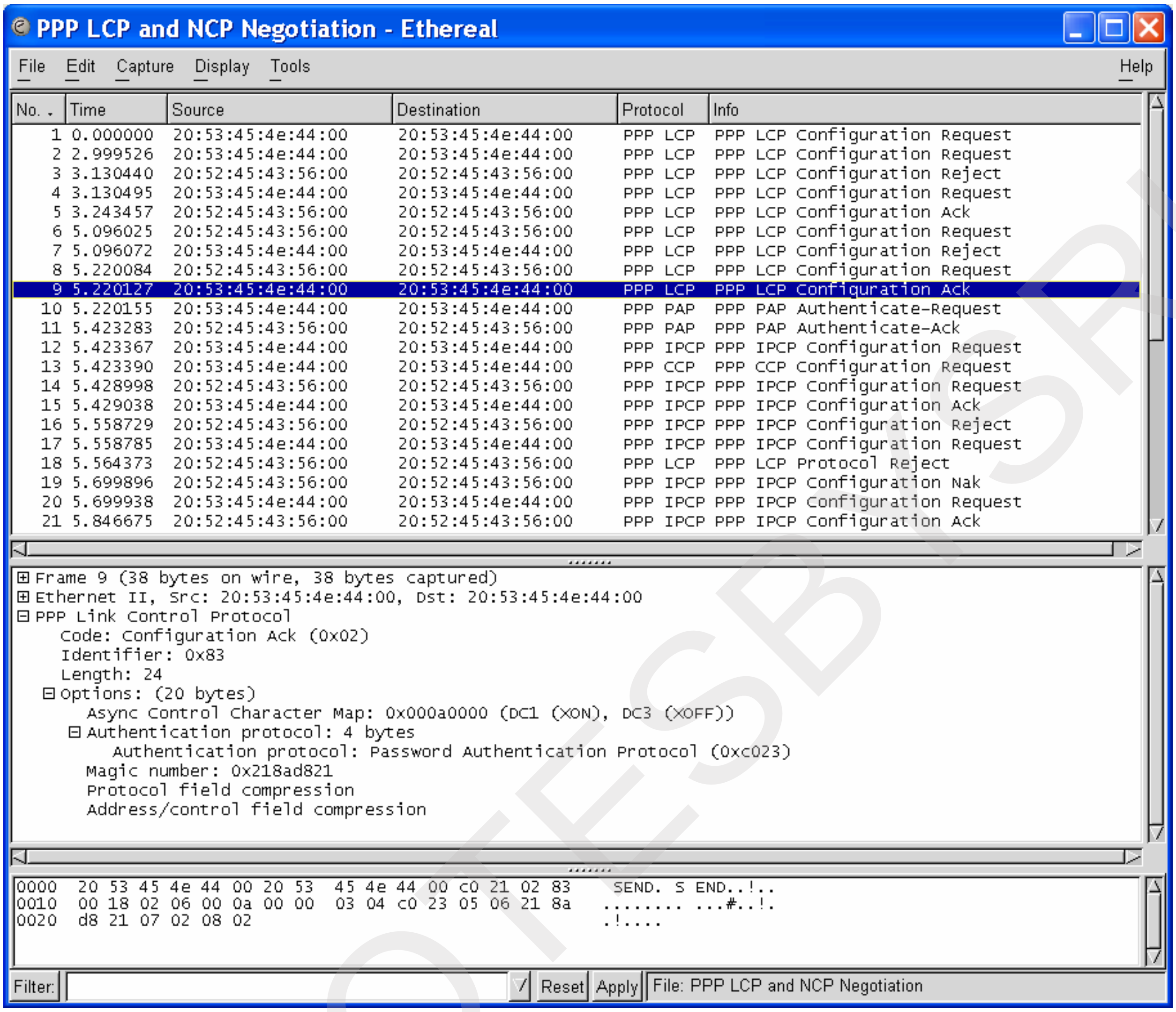
63. Use the Ethereal packet capture tool to analyze the exchange of packets during PPP setup using ad dial up connection to a local ISP. Start the Ethereal packet capture and then initiate the connection to the ISP. The number of captured packets will stop increasing after some point.

Solutions follow questions:

a. Analyze the options fields of the packets during the LCP negotiations. You may need to consult RFC 1662 to interpret some of the packet contents.

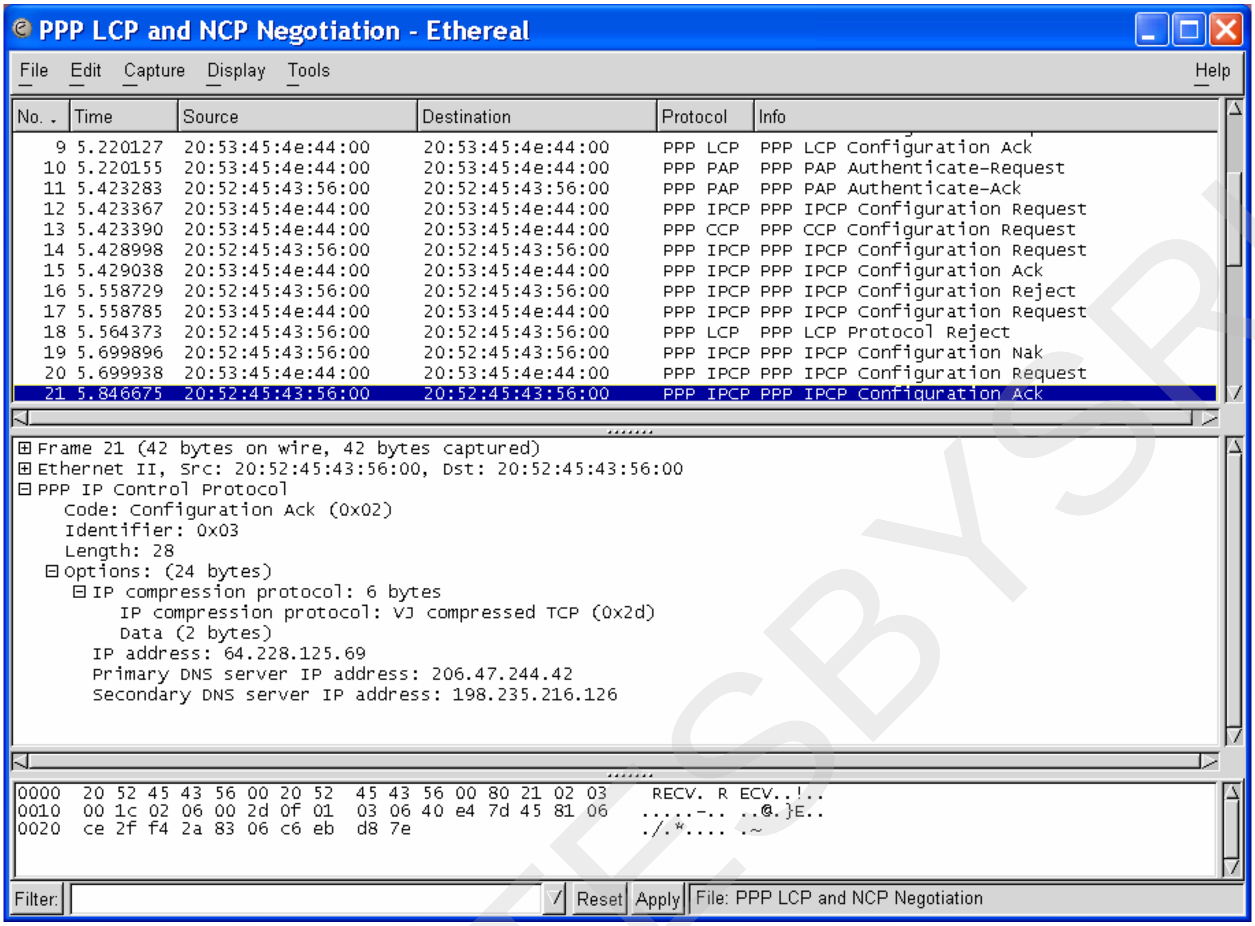
b. Examine the packet contents during the PAP exchange. Repeat the packet capture using CHAP.

The screen capture below shows a sequence of LCP and NCP negotiations for PPP. The twp stations exchange LCP messages where they propose and reject various configuration options. The highlighted frame (9) shows the final ACK accepting a configuration. It can be seen from the middle pane that PAP authentication will be used. Frame 10 sends the password request, and frame 11 sends the password.



c. Analyze the IPCP packet exchange. What IP addresses are configured during the exchange? What other options are negotiated? You may wish to consult RFC 1332 for this part.

The screen capture below shows a sequence the final IPCP ACK message confirming the use of protocol field compression and various IP addresses.



64. Explain the differences between PPP and HDLC.

Solution:

HDLC can support various data transfer modes, supports multipoint links and point to point links, and can implement error control and flow control mechanisms. PPP uses HDLC-like frames but does not use error control and flow control protocols. Instead PPP supports powerful link and network control protocols.

PPP is character based and can be implemented on any physical layer, HDLC is bit based and can be implemented only on bit synchronous physical layer.

65. 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 x 10^8 meters/second.

Solution:

The round trip propagation delay is:

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

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

Go-Back-N:

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

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

Selective Repeat:

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

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

66. 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?

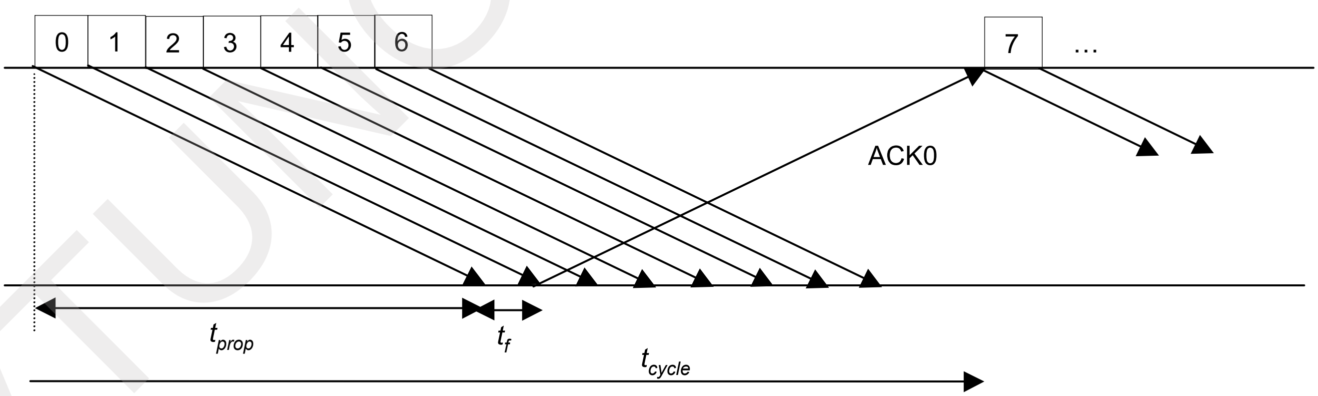
Solution:

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 x 10^6 / 3 x 10^8 = 120 ms

$t_f = n_f/R$ = 2000/1.5 x 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 + 2x120 = 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:

$R_{max} = N.n_f / t_{cycle}$ = 127x2000/(241.33x10^{-3}) = 1.052 Mbps

67. 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).

68. 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.

69. 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.

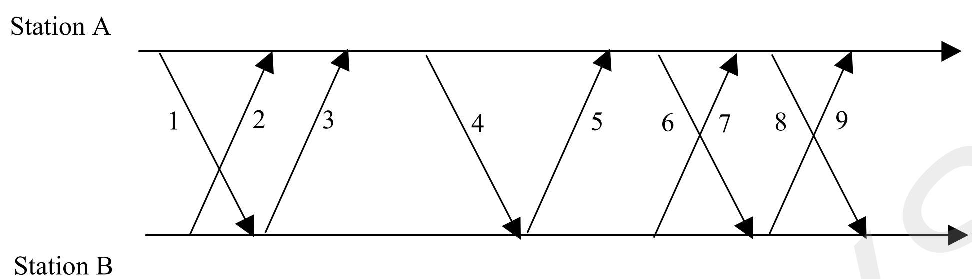
70. HDLC specifies that the N(R) in a SREJ frame requests the retransmission of frame N(R) and also acknowledges all frames up to N(R) – 1. Explain why only one SREJ frame can be outstanding at a given time.

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.

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

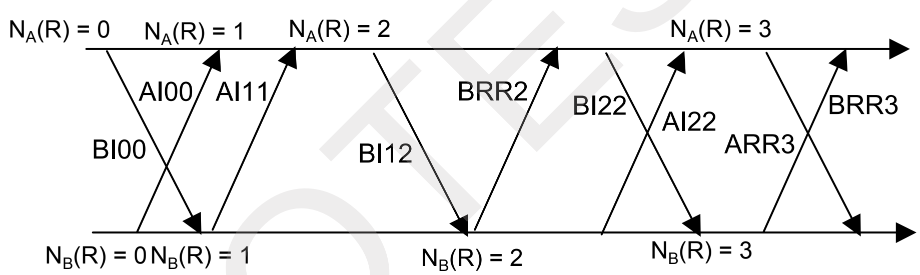


1. BI00	2. AI00	3. xIxx	4. xIxx	5. xRRxx
6. xIxx	7. xIxx	8. xRRxx	9. xRRxx	

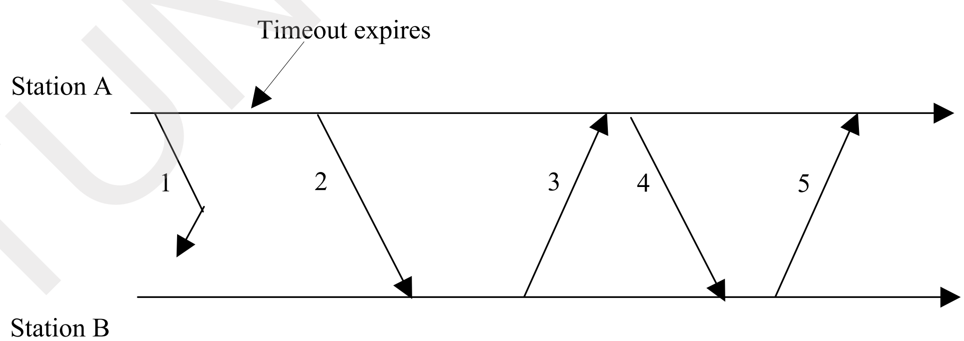
a. Complete the diagram by completing the labeling of the frame exchanges.

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

Solution:



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

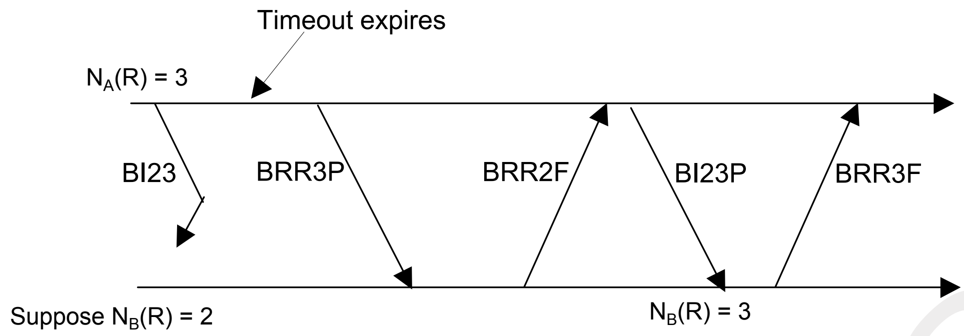


1. BI23	2. xRRxP	3. xRRxy	4. xIxx	5. xRRx

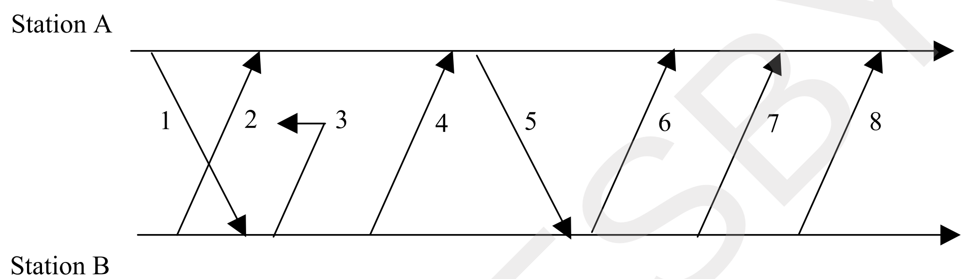
a. Complete the diagram in HDLC ABM by completing the labeling of the frame exchanges.

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

Solution:



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

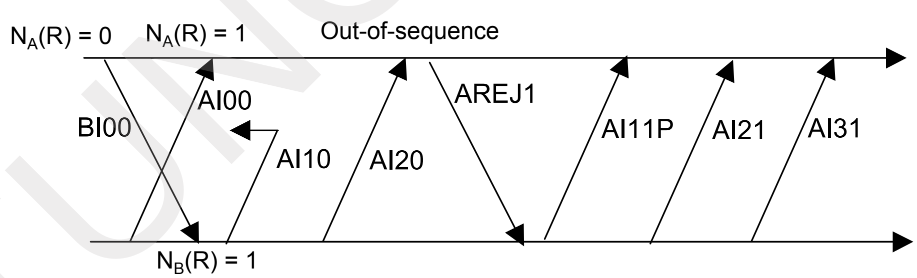


1. BI00	2. AI00	3. xIxx	4. xIxx
5. xREJx	6. xIyx	7. xyxx	8. xyxx

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:



74. 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:

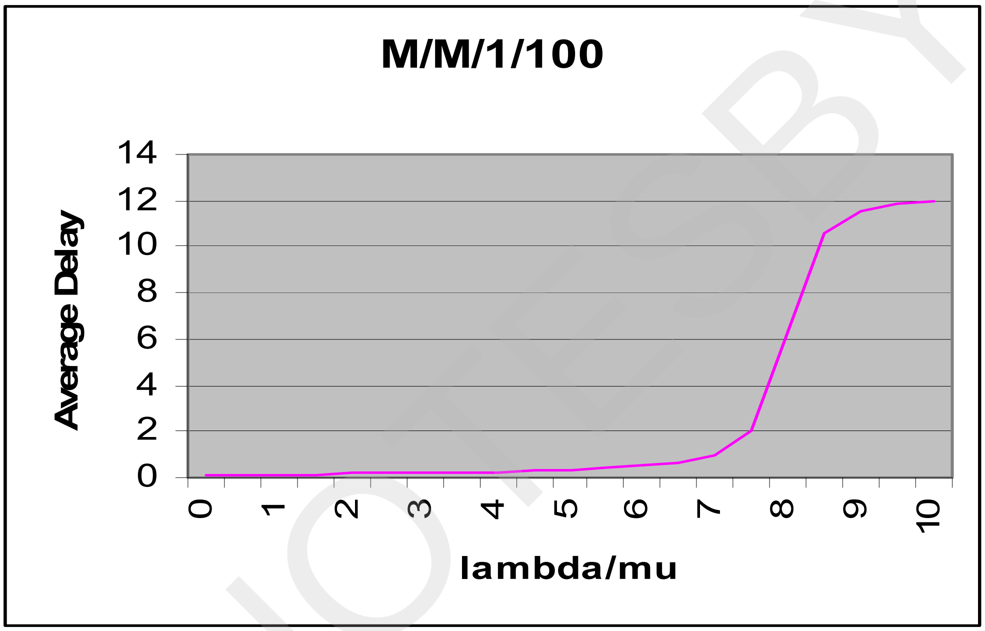
This is an M/M/1/K system in which:

Service rate μ = 64000 bps / 8000 bits/packet = 8 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 $E[N] = \dfrac{\lambda}{\mu-\lambda} - \dfrac{(K+1)\lambda^{K+1}}{\mu^{K+1}-\lambda^{K+1}}$



75. 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

76. 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:

Assume 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.

$R = 10^6$ bps
L = constant (250, 500, and 1000 bytes)
$K \rightarrow \infty$
n_o = 48 bytes
$\mu = R/(8*L)$ packets/sec = 500 (L=250 B); 250 (L=500 B); 125 (L=1000)

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

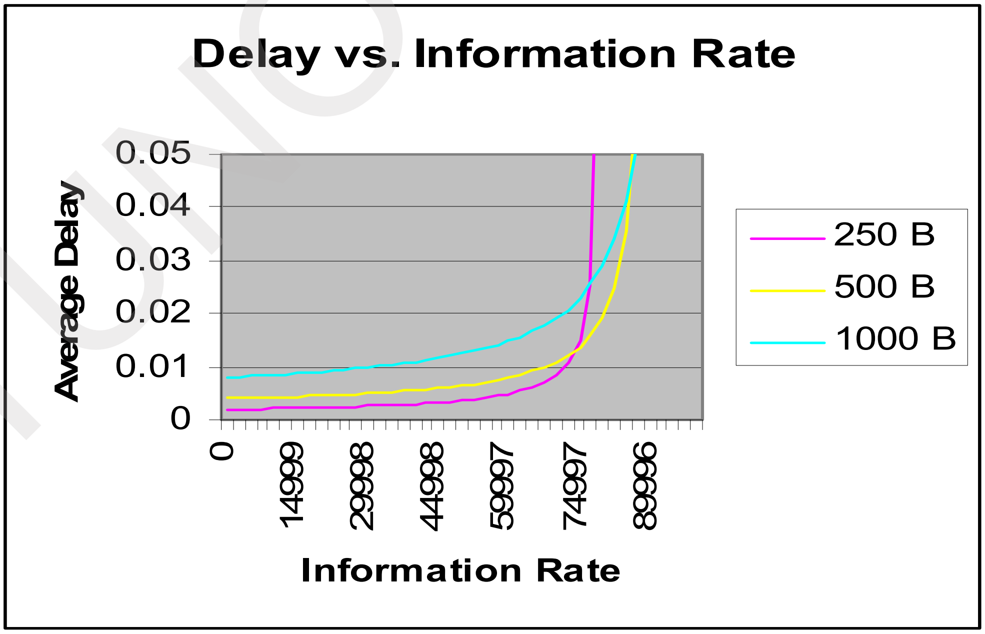
Average delay, $E[T_D] = (1+\frac{\lambda}{2(\mu-\lambda)})\frac{1}{\mu}$

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

$$\gamma = \lambda(8L)\frac{(L-n_o)}{L} = \lambda(8L)(1-\frac{48}{L}) \text{ or } \lambda = \frac{\gamma}{8L-384}$$

And so, the delay as a function of the user information bit rate is

$$E[T_D] = (1+\frac{\frac{\gamma}{8L-384}}{2(\frac{R}{8L}-\frac{\gamma}{8L-384})})\frac{8L}{R}$$



77. 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 is sent to the second line, so the average number sent to line 2 is 6 – 4.59 = 1.41 packets per T-second period.

78. 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.

Changes to this document

Version No.	*Problem No.*	*Change*
1.00 to 1.01	5.14a	Changed reference Figure 5.12 to 5.10 in solution text
	5.35	Changed reference Table 5.12 to Equation 5.11 in solution text.
	Throughout	Cosmetic changes resulting in changes in pagination

Solutions to Chapter 6

1. Why do LANs tend to use broadcast networks? Why not use networks consisting of multiplexers and switches?

Solution:

The computers in a LAN are separated by a short distance (typically < 100m) so high speed and reliable communication is possible using a shared broadcast medium. The cost of the medium is negligible and the overall cost is dominated by the cost of the network interface cards in each computer. In addition, the LAN users usually belong to the same group where all users are generally trusted, so broadcast does not pose much security danger.

The original reason for avoiding a multiplexer and switch approach to LANs is that a centralized, expensive "box" is required. The availability of Application Specific Integrated Circuits (ASICs) has reduced the cost of switching boxes and made switch-based LANs feasible, and in some environments the dominant approach.

2. Explain the typical characteristics of a LAN in terms of network type, bit rate, geographic extent, delay-bandwidth product, addressing, and cost. For each characteristic, can you find a LAN that deviates from the typical? Which of the above characteristics is most basic to a LAN?

Solution:

- Type: broadcast network in a bus, ring or star topology.
- Bit rate: from 1Mbps to 100 Mbps.
- Delay-bandwidth product: small.
- Addressing: flat.
- Geographical extent: up to 1000 m (small).
- Cost: low.
- 10 Gbps Ethernet deviates from the above characteristics in that it is non-broadcast, of much higher bit rate, large delay-bandwidth product, larger geographic extent and high cost.

The most basic characteristic of a LAN is small geographical extent.

3. Compare the two-channel approach (Figure 6.4) with the single-channel approach (Figure 6.5) in terms of the types of MAC protocols they can support.

Solution:

Figure 6.4 consists of two unidirectional channels, one outbound from a central node to secondary nodes, and another inbound from the secondaries to the central node. The bandwidth that is available in each direction is fixed. This arrangement can support polling protocols as well as contention protocols in the inbound direction. Figure 6.5 provides a single channel that is shared by all stations. This arrangement also supports polling and contention-type MAC protocols. The bandwidth available in each direction can be controlled dynamically.

4. Suppose that the ALOHA protocol is used to share a 56 kbps satellite channel. Suppose that frames are 1000 bits long. Find the maximum throughput of the system in frames/second.

Solution:

Maximum throughput for ALOHA = 0.184
Maximum throughput in frames/sec = (56000 bits/sec) x (1 frame/1000 bits) x 0.184 = 10.304
The maximum throughput is approximately 10 frames/sec.

5. Let *G* be the total rate at which frames are transmitted in a slotted ALOHA system. What proportion of slots goes empty in this system? What proportion of slots go empty when the system is operating at its maximum throughput? Can observations about channel activity be used to determine when stations should transmit?

Solution:

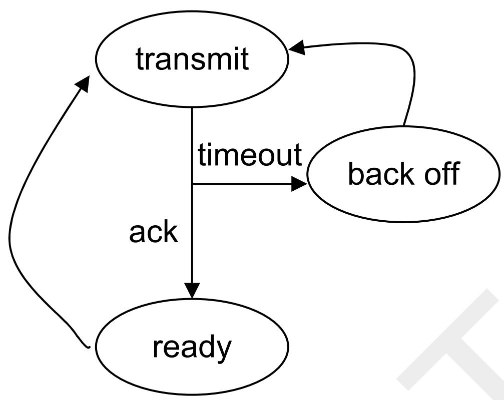
Proportion of empty slots = P[0 transmission] = $[G^0/0!]e^{-G} = e^{-G}$

Maximum throughput = 0.368; $G_{mt} = 1$
Proportion of empty slots at maximum throughput = $e^{-1} = 0.368$

Any attempt to decrease the proportion of empty slots below e^{-1} is counterproductive as this action will push the throughput below its maximum value.

6. Modify the state transition diagram of Stop-and-Wait ARQ to handle the behavior of a station that implements the ALOHA protocol.

Solution:



7. Suppose that each station in an ALOHA system transmits its frames using spread spectrum transmission. Assume that the spreading sequences for the different stations have been selected so that they have low cross-correlations. What happens when transmissions occur at the same time? What limits the capacity of this system?

Solution:

The transmitted signals from different stations occupy the whole frequency band simultaneously. To each receiver, the aggregate of the other signals appears like noise after the receiver applies its spreading sequence in the demodulation. Consequently, the receiver can reliably extract its desired signal as long as the signal to noise ratio remains sufficiently high.

The system capacity is limited by the maximum amount of signal interference allowed at the receiver. Increasing the number of signals in the system increases the interference power level. As the number of signals is increased, the SNR decreases and the bit error rate in each receiver will increase.

8. Consider four stations that are all attached to two different bus cables. The stations exchange fixed-size frames of length 1 second. Time is divided into slots of 1 second. When a station has a frame to transmit, the station chooses either bus with equal probability and transmits at the beginning of the next slot with probability *p*. Find the value of *p* that maximizes the rate at which frames are successfully transmitted.

Solution:

To maximize the successful transmission rate is to maximize the probability of successful transmission.

P(success) = (number of stations) × *P*(one station transmits on one bus) × *P*(no other station transmit on the same bus)

$$= 4(\frac{1}{2}p)(1-\frac{1}{2}p)^3 = 2p(1-\frac{1}{2}p)^3$$

Take the derivative with respect to p,

$$\frac{\partial P(success)}{\partial p} = 2(1-\frac{1}{2}p)^3 - (3p)(1-\frac{1}{2}p)^2$$

set it to 0 and find the value of p that maximizes P(success).

$$p = \frac{1}{2}$$

9. In a LAN, which MAC protocol has a higher efficiency: ALOHA or CSMA-CD? What about in a WAN?

Solution:

The maximum efficiency achieved by the Slotted ALOHA is 0.368. The efficiency of CSMA-CD is given by 1/(1 + 6.4a), and is sensitive to $a = t_{prop}R/L$, the ratio between delay-bandwidth product and frame length.

In a LAN environment, the end-to-end distance is around 100m and the transmission rates are typically 10Mbps, 100Mbps and 1Gbps (See Table 6.1). An Ethernet frame has a maximum length of 1500 bytes = 12,000 bits.

The table shows the efficiency of CSMA-CD at various transmission rate. Assume *L* = 12,000 bits and propagation speed of 3 x 10^8.

	a	Efficiency
10 Mbps	3 x 10^{-4}	0.998
100 Mbps	3 x 10^{-3}	0.981
1 Gbps	3 x 10^{-2}	0.839

Note however that if shorter frame sizes predominate, e.g. 64 byte frames, then a increases by a factor of about 20. According to the above formula the efficiency of CSMA-CD at 1 Gbps then drops to about 0.7. The situation however is worse in that the minimum frame size at 1 Gbps needs to be extended to 512 bytes, as discussed in page 436 of the text.

In a WAN environment *d* is larger. Assuming 100 Km, a is larger by a factor of 10^3 resulting in an efficiency of 0.36, 0.05, and 0.005 respectively for 10 Mbps, 100 Mbps, and 1 Gbps transmission rates. In the case of 10 Mbps transmission rate the efficiency of CSMA-CD is close to the efficiency of ALOHA but in the other two cases it is much less than ALOHA.

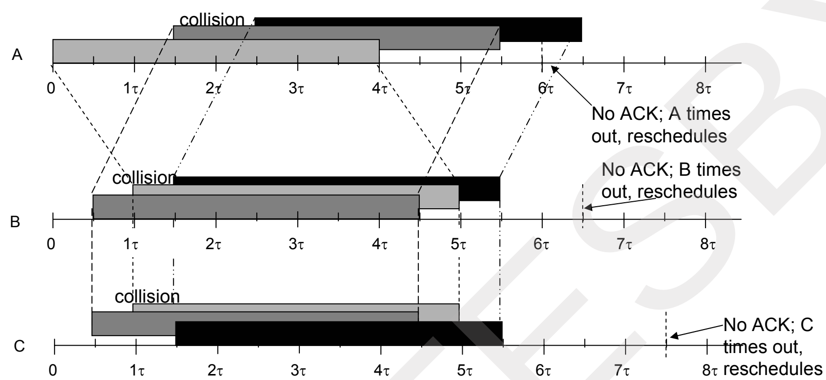
10. A channel using random access protocols has three stations on a bus with end-to-end propagation delay τ. Station A is located at one end of the bus, and stations B and C are together located at the other end of the bus. Frames arrive at the three stations and are ready to be transmitted at stations A, B, and C at the respective times t_A = 0, t_B = τ/2, and t_C = 3τ/2. Frames require transmission times of 4τ. In appropriate figures, with time as the horizontal axis, show the transmission activity of each of the three stations for

Frame arrival times:

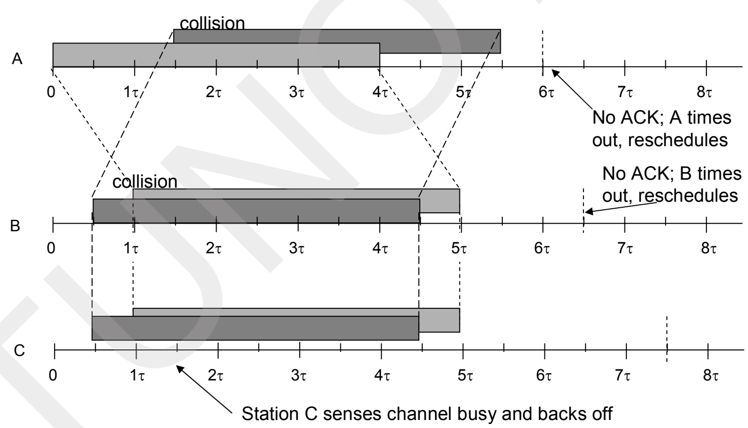
A: $t_A = 0$
B: $t_B = \tau/2$
C: $t_C = 3\tau/2 = 1\ 1/2\ \tau$

$t_p = \tau$ and $X = 4\tau$

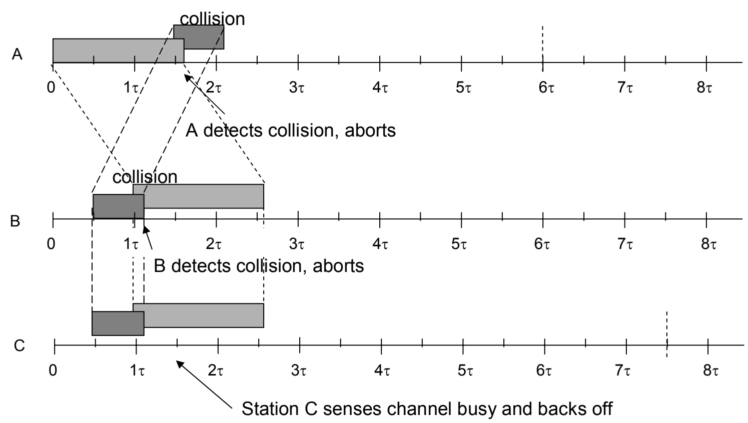
a. ALOHA



b. Non-persistent CSMA



c. Non-persistent CSMA-CD.



11. Estimate the maximum throughput of the CDPD system assuming packets are 1096 bytes in length.

Hint: What is a for this system?

Solution:

During the contention interval, it takes two microblock delays ($2t_{\mu\text{block}}$) for a station to know if it has captured the reverse channel. Divide the time into minislots of $2t_{\mu\text{block}}$.

As in the CSMA-CD system, the maximum probability of successful transmission is $P_{success}^{\max} = \frac{1}{e}$, and the average number of minislots in a contention period is $E[J] = \frac{1}{P_{success}^{\max}} = e$.

The maximum throughput occurs when the channel time is spent in packet transmission followed by a contention period. After the packet transmission $E[X]$, each station spends one $t_{\mu\text{block}}$ to determine if the channel is idle or busy.

The maximum throughput is given by

$$\rho_{\max} = \frac{E[X]}{E[X] + t_{ublock} + e(2t_{ublock})} = \frac{1}{1+(2e+1)a} \quad \text{where } a = \frac{t_{ublock}}{E[X]}.$$

Calculation:

$$t_{ublock} = \frac{60bits}{19{,}200bps} = 3.1ms$$

Assume the packet is already in HDLC format.

$$1096\,bytes = 8768\,bits$$

Packet length after segmentation and header insertion:

$$L = \frac{8768}{274} \times \frac{378}{54} \times (54+6) = 13440 bits$$

$$E[X] = \frac{13440 bits}{19200 bps} = 0.7s$$

$$a = \frac{3.1ms}{0.7s} = 4.43 \times 10^{-3}$$

Therefore, the maximum throughput is

$$\rho_{\max} = \frac{1}{1+(2e+1)(4.43\times 10^{-3})} = 0.97 .$$

12. Can the Digital Sense Multiple Access protocol, which is used by CDPD, also be used on the digital carrier of GSM? If yes, explain how.

Solution:

Digital Sense Multiple Access would require redesign of the GSM frame structure. In the downlink transmission broadcast from the base station, a version of the flag word would need to be inserted to indicate whether the medium is busy. Furthermore since the frame structure in GSM effectively divides the frame into 8 channels, a separate Digital Sense Multiple Access protocol would need to be in operation for each channel. This in turn would require the presence of a flag word per channel in the downlink frames.

13. M terminals are attached by a dedicated pair of lines to a hub in a star topology. The distance from each terminal to the hub is d meters, the speed of the transmission lines is R bits/second, all frames are of length 12500 bytes, and the signal propagates on the line at a speed of 2.5 (10^8) meters/second. For the four combinations of the following parameters {d = 25 meters or d = 2500 meters; R = 10 Mbps or R = 10 Gbps}, compare the maximum network throughput achievable when the hub is implementing: Slotted ALOHA; CSMA/CD.

Solution:

L = 12500 x 8 bits, $t_{prop} = d$ / (2.5 x 10^8 meters/sec), $a = t_{prop} R/L$

Values for a:

R/d

	2x25	2x2500
1.00E+07	2E-05	2E-03
1.00E+10	2E-02	2E+00

Maximum Throughput for Slotted ALOHA:

R/d

	2x25	2x2500
1.00E+07	0.367879	0.367879
1.00E+10	0.367879	0.367879

Maximum throughput for CSMA-CD:

R/d

	2x25	2x2500
1.00E+07	0.999872	0.98736
1.00E+10	0.886525	0.07246

14. Consider the star-topology network in problem 6.13 when the token-ring protocol is used for medium access control. Assume single-frame operation, eight-bit latency at each station, $M = 125$ stations. Assume a free token is three bytes long.

Solutions follow questions:

$M = 125$
$b = 8$ bits
$L_{token} = 3$ bytes
$L_{frame} = 12{,}500$ bytes
$v = 2.5 \times 10^8$ m/sec

a. Find the effective frame transmission time for the four combinations of d and R.

The distance from each terminal to the hub is d meters, so the total distance around the ring is then $M2d$. Assuming *single frame transmission token reinsertion*, then

X_{eff} = token transmission time + frame transmission time + ring latency

$$= \frac{L_{token} + L_{frame}}{R} + M\left(\frac{2d}{v} + \frac{b}{R}\right)$$

b. Assume that each station can transmit up to a maximum of k frames/token. Find the maximum network throughput for the four cases of d and R.

The maximum throughput occurs when all stations transmit k frames per token. After completing the transmission of k frames, each station waits one ring latency time and then transmits a free token into the ring.

$$\rho_{\max} = \frac{M \times k \times X_{frame}}{M(k \times X_{frame} + X_{token} + \tau) + \tau} = \frac{1}{1 + \dfrac{X_{token} + \tau}{kX_{frame}} + \dfrac{\tau}{MkX_{frame}}}$$

where

$\tau = M\left(\frac{2d}{v} + \frac{b}{R}\right)$ is the amount time spent in passing the token around the ring.

Calculation:

i. $d = 25$ m, $R = 10$ Mbps

$\tau = 0.125$ ms, $X_{frame} = 10$ ms, $X_{token} = 2.4\mu$sec,

$$X_{eff} = \frac{(3 + 12500) \times 8}{10M} + 125\left(\frac{2 \times 25}{2.5 \times 10^8} + \frac{8}{10M}\right) = 10.1 \text{ ms}$$

$$\rho_{\max} \approx \frac{1}{1+\frac{.125}{10k}} \approx 99\%$$

X_{token} and τ are negligible.

ii. d = 25 m, R = 10 Gbps

τ = 25.1 μsec, X_{frame} = 10 μsec, X_{token} = 0.0024 μsec,

X_{eff} = 35.1 μsec

$$\rho_{\max} \approx \frac{1}{1+\frac{25.1}{10k}} = \frac{1}{1+\frac{2.5}{k}}$$

ρ_{max} = 28% for k =1 and improves as k increases.

iii. d = 2500 m, R = 10 Mbps

τ = 2.6 ms, X_{frame} = 10 ms, X_{token} = 2.4 μsec,

X_{eff} = 12.6 ms

$$\rho_{\max} \approx \frac{1}{1+\frac{2.6}{10k}} = \frac{1}{1+\frac{0.26}{k}}$$

ρ_{max} = 79% for k = 1 and improves as k increases.

iv. d = 2500 m, R = 10 Gbps

τ = 2.5 ms, X_{frame} = 10 μsec, X_{token} = 0.0024 μsec,

X_{eff} = 2.51 ms

$$\rho_{\max} \approx \frac{1}{1+\frac{2.5}{.010k}} = \frac{1}{1+\frac{250}{k}}$$

ρ_{max} = 0.40%, nearly zero.

15. A wireless LAN uses polling to provide communications between M workstations and a central base station. The system uses a channel operating at 25 Mbps. Assume that all stations are 100 meters from the base station and that polling messages are 64 bytes long. Assume that frames are of constant length of 1250 bytes. Assume that stations indicate that they have no frames to transmit with a 64-byte message.

Solutions follow questions:

d = 100 m between the base station and the stations
$v = 3 \times 10^8$ m/sec
$t_{prop} = 100 / (3 \times 10^8) = 0.33\ \mu$sec
R = 25 Mbps
X_{frame} = 10000/25Mbps = 400 μsec
X_{poll} = 512/25Mbps = 20 μsec
R = 2.5 Gbps
X_{frame} = 10000/2.5Gbps = 4 μsec
X_{poll} = 512/2.5Gbps = 0.2 μsec
$X_{end} = X_{poll}$

a. What is the maximum possible arrival rate that can be supported if stations are allowed to transmit an unlimited number of frames/poll?

$\rho_{max} = 1$ and $\rho = \lambda X$
R = 25 Mbps
λ_{max} = 1/400 μsec = 2,500 frames/sec

b. What is the maximum possible arrival rate that can be supported if stations are allowed to transmit N frames/poll?

$$\rho_{max} = \frac{MNX_{frame}}{M\{NX_{frame} + X_{end} + X_{poll} + 2t_{prop}\}} = \frac{1}{1 + \frac{20 + 20 + 0.66}{400\text{N}}} \approx \frac{1}{1 + \frac{0.1}{\text{N}}}, \quad \lambda_{max} = \frac{\rho_{max}}{\text{X}}$$ as N increases, ρ_{max} approaches 1.

R = 25 Mbps
$N = 10$, ρ_{max} = 99%
$\lambda_{max} = 1/400\mu$ = 2,500 frames/sec

c. Repeat parts (a) and (b) if the transmission speed is 2.5 Gbps.

(a) R = 2.5 Gbps
$\rho_{max} = 1$
λ_{max} = 1/4 μsec = 250,000 frames/sec

(b) R = 2.5 Gbps

$$\rho_{max} = \frac{1}{1 + \frac{0.2 + 0.2 + 0.66}{4\text{N}}} \approx \frac{1}{1 + \frac{0.26}{\text{N}}}, \quad \lambda_{max} = \frac{\rho_{max}}{\text{X}}$$

$N = 10$,
ρ_{max} = 97.4%
$\lambda_{max} = 0.788/4\mu$ = 243,700 frames/sec

16. A token-ring LAN network interconnects M stations using a star topology in the following way. All the input and output lines of the token-ring station interfaces are connected to a cabinet where the actual ring is placed. Suppose that the distance from each station to the cabinet is 100 meters and that the ring latency per station is eight bits. Assume that frames are 1250 bytes and that the ring speed is 25 Mbps.

Solutions follow questions:

d = 100 m from each station to the cabinet
v = 2 x 10^8 m/sec
b = 8 bits
L = 1250 bytes = 10000 bits
R = 25 Mbps

$$X = \frac{10000}{25M} = 4 \times 10^{-4} \text{ sec}$$

$$\tau' = \frac{M2d}{v} + \frac{M8}{R} = \frac{M200}{2 \times 10^8} + \frac{M8}{25 \times 10^6} = 1.32 \times 10^{-6} M$$

$$a' = \frac{\tau'}{X} = \frac{1.32 \times 10^{-6} M}{4 \times 10^{-4}} = 3.3 \times 10^{-3} M$$

a. What is the maximum possible arrival rate that can be supported if stations are allowed to transmit an unlimited number of frames/token?

When all stations are allowed to transmit an unlimited number of frames/token, ρ_{max} = 1 and $\rho = \lambda X$.

$$\lambda_{\max} = \frac{1}{4 \times 10^{-4}} = 2500 \text{ frames/sec}$$

b. What is the maximum possible arrival rate that can be supported if stations are allowed to transmit 1 frame/token using single-frame operation? Using multiple token operation?

Single-frame operation:

$$\rho_{\max} = \frac{1}{1 + a'\left(1 + \frac{1}{M}\right)}$$

$$\lambda_{\max} = \frac{1}{1 + (3.3 \times 10^{-3} M)(1 + \frac{1}{M})} \cdot \frac{1}{4 \times 10^{-4}} = \frac{2500}{1.0033 + 0.0033M}$$

Multitoken operation:

$$\rho_{\max} = \frac{1}{1 + a'/M} = \frac{1}{1 + 3.3 \times 10^{-3} M/M} = 0.997$$

$$\lambda_{\max} = 0.997 \frac{1}{4 \times 10^{-4}} = 2492 \text{ frames/sec}$$

c. Repeat parts (a) and (b) if the transmission speed is 2.5 Gbps.

R = 2.5 Gbps

$$X = \frac{10000}{2.5G} = 4 \times 10^{-6} \text{ sec}$$

$$\tau' = \frac{M2d}{v} + \frac{M8}{R} = \frac{M200}{2\times 10^8} + \frac{M8}{2.5G} = 1\times 10^{-6}M$$
$$a' = \frac{\tau'}{X} = \frac{1\times 10^{-6}M}{4\times 10^{-6}} = 0.25M$$

Unlimited number of frames per token:

$$\lambda_{\max} = \frac{1}{4\times 10^{-6}} = 250000 \text{ frames/sec}$$

Single-frame operation:

$$\lambda_{\max} = \frac{1}{1+(0.25M)(1+\frac{1}{M})} \cdot \frac{1}{4\times 10^{-6}} = \frac{250000}{1.25+0.25M} \text{ frames/sec}$$

Multitoken operation:

$$\lambda_{\max} = \frac{1}{1+0.25M/M} \cdot \frac{1}{4\times 10^{-6}} = 0.2\times 10^6 \text{ frames/sec}$$

17. Suppose that a LAN is to carry voice and packet data traffic. Discuss what provisions if any are required to handle the voice traffic in the reservation, polling, token ring, ALOHA and CSMA-CD environments. What changes if any are required for the packet data traffic?

Solution:

Voice traffic is delay sensitive, so the MAC protocols must ensure the delays experienced by voice data packets are sufficiently low.

The reservation, polling and token ring systems use scheduling approach to medium access control and so are better able to provide predictable delay performance. Random access approaches on the other hand have greater variability in delay.

Reservation: Reservation schemes can provide performance close to time-division multiplexing and hence can support voice traffic.

Polling: Polling schemes can be configured to provide a bound on the value on the cycle time. This provides a guaranteed delay bound for voice traffic.

Token Ring: Limit the transmission time per token to place a maximum value on the token rotation time. Avoid single-token or single-frame operation if ring latency is large. Priority token operation can also be used if there is non-voice traffic in the network.

The ALOHA and CSMA-CD systems use random access approach to medium access control. The system throughput and delay performance are affected by traffic load. At a high load, the collision rate is high, and the packets experience long delays. To meet the delay requirement of voice traffic, the maximum traffic load should be kept very low for Aloha and relatively low for CSMA-CD. The CSMA-CD systems can be modified to provide different levels of access priority. Lower delay performance can be provided by giving voice traffic higher priority.

Packet data traffic is usually not delay sensitive, but does require low loss. In carrying both voice and data, the MAC protocol must provide voice traffic with low delay and data with low loss. In general, this requires giving voice priority access over data. Otherwise surges in data traffic can introduce unacceptable delay for voice traffic.

18. A wireless LAN has mobile stations communicating with a base station. Suppose that the channel available has *W* Hz of bandwidth and suppose that the inbound traffic from the mobiles to the base is *K* times smaller than the outbound traffic from the base to the workstations. Two methods are considered for dealing with the inbound/outbound communications. In frequency-division duplexing the channel is divided into two frequency bands, one for inbound and one for outbound communications. In time-division duplexing all transmissions use the full channel but the transmissions are time-division multiplexed for inbound and outbound traffic.

Solutions follow questions:

a. Compare the advantages and disadvantages of the two methods in terms of flexibility, efficiency, complexity, and performance.

Time Division Duplexing (TDD) is more flexible than Frequency Division Duplexing (FDD) because it can be allocated dynamically without changing hardware. FDD is less flexible because the allocation of bandwidth to channels cannot be changed. However this is not a problem if the parameter *K* is fixed.

TDD is a more efficient than FDD because the need for frequency guard bands is avoided.

The relative performance of TDD and FDD depends on the nature of the traffic. If the network is working with steady traffic then both systems have roughly the same performance. But if the network is working with bursty traffic then in FDD the whole bandwidth is not used, whereas in TDD the empty slots can be used by others.

FDD systems tend to be simpler because they operate at a per-station bit rate, whereas TDD systems need to operate at aggregate bit rate.

b. How is the ratio *K* taken into account in the two methods?

For FDD, the amount of frequency bandwidth is allocated according the ratio *K*. In TDD the ratio *K* becomes apparent in the relative usage of time slots in either direction.

19. Consider the following variation of FDMA. Each station is allotted two frequency bands: a band on which to transmit and a band in which to receive reservations from other stations directing it to listen to a transmission from a certain station (frequency band) at a certain time. To receive a frame, a station tunes in the appropriate channel at the appropriate time. To make a reservation, a station transmits at the receiving stations reservation channel. Explain how transmitting and receiving stations can make use of the reservation channels to schedule frame transmissions.

Solution:

Suppose station A is the transmitting station, and station B is the receiving station. Each station has two unique frequency bands for transmission and receiving reservations:

A: F1 for transmission
F2 for receiving reservation
B: F3 for transmission
F4 for receiving reservation

Assume both stations have tunable transmitter and receiver, and know each other's reservation channel's frequency initially.

First, station A tunes its transmitter to F4, B's reservation channel, to notify station B the pending frame transmission and its transmission channel F1. After the reservation, station A transmits the frames on F1, and station B tunes its receiver to F1 to receive the frames. The procedure is straightforward except for the fact that contention can arise in the reservation channels when two or more stations wish to transmit to the same station. A medium access control protocol such as slotted Aloha is thus required in the reservation channel.

20. Compare FDMA, TDMA, and CDMA in terms of their ability to handle groups of stations that produce information flows that are produced at constant but different bit rates.

Solution:

Suppose the total bit rate supported by the transmission medium is R. In the typical FDMA, each station can transmit at a rate of R/M on its assigned frequency, where M is the total number of stations. The bit rate R/M is fixed for each station, and must satisfy the highest bit rate generated by the group. But for stations with lower bit rate, unused bandwidth is wasted. FDMA is inflexible and inefficient in handling flows with different bit rates. FDMA would need to be modified to allocate bands of different bandwidth to different users to accommodate differences in bit rate requirements.

In TDMA, each time slot gives an average bit rate of R/M, where M is the number of time slots. TDMA is more flexible than FDMA, because it can accommodate flows of different bit rates in two ways: assign multiple slots to each flow according to their bit rate, or allow the slot to be variable in duration. In the first method, the bit rate of the flow must be some multiple of a basic bit rate. In the second method, a means of identifying the endpoints of the variable length frame must be provided, and this adds overhead. TDMA is more flexible than FDMA in handling flows of various bit rates, but does not necessarily does this efficiently.

Unlike TDMA and FDMA where transmission bandwidth is statically allocated to different stations in terms of time or frequency, in CDMA, the signal of every flow occupies the entire frequency band at the same time. In general CDMA assumes that the bit rate of the information source is fixed, and the symbol rate of the spreading sequence is an integer multiple of the information bit rate. If the bit rate is reduced by some integer factor, for example by 2, then the integration period for each bit time (in Figure 6.33) is twice as long. This means that the signal-to-noise ratio at the receiver will be higher, and hence the transmitter can reduce its transmitted power. The capacity of the CDMA system is limited by the interference between different flows. Thus flows of lower bit rate require lower transmission power, and cause lower interference to other flows. The extra capacity can be used to handle more flows. CDMA can efficiently handle flows of different bit rates as long as the inter-flow interference is below an acceptable level.

21. Calculate the autocorrelation function of the pseudorandom sequence in Figure 6.30 as follows. Replace each 0 by –1 and each 1 by +1. Take the output sequence of the generator and shift it with respect to itself; take the product of seven (one period) symbol pairs and add. Repeat this calculation for shift values of 0, 1,…, 7. In what sense does the result approximate the autocorrelation of a random sequence?

Solution:

Autocorrelation $R(n) = \sum_{i}^{m} A0_i \, An_i$

where n = 0,..., 7 is the shift value, A0 is the original sequence, An is the sequence after n shifts, and m is the number of bits in the pseudorandom sequence.

Shift to the right every time:

	1	2	3	4	5	6	7	
A0:	+1 (1)	+1 (1)	+1 (1)	-1 (0)	+1 (1)	-1 (0)	-1 (0)	R(0) = 7
A1:	-1	+1	+1	+1	-1	+1	-1	R(1) = -1
A2:	-1	-1	+1	+1	+1	-1	+1	R(2) = -1
A3:	+1	-1	-1	+1	+1	+1	-1	R(3) = -1
A4:	-1	+1	-1	-1	+1	+1	+1	R(4) = -1
A5:	+1	-1	+1	-1	-1	+1	+1	R(5) = -1
A6:	+1	+1	-1	+1	-1	-1	+1	R(6) = -1
A7:	+1	+1	+1	-1	+1	-1	-1	R(7) = 7

The normalized autocorrelation *R(n)/N*, where *N* = 7 is:

$$\frac{R(n)}{7} = \begin{cases} 1 & n = 0,7,14,\ldots \\ \frac{-1}{7} & otherwise \end{cases}$$

This is similar to the autocorrelation of a random sequence

$$R(n) = \begin{cases} 1 & n = 0 \\ 0 & otherwise \end{cases}$$

However, the pseudorandom sequence is not truly, that is, *R(n)* repeats itself after a period of *N* shifts. However as we go to longer pseudorandom sequences, i.e. *N* increases, *R(n)* approaches the autocorrelation function of a truly random sequence.

22. Construct the Walsh orthogonal sequences of length 16.

Solution:

$$W_{16} = \begin{bmatrix} W_8 & W_8 \\ W_8 & W_8^c \end{bmatrix}$$

$$W_8 = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$W_{16} = \left[\begin{array}{cccc|cccc|cccc|cccc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0\\
\hline
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0\\
0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0\\
0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0\\
0&0&1&1&0&0&1&1&1&1&0&0&1&1&0&0\\
0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1\\
\hline
0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0\\
0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1\\
0&0&1&1&1&1&0&0&1&1&0&0&0&0&1&1\\
0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0
\end{array}\right]$$

23. Decode the sum signal in Figure 6.33 using the Walsh sequence for channel 4. What do you get? Explain why.

Solution:

$\mathbf{S_4}$ = (-1, 1, 1, -1)

Sum signals:
$\mathbf{r_1}$ = (+1, -1,-1,-3)
$\mathbf{r_2}$ = (-1, +1, -3, -1)
$\mathbf{r_3}$ = (+1, -1, +3, +1)

$\mathbf{S_4} * \mathbf{r_1} = -1 -1 -1 + 3 = 0$
$\mathbf{S_4} * \mathbf{r_2} = 1 + 1 -3 + 1 = 0$
$\mathbf{S_4} * \mathbf{r_3} = -1 -1 +3 -1 = 0$

We obtain 0 for every information bit. This is because no channel 4 signal is present in the sum signal, and the Walsh sequence of channel 4 is orthogonal to other channel signals.

24. Compare IS-54 and GSM in terms of their handling of speech and the effect on spectrum efficiency.

Solution:

GSM provides higher a bit rate (22.8 kbps) to a full-rate speech channel than IS-54 which only provides 16.2 kbps. The high traffic channel bit rate decreases the spectrum efficiency of GSM.

However, much of the 22.8 kilobits are used in the error protection of the speech signal. This decreases the interference between adjacent cells and improves GSM's frequency reuse scheme. Compared to the frequency reuse factor of 7 of IS-54, GSM has a reuse factor of 3.

Both traffic channel bit rate and the frequency reuse factor affects the spectrum efficiency of the standards. In the end, GSM achieves a spectrum efficiency close to IS-54, but provides a higher speech quality.

25. Suppose that the A provider in the 800 MHz cellular band uses GSM and the B provider uses IS-95. Explain how a call from a mobile user in system B to a user in system A is accomplished.

Solution:

The GSM and IS-95 systems use different encodings for voice. At the very least, the formats between the two systems need to be converted in each direction. In practice the two systems are interconnected by the conventional telephone network, and so the signals from both systems are first converted to PCM from conventional telephony and then to the other format.

26. Suppose that a 1 MHz channel can support a 1 Mbps transmission rate. The channel is to be shared by 10 stations. Each station receives frames with exponential interarrivals and rate λ =50 frames /second and frames are constant length L = 1000 bits. Compare the total frame delay of a system that uses FDMA to a system that uses TDMA.

Solution:

Given:

R = 1 Mbps
L = 1000 bits
M = 10 stations
$X = L/R$ = (1000 bits)/(1000000 bits/sec) = $1\text{x}10^{-3}$ sec

$\rho = (\lambda/M)(X \text{ x } M) = 50(10\text{x}10^{-3}) = 0.5$

For FDMA:

Total Packet Delay $= [\rho M/(2(1-\rho)) + M/2 + M]\, E[X]$
$= [(0.5)(10)/(2(1 - 0.5)) + 10/2 + 10][10^{-3}]$
$= [20]\,[10^{-3}] = 0.02$

For TDMA:

Total Packet Delay $= [\rho M/(2(1 - \rho)) + M/2 + 1]E[X]$
$= [(0.5)(10)/(2(1 - 0.5)) + 10/2 + 1]\,[10^{-3}]$
$= [11]\,[10^{-3}] = 0.011$

Both TDMA and FDMA are sensitive to the number of stations. We can observe that TDMA outperforms FDMA because of the faster frame transmission time.

27. Discuss how the delay and throughput performance of GPRS vary with the allocation in the number of access request channels, access grant channels, and data channels.

Solution:

Allocating more access request channels reduces the contention in each channel and results in lower delay and higher throughput. Allocating more access grant channels does not improve the performance assuming that the existing channel can service all the grants. Allocating more data channels provides more resources allowing more requests to be granted. This results in better delay and throughput performance.

28. Consider an "open concept" office where 64 carrels are organized in an 8 x 8 square array of 3m x 3m space per carrel with a 2m alley between office rows. Suppose that a conduit runs in the floor below each alley and provides the wiring for a LAN to each carrel.

Solutions follow questions:

a. Estimate the distance from each carrel to a wiring closet at the side of the square office.

Distance from carrel to wiring room in meters:

	1	2	3	4	5	6	7	8
1	15	18	21	24	27	30	33	36
2	15	18	21	24	27	30	33	36
3	8	11	14	17	20	23	26	29
4	8	11	14	17	20	23	26	29
5	8	11	14	17	20	23	26	29
6	8	11	14	17	20	23	26	29
7	15	18	21	24	27	30	33	36
8	15	18	21	24	27	30	33	36

b. Does it matter whether the LAN is token ring or Ethernet? Explain.

All distances to wiring room are less than 100m, well within the distance that can be handled by Ethernet.

A token ring LAN requires a wire to and from the wiring closet for each station. The length of the ring is then the sum of twice the sum of all distances from the stations to the closet: 2816 meters. It is clear that the token ring must deal with much greater distance than Ethernet.

c. Discuss the merits of using a wireless LAN in this setting.

A wireless LAN can obviously save a lot of wiring and therefore reduce the overall cost. In addition, the distances from the stations to the base will be shorter and more uniform. However the bandwidth in a wireless LAN will be shared among all stations. In a switched Ethernet, it is possible to reduce the amount of bandwidth sharing by introducing separate collision domains.

29. Suppose that a LAN is to provide each worker in problem 6.28 with the following capabilities: digital telephone service; H.261 video conferencing capability; 250 ms retrieval time for a 1 Mbyte file from servers in the wiring closet; 10 e-mails/hour sent and received by each worker (90 percent of e-mails are short, and 10 percent contain a 100-kilobyte attachment).

Solutions follow questions:

a. Estimate the bit rate requirements of the LAN.

Capability	Bit Rate
digital telephone service	64 kbps
H.261 videoconferencing, 2 x 64 kbps	128 kbps
large file retrieval, 8 Mb/250 ms =	32 Mbps
e-mails, $\frac{10 \times 0.1 \times 800\text{k}}{60\,\text{min/hr} \times 60\,\text{sec/min}} =$	222 bps
Total =	32+ Mbps

In this case, the fast file download requirement dominates the overall bit rate requirement. If one were to accommodate this "peak rate" requirement for all users simultaneously, then an aggregate peak bit rate of 64x32 Mbps= 2 Gbps would be required. This peak bit rate can be reduced by taking into account the probability that various numbers of users simultaneously download files.

Another consideration is the impact of the peak bit rate events on the quality of the digital voice service. Unless the digital voice is given some sort of priority access, periods of high peak rate traffic will result in long delays for voice packets and associated loss in delivered voice quality.

b. Is it worthwhile to assign the users to several different LANs and to interconnect these LANs with a bridge?

The bandwidth requirements are dominated by the fast retrieval time required for the extra large files. These retrievals are likely to be spurious in time, so consequently the bandwidth requirement for each user is extremely bursty. The digital telephone and videoconferencing applications are much lower in the bit rate requirement, but these are likely to be sustained over periods of minutes or even hours.

Suppose that 25% of the 64 users in the LAN are involved in either a digital telephone or a videoconference call. The required bit rate is then .25x64x128 kbps ≈ 2 Mbps. Thus the users can be kept in one LAN, as long as there is enough spare bandwidth to provide the rapid response required for the retrieval of the long files.

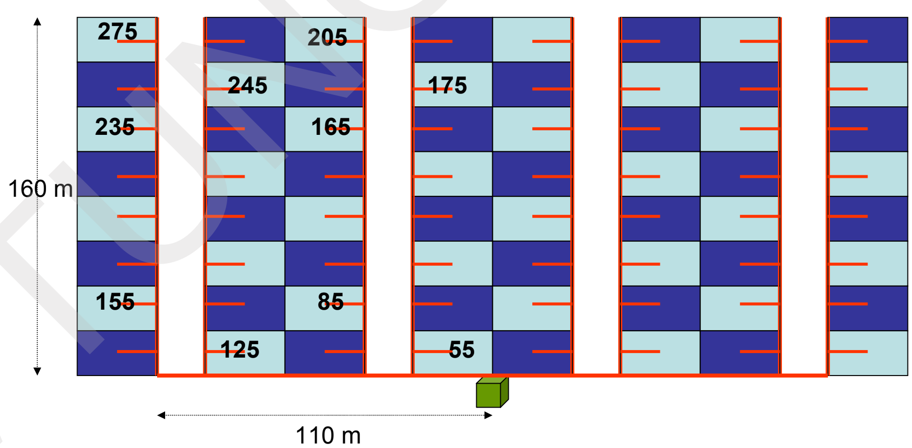
30. Consider a LAN that connects 64 homes arranged in rows of 8 homes on 20m x 30m lots on either side of a 10-meter-wide street. Suppose that an underground conduit on either side of the street connects the homes to a pedestal at the side of this rectangular array.

Solutions follow questions:

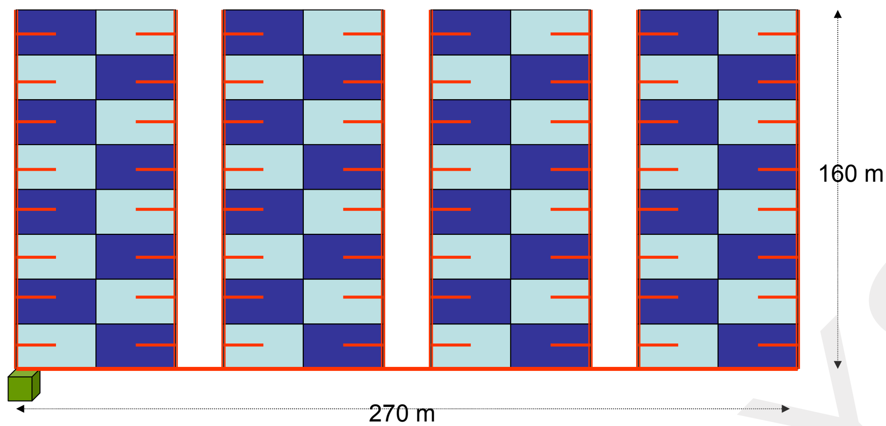
a. Estimate the distance from each home to the pedestal.

Distance from center of homes to pedestal in meters:

Placement 1:



Placement 2:



b. Estimate the bit rate requirements of the LAN. Assume two telephone lines, three MPEG2 televisions, and intense peak-hour Web browsing, say two Web page retrievals per minute, at an average of 20 kilobytes/page.

Web Pages: 20 Kb/page x (2 pages/min) x (1 min/60 sec) = 5.333 kbps
MPEG2: (6 Mbps) x (3) = 18 Mbps
Telephone: (64 Kbps) x (2) = 128 kbps

The total bit rate should be higher than (18 Mbps) x 64 homes = 1,152 Mbps

The bit rate is dominated by the MPEG2 streams. Let's examine the web page download requirement more closely. The average total bit rate for the 64 homes is 341 kbps which is quite low.

The problem does not consider the impact of peer-to-peer file sharing. Suppose one quarter of the homes are downloading 1 Mbyte files at the same time and that we would like each file to be downloaded in 1 minute. The required bit rate is: $64 \times ¼ \times 8 \times 10^6/60 = 2$ Mbps.

c. Can a single LAN meet the service requirements of the group of 64 homes? Explain.

The average bit rate required for each home is about 20 Mbps. This bit rate can be met with a 100 Mbps Ethernet link that connects each home to an Ethernet switch at the pedestal, although some sort of priority mechanism is required for the voice traffic. The switch however needs to be capable of handling more than 1 Gbps aggregate throughput. This can be met by providing several 1 Gbps links into the network backbone.

31. Use HDLC and Ethernet to identify three similarities and three differences between medium access control and data link control protocols. Is HDLC operating as a LAN when it is used in normal response mode and multipoint configuration?

Solution:

Similarities: Both medium access control and data link protocols deal with the transfer of blocks of information across a "single hop" which can consist of a point-to-point line, a multipoint line, or a shared medium.

The frames in HDLC and Ethernet contain framing information that delineates the beginning and end of each frame. HDLC inserts flag bits, and Ethernet inserts a preamble to delineate their frames.

Both data link control and MAC protocols check the CRC in the received frames for errors. Frames with errors are discarded.

Differences: The main functionality of the MAC protocols is to coordinate the access of multiple nodes to a shared transmission medium. To do so, the MAC protocols implement contention resolution procedures.

The data link layer protocols implement error control and flow control functions to provide reliable transmission. The MAC protocols do not implement these functions.

The data link layer protocols such as HDLC are concerned with providing connection-oriented, connectionless, and acknowledged connectionless packet transfer services to the network layer. The MAC protocols only provide connectionless packet transfer.

The data link control protocol often handles packet transfer only over a point-to-point link between two nodes, where medium access control is not required. However, the multipoint-to-point configuration is an exception. When HDLC operates in normal response mode, it in fact is implementing polling to control access to the medium.

32. An application requires the transfer of network layer packets between clients and servers in the same LAN. Explain how reliable connection-oriented service can be provided over an Ethernet LAN. Sketch a diagram that shows the relationship between the PDUs at the various layers that are involved in the transfer.

Solution:

A connection-oriented service has to be built on top of the connectionless service provided by an Ethernet LAN. The IEEE Logical Link Control layer can implement an HDLC-like error control protocol to provide this service. The relationship between layers is shown in Figure 6.48 in the text. The relationship between SDUs and PDUs can be visualized using Figure 6.50: the network layer SDU is passed to the LLC which prepares a PDU suitable for error control. The LLC PDU is then passed to the Ethernet layer.

33. Calculate the difference in header overhead between a DIX Ethernet frame and an IEEE 802.3 frame with SNAP encapsulation.

Solution:

DIX Ethernet frame:

7	1	6	6	2		4
preamble	SD	dest. addr.	src. addr.	type	data	FCS

header overhead = 7 + 1 + 6 + 6 + 2 + 4 = 26 bytes.

IEEE 802.3 frame with SNAP encapsulation:

22	1	1	1	5		4
MAC header	DSAP	SSAP	ctrl	SNAP header	data	FCS

header overhead = 22 + 3 + 5 + 4 = 34 bytes.

34. Suppose that a group of 10 stations is serviced by an Ethernet LAN. How much bandwidth is available to each station if (a) the 10 stations are connected to a 10 Mbps Ethernet hub; (b) the 10 stations are connected to a 100 Mbps Ethernet hub; (c) the 10 stations are connected to a 10 Mbps Ethernet switch.

Solution:

a. Assuming essentially 100% efficiency, the 10 Mbps are shared equally by the 10 stations, so each station can receive a maximum of 1 Mbps on average.

b. Assuming essentially 100% efficiency, the 100 Mbps are shared equally by the 10 stations, so each station receives a maximum of 10 Mbps on average.

c. The bit rate available to each station depends on the number of collision domains that are configured in the switch. In the best case, each station has nearly 10 Mbps to the Ethernet switch. Each station will have full access to the 10 Mbps if the switch capacity can handle the aggregate rate from all the stations.

35. Suppose that an Ethernet LAN is used to meet the requirements of the office in problem 3.28.

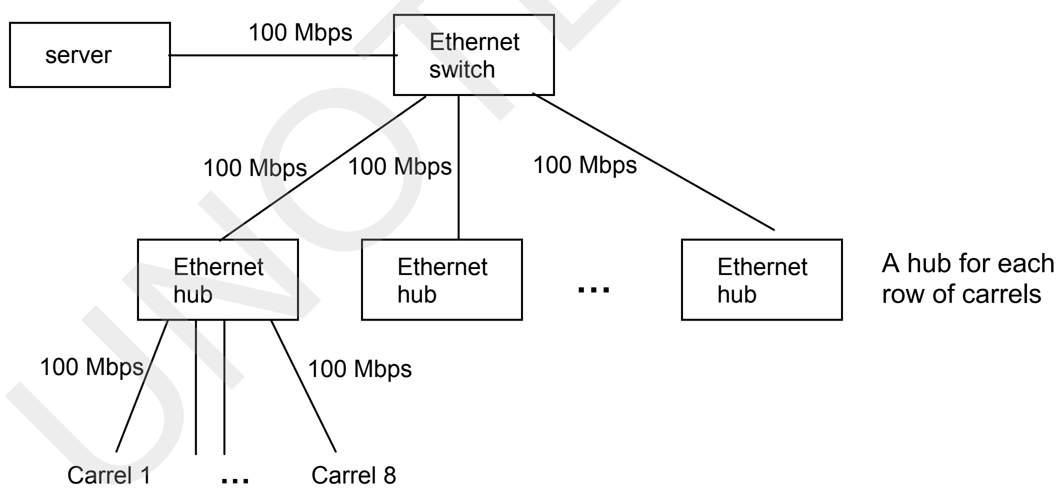
Solutions follow questions:

a. Can the requirements of one row of carrels be met by a 10 Mbps Ethernet hub? By a 10 Mbps Ethernet switch?

No. The bit requirement for each worker is 33.6 Mbps (for example, a 1 Mbyte file retrieved in 250 ms). The maximum rates of both Ethernet hub and switch are both 10 Mbps. At 10 Mbps, the file transfer would take 800 ms.

b. Can the requirements of the office be met by a hierarchical arrangement of Ethernet switches as shown in Figure 6.57?

For the arrangement shown below, each group of 8 users in a row shares 100 Mbps. This would allow up to three users to be downloading a file and still meet the 250 ms file download requirement.



36. Suppose that 80 percent of the traffic generated in the LAN is for stations in the LAN, and 20 percent is for stations outside the LAN. Is an Ethernet Hub preferable to an Ethernet switch? Does the answer change if the percentages are reversed?

Solution:

The difference between a hub and a switch is that in the hub frame are broadcast to all lines, while in a switch, frames are forwarded to another collision domain only if the destination is in that domain.

When 80% of the traffic is local, the switch will only forward 20% of frames to other collision domains, thus increasing the bandwidth available on those domains. If 80% of the traffic is to other collision domains, then the switch forward more traffic to other domains. If there is only one other domain, then the switch forwards almost as much traffic as a hub would and is thus ineffective in containing broadcast traffic. On the other hand, if the switch has multiple broadcast domains, then the amount of traffic forwarded from the switch will be less than that forwarded by a hub.

37. Calculate the parameter *a* and the maximum throughput for a Gigabit Ethernet hub with stations at a 100-meter distance and average frame size of 512 bytes; 1500 bytes; and 64,000 bytes.

d = 100
T_{prop} = 0.0000005
R = 1.00E+09

	512	1500	64000
a	0.12207	0.041667	0.000977
Throughput	0.56	0.788525	0.993754

38. Provide a brief explanation for each of the following questions:

a. Under a light load, which LAN has a smaller delay: Ethernet or token ring?

Ethernet has smaller delay under a light load. In Ethernet under a light load, there is little or no contention for the channel, the delay incurred is close to the frame transmission time. In token ring, however, there is always the additional delay incurred from circulating the token around the ring.

b. Under a high load, which LAN has a smaller delay: Ethernet or token ring?

Token ring has smaller delay under a high load. In Ethernet there is more contention for the channel, much of the time is spent in collision and backoff, so on average the frames experience longer delay and higher delay variability. In comparison, token ring provides each station with an orderly and round-robin access to the channel by passing the token around. When the number of frames transmitted per token is limited, and frames are kept at a fixed length, token ring can guarantee a maximum delay for each station.

39. Suppose that a token-ring LAN is used to meet the requirements of the office in problem 6.29

a. Calculate the ring latency if all carrels are to be connected in a single ring as shown in Figure 6.58. Repeat for a ring for a single row of carrels.

M = 64
b = 2.5
d = 2816 m

$\tau' = (d/v) + M(b/R) = (2816/2\text{x}10^8) + 64(2.5/R) = 14 \text{ x } 10^{-6} + (160/R)$
If R = 16 Mbps
$\tau' = 14\text{x}10^{-6} + (160/32 \text{ x } 10^6) = 14 \text{ x } 10^{-6} + 5 \text{ x } 10^{-6} = 24 \text{ x } 10^{-6} = 24\ \mu\text{sec}$

If R = 32 Mbps
$\tau' = 14\text{x}10^{-6} + (160/32 \text{ x } 10^6) = 14 \text{ x } 10^{-6} + 5 \text{ x } 10^{-6} = 19 \text{ x } 10^{-6} = 19\ \mu\text{sec}$

M = 8
b = 2.5
d = 408 m

$\tau' = (d/v) + M(b/R) = (408/2 \times 10^8) + 8(2.5/R) = 2 \times 10^{-6} + (20/R)$
If R = 16 Mbps
$\tau' = 2 \times 10^{-6} + (20/32 \times 10^6) = 2 \times 10^{-6} + 0.6 \times 10^{-6} = 3.2 \times 10^{-6} = 3.2\ \mu$sec

R = 32 Mbps
$\tau' = 2 \times 10^{-6} + (20/32 \times 10^6) = 2 \times 10^{-6} + 0.6 \times 10^{-6} = 2.6 \times 10^{-6} = 2.6\ \mu$sec

b. Can the requirements of one row of carrels be met by a 16 Mbps token ring?

The file download transfer time of 250 ms cannot be met by a 16 Mbps token ring. At this speed the download time is 500 ms. On the other hand, the requirements of the other applications can be met by a 16 Mbps ring.

c. Can the requirements of the office be met by an FDDI ring?

For FDDI, R = 100 Mbps and so
$\tau' = 14 \times 10^{-6} + (160/100 \times 10^6) = 14 \times 10^{-6} + 1.6 \times 10^{-6} = 15.6 \times 10^{-6} = 15.6\ \mu$sec.

At R = 100 Mbps, up to three stations out of 64 can be busy downloading files while still meeting the 250 ms requirement. The other requirements of the office are met easily.

40. Suppose that a group of 32 stations is serviced by a token-ring LAN. For the following cases calculate the time it takes to transfer a frame using the three token reinsertion strategies: after completion of transmission; after return of token; after return of frame.

a. 1000 bit frame; 10 Mbps speed; 2.5-bit latency per adapter; 50 meters between stations.
b. Same as part (a) except 100 Mbps speed and 8-bit latency/adapter.
c. Same as part (a) except 1 km distance between stations.

Solution:

$M = 32$
$v = 2 \times 10^8$ m/sec
$L = 1000$ bits

I. Reinsertion after completion of transmission

Assume the token transmission time is small and can be ignored. The one-hop walk time is for passing the token to the next station.

X_{eff} = frame transmission time + one hop walk time

$= L/R + (d/v + b/R)$

a. R = 10 Mbps, b = 2.5 bits, d = 50 m
$X_{eff} = 100.5\ \mu$sec

b. R = 100 Mbps, b = 8 bits, d = 50 m
$X_{eff} = 10.3\ \mu$sec

c. R = 10 Mbps, b = 2.5 bits, d = 1 km
$X_{eff} = 105\ \mu$sec

II. Reinsertion after return of token

X_{eff} = max{ frame transmission time, ring latency} + one hop walk time
= max{ L/R, τ'} + (d/v + b/R)

$$\tau' = M\left\{\frac{d}{v} + \frac{b}{R}\right\}$$

a. X_{eff} = 101 μsec
b. X_{eff} = 11 μsec
c. X_{eff} = 173 μsec

III. Reinsertion after return of frame

X_{eff} = frame transmission time + ring latency + one hop walk time
= $L/R + \tau' + (d/v + b/R)$

a. X_{eff} = 117 μsec
b. X_{eff} = 21 μsec
c. X_{eff} = 273 μsec

41. Suppose that an FDDI LAN is used to meet the packet voice requirements of a set of users. Assume voice information uses 64 kbps coding and that each voice packet contains 20 ms worth of speech.

a. Assume that each station handles a single voice call and that stations are 100 meters apart. Suppose that the FDDI ring is required to transfer each voice packet within 10 ms. How many stations can the FDDI accommodate while meeting the transfer requirement?

packet length = 20 ms x 64 kbps = 1280 bits
header length = 28 x 8 = 224 bits
X_{frame} = (1280 + 224) bits/100 Mbps = 15.04 μsec
token length = 11 bytes x 8 = 88 bits
X_{token} = 88 bits / 100M = 0.88 μsec

d = 100 m between the stations
b = 10 bits
$v = 2 \times 10^8$ m/sec
ring length = 100m ≤ 200 km (upper limit)

Every station handles a single call, and each voice packet must be transmitted within 10ms. This means the target token rotation time (TTRT) must be within ½ x10ms = 5 ms.

TTRT = ring latency + M stations transmitting 1 busy token + 1 frame + 1 free token

$$= \left(\frac{100M}{2\times10^8} + \frac{10M}{100\times10^6}\right) + M\left(2\times0.88\mu + 15.04\mu\right)$$

$= M(0.6 + 1.76 + 15.04) \times 10^6 = 1.74 \times 10^{-5}M \le 0.005$

$M \le 287$

Therefore, the FDDI can accommodate maximum 287 stations.

b. How many simultaneous calls can be handled if each station is allowed to handle up to 8 calls?

In part (a), note that the first component of the TTRT (0.6 μsec) is due to propagation delay, the second component (1.76 μsec) is due to station latency, and the last component is the frame time (15.04 μsec). When a station handles multiple voice calls, we can imagine each station as sending 8 voice packets per token opportunity, so in effect the frame transfer time becomes 8 x 15.04 = 120.32 μsec. To keep the TTRT to less than 5 ms, we then require that

$M(0.6 + 1.76 + 120.32) < 5000$

The maximum number of stations is 40, and so the maximum number of calls is 8 x 40 = 320.

42. Use IEEE 802.3 and IEEE 802.11 to discuss three differences between wired and wireless LANs.

Solution:

Error rate: Unlike wired LANs, wireless LANs have high error rate due to interference and noise. Wireless LANs need to implement ARQ and/or error correction to increase the reliability of the communication channel.

Station mobility: Unlike wired LANs where stations connected to the LANs are static, in wireless LANs, the stations can be mobile and portable. Wireless LAN protocols may have to implement dynamic traffic routing and service handoff when the station moves from one service area to another.

Collision detection: Collision detection is not effective in wireless LANs due to the hidden station problem. Consequently, the sender must wait for explicit acknowledgment (e.g. RTS/CTS) from the receiver to know whether or not a frame has been received. The wireless LAN protocol implements a collision avoidance algorithm rather than the collision detection in wired LAN, and the delay in the contention period is longer than thee round-trip delay of $2t_{prop}$ of wired LAN because of waiting for the receiver's acknowledgment.

Other differences:

Security: In a wired LAN, the transmission medium is usually physically secure. In a wireless LAN, any device within the geographic transmission area can intercept the transmissions. To provide data security, wireless LANs need to implement encryption at the expense of higher cost and reduced performance.

Power consumption: Portable and mobile devices are usually battery powered, and thus have limited power capacity. The wireless LAN protocol must be designed to be power efficient.

All these issues are addressed in the IEEE 802.11 wireless LAN protocol.

43. For data packet radio networks, discuss the advantages and disadvantages of providing reliability by (a) implementing error correction at the physical layer, (b) implementing error control as part of the MAC layer, and (c) implementing error control at the LLC layer.

Solution:

Error correction at the physical layer uses up bandwidth in the form of check bits and adds complexity in terms of hardware for detecting and correcting errors. If the bit error rate is low, ARQ at a higher layer may be simpler and more efficient. However, if the bit error rate is high, error correction may become essential to be able to communicate at all over the radio medium.

Error control at the MAC layer is the preferred approach if the error rate is not too high. The number of check bits required for error detection is less than for error correction. Bandwidth is "wasted" only on retransmissions, not in a large number of check bits in every transmission. The implementation of ARQ is much simpler than complex error correction.

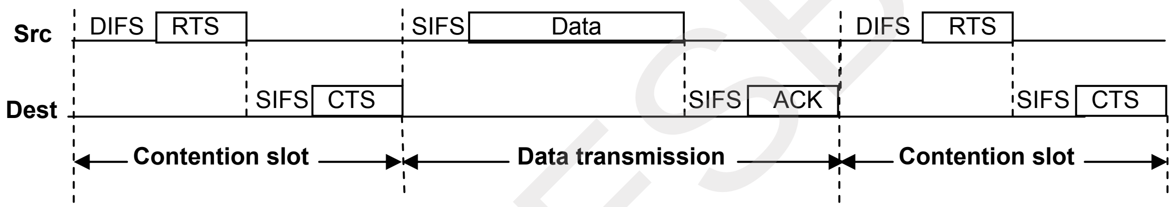
Error control at the LLC layer provides flexibility when operating over a MAC sublayer. The MAC sublayer can provide connectionless service and the LLC can add reliability for those network layers that require it.

44. Consider the distributed coordination function in IEEE 802.11. Suppose that all packet transmissions are preceded by a RTS-CTS handshake. Find the capacity of this protocol following the analysis used for CSMA-CD.

Solution:

In IEEE 802.11 DCF with RTS-CTS handshake, the sender stations contend for the channel by sending a RTS frame to the receiver, and it successfully captures the channel only when it receives a CTS frame from the receiver.

The sender does not know if it has succeeded until time = DIFS + X_{RTS} + SIFS + X_{CTS}. By this time, if no CTS frame arrives, the sender knows it has failed and will execute a backoff. This duration is similar to $2t_{prop}$ in CSMA-CD. Therefore, in CSMA-CA, the time can be divided in contention slots of size (DIFS + X_{RTS} + SIFS + X_{CTS}).



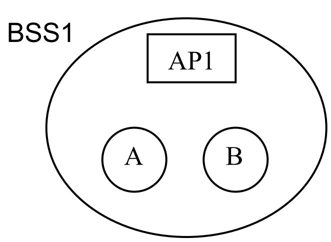
Similar to CSMA-CD, suppose all stations are contending for the channel and have a probability of *p* to transmit RTS during a contention slot. Then the maximum probability of success is *1/e* as the number of station increases, and the average # of contention slots in a contention period is *e*.

The maximum throughput occurs when all of the channel time is spent in transmission period followed by contention intervals.

$$\rho_{\max} = \frac{X_{data}}{X_{data} + 2SIFS + X_{ack} + e(DIFS + X_{rts} + SIFS + X_{cts})}$$

45. Suppose one station sends a frame to another station in an IEEE 802.11 ad hoc network. Sketch the data frame and the return ACK frame that are exchanged, showing the contents in the relevant fields in the headers.

Solution:



Suppose station A sends to station B without replaying through AP1.

Data Frame:

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control
	X_{data} + SIFS + X_{ack}	B	A	BSS1	—

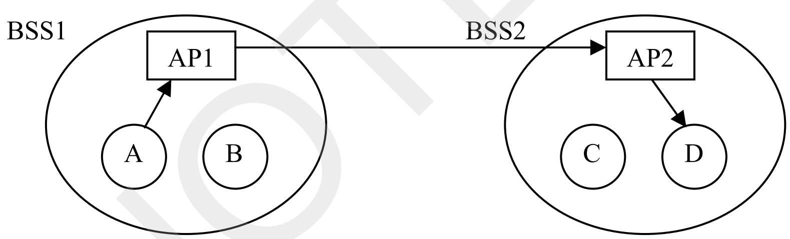
prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	data	data	0	0	0	—	—	—	—	—

ACK Frame:

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control
	0	A	B	BSS1	—

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	ctrl.	ack	0	0	0	—	—	—	—	—

46. Suppose one station sends a frame to another station in a different BSS in an IEEE 802.11 infrastructure network. Sketch the various data frames and ACK frames that are exchanged, showing the contents in the relevant fields in the headers.



Solution:

Suppose station A sends frame to station D. To exchange frames between BSS, the frames are relayed through the APs.

Data Frames:

1. A → AP1

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control
	X_{data} + SIFS + X_{ack}	BSS1	A	D	—

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	data	data	1	0	0	—	—	—	—	—

2. AP1 → AP2

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control	Addr. 4
	X_{data} + SIFS + X_{ack}	AP2	AP1	D	—	A

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	data	data	1	1	0	—	—	—	—	—

3. AP2 → D

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control
	X_{data} + SIFS + X_{ack}	D	BSS2	A	—

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	data	data	0	1	0	—	—	—	—	—

ACK Frames:

1. D → AP2

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control
	0	BSS2	D	A	—

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	ctrl	ack	1	0	0	—	—	—	—	—

2. AP2 → AP1

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control	Addr. 4
	0	AP1	AP2	A	—	D

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	ctrl	ack	1	1	0	—	—	—	—	—

3. AP1 → A

Frame control	Duration/ID	Addr. 1	Addr. 2	Addr. 3	Sequence control
	0	A	BSS1	D	—

prot. ver.	type	sub-type	To DS	From DS	More flag	Retry	Pwr mgt.	More data	WEP	Rsvd
00	ctrl	ack	0	1	0	—	—	—	—	—

47. Why is error control (ARQ and retransmission) included in the MAC layer in IEEE 802.11 and not in IEEE 802.3?

Solution:

Error control is required in the MAC in IEEE 802.11 because of the noise and interference in the wireless medium. In contrast, the wired medium for IEEE 802.3 protocol has very low error rates.

48. Consider the exchange of CSMA-CA frames shown in Figure 6.72 Assume the IEEE 802.11 LAN operates at 2 Mbps using frequency-hopping physical layer. Sketch a time diagram showing the frames transmitted including the final ACK frame. Show the appropriate interframe spacings and NAV values. Use Table 6.6 to obtain the appropriate time parameters. Assume that the data frame is 2000 bytes long.

Solution:

Assume the data frame includes the MAC header and the CRC field.

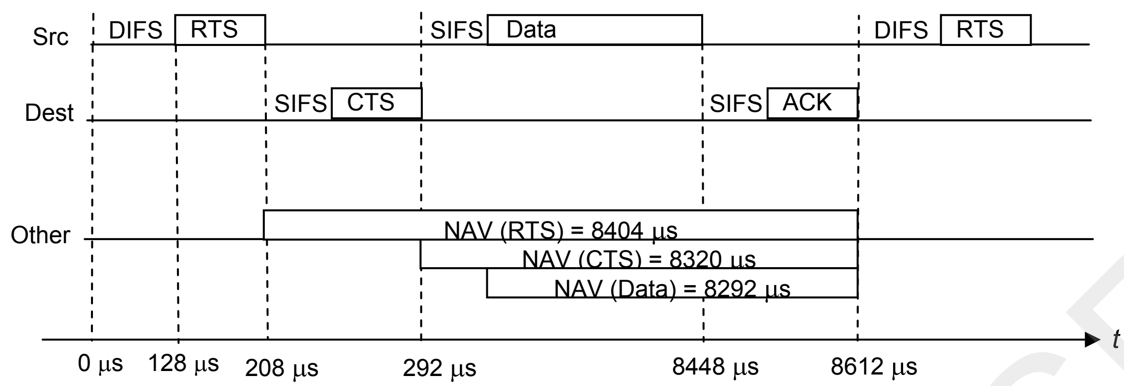
X_{frame} = (2000 x 8) / 2M = 8x10^{-3} sec

Data frame transmission time

X_{data} = X_{frame} + $X_{preamble}$ + $X_{PLCPheader}$ = 8 x 10^{-3} + (96 + 32) x 10^{-6} = 8.128x 10^{-3} sec
X_{rts} = (20 x 8) / 2M = 80 μsec
X_{cts} = (14 x 8) / 2M = 56μsec

The ACK frame consists of the MAC header (30 bytes) and the CRC field (4 bytes). Its total length is 34 bytes.

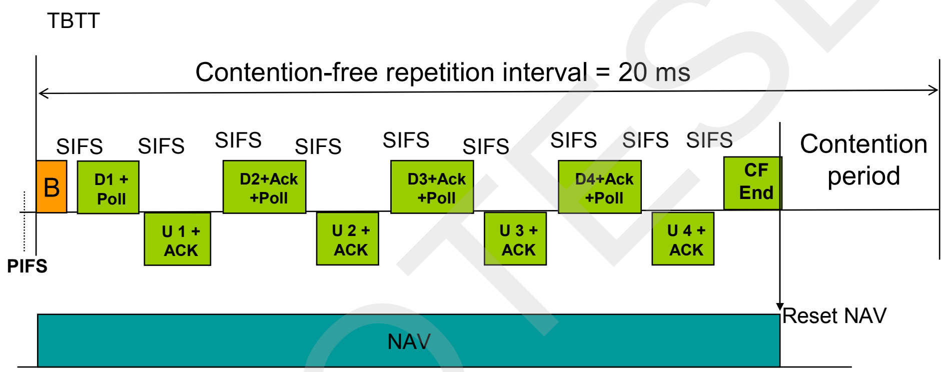
X_{ack} = (34 x 8) / 2M = 136 μs
slot time = 50 μs
SIFS = 28 μs
PIFS = SIFS + 1 slot time = 78 μs
DIFS = PIFS + 1 slot time = 128 μs



49. Suppose that four stations in an IEEE 802.11 infrastructure network are in a polling list. The stations transmit 20 ms voice frames produced by 64 kbps speech encoders. Suppose that the contention-free period is set to 20 ms. Sketch a point-coordination frame transfer with the appropriate values for interframe spacings, NAV, and data and ACK frames.

Solution:

The point-coordination frame transfer is shown in below.



D1, D2, D3, D4 = frames sent by point coordinator
U1, U2, U3, U4 = frames sent by polled stations
TBTT = target beacon transmission time
B = beacon frame

Every 20 ms, each voice source produces $64 \times 10^3 \times 20$ ms = 1280 bits, which forms the payload for each voice frame. This payload is carried in a MAC frame that includes 30 bytes of header and 4 bytes of CRC, for a total of 272 bits of overhead. Assuming the parameters used in Problem 6.48, the bit rate is 2 Mbps, and the frame transmission time is:

X_{frame}= (1280+272)/2 Mbps=776 μsec.

Assume that each frame sent by the point coordinator also contains a voice frame. In this case, frames from the point coordinator have the same duration as frames from the users. Each MAC frame is transmitted within a PLCP frame of total duration:

X_{data} = 776 + 96 + 32 μsec = 904 μsec.

Each SIF interval is 28 μsec, and the CF control frame and the Beacon frame are 34 bytes long and of duration

$X_{cf} = X_{beacon} = 34 \times 8/2\text{Mbps} = 136\ \mu\text{sec}.$

Therefore the duration of the contention free interval is:

$8 \times X_{data} + 9X_{sifs} + X_{cf} + X_{beacon} = 8 \times 904 + 9 \times 28 + 2 \times 136 = 7756\ \mu\text{sec}.$

50. Can a LAN bridge be used to provide the distribution service in an IEEE 802.11 extended service set? If so, explain how the service is provided and give an example of how the frames are transferred between BSSs.

Solution:

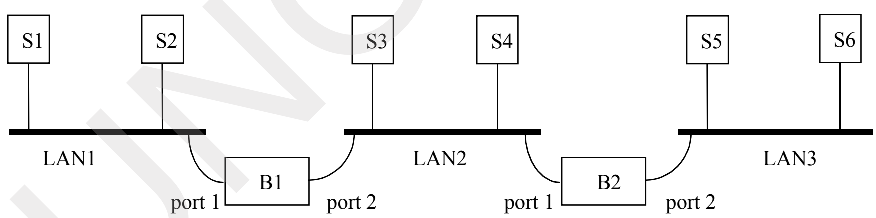
Yes, a LAN bridge can be used to provide the distribution service in an IEEE 802.11 LAN. A bridge is associated with each Access Point. The bridge learns the MAC addresses that should be forward on each port into the distribution system. The 802.11 frames emerging out of a BSS and onto a port must be encapsulated in a layer 2 tunnel and transferred to the input port of a bridge associated with the destination AP. The IEEE 802.11 frame is recovered at the destination AP and forwarded to the appropriate station.

51. Can a router be used to provide the distribution service in an IEEE 802.11 extended service set? If so, explain how addressing is handled and give an example of how the frames are transferred between BSSs.

Solution:

A router can be used to provide the distribution service in an IEEE 802.11 LAN. A bridge is associated with each Access Point. The bridge learns the MAC addresses that should be forward on each port into the distribution system. The 802.11 frames emerging out of a BSS and onto a port must be encapsulated in a layer 3 tunnel and transferred across an IP network to the input port of a bridge associated with the destination AP. The IEEE 802.11 frame is recovered at the destination AP and forwarded to the appropriate station.

52. Six stations (S1-S6) are connected to an extended LAN through transparent bridges (B1 and B2), as shown in the figure below. Initially, the forwarding tables are empty. Suppose the following stations transmit frames: S2 transmits to S1, S5 transmits to S4, S3 transmits to S5, S1 transmits to S2, and S6 transmits to S5. Fill in the forwarding tables with appropriate entries after the frames have been completely transmitted.



Station	Port

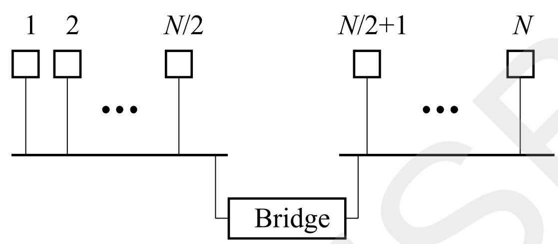
Station	Port

Solution:

Station	Port
S2	1
S5	2
S3	2
S1	1

Station	Port
S2	1
S5	2
S3	1
S1	1
S6	2

53. Suppose *N* stations are connected to an extended Ethernet LAN, as shown below, operating at the rate of 10 Mbps. Assume that the efficiency of each Ethernet is 80 percent. Also assume that each station transmits frames at the average rate of *R* bps, and each frame is equally likely to be destined to any station (including to itself). What is the maximum number of stations, *N*, that can be supported if *R* is equal to 100 kbps? If the bridge is replaced with a repeater, what is the maximum number of stations that can be supported? (Assume that the efficiency of the entire Ethernet is still 80 percent.)



Solution:

10 Mbps LAN
R = 100 kbps
efficiency = 0.80

Bridge:

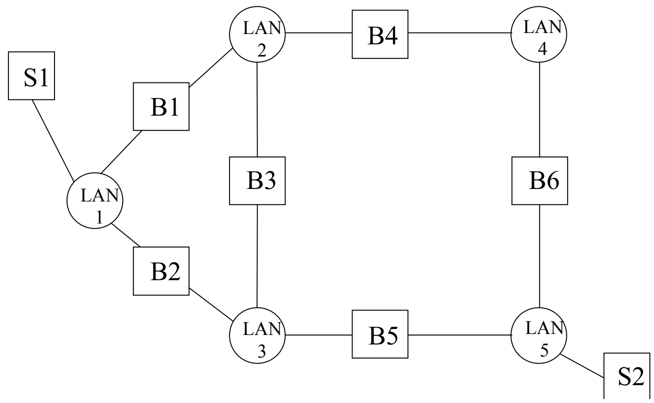
Each station is equally likely to transmit to any other stations in the extended LAN. Then *R*/2 traffic is local.

$$\frac{100kbps\,(\frac{1}{2})\,N}{10Mbps} \le 0.8 \qquad N \le 160$$

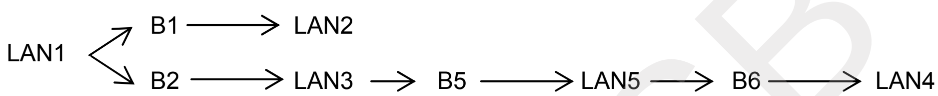
Repeater:

$$\frac{100kbps\,N}{10Mbps} \le 0.8 \qquad N \le 80$$

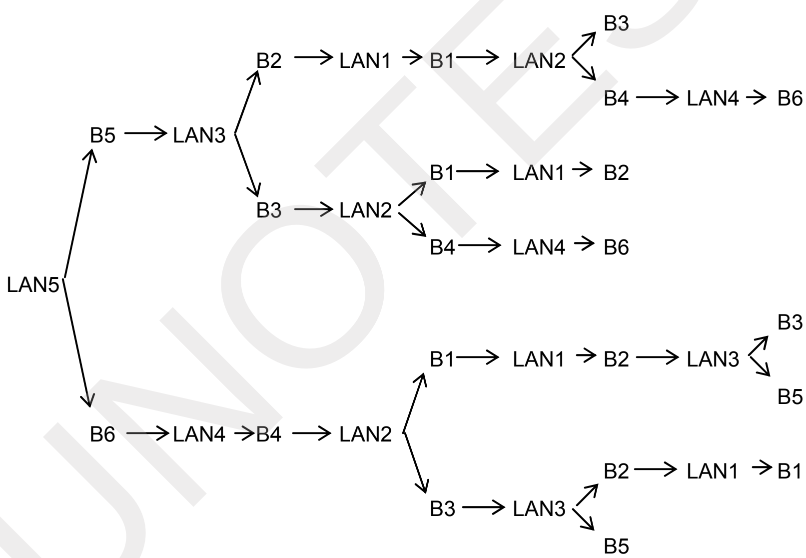
54. The LANs in the figure below are interconnected using source routing bridges. Assume that bridges 3 and 4 are not part of the initial spanning tree.



a. Show the paths of the single route broadcast frames when S1 wants to learn the route to S2.



b. Show the paths of all routes broadcast frames returned by S2.



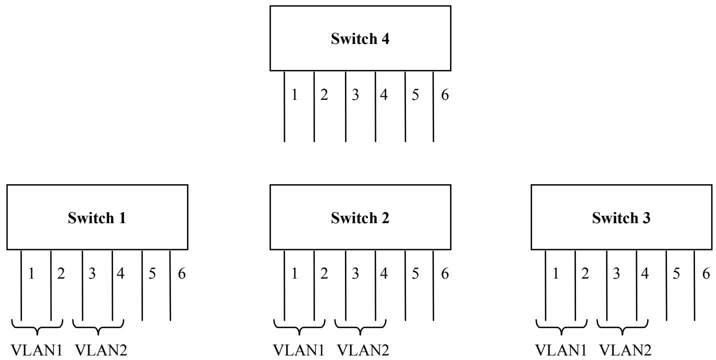
c. List all possible routes from S1 to S2 from part (b).

LAN1 → B2 → LAN3 → B5 → LAN5
LAN1 → B1 → LAN2 → B3 → LAN3 → B5 → LAN5
LAN1 → B1 → LAN2 → B4 → LAN4 → B6 → LAN5
LAN1 → B2 → LAN3 → B3 → LAN2 → B4 → LAN4 → B6 → LAN5

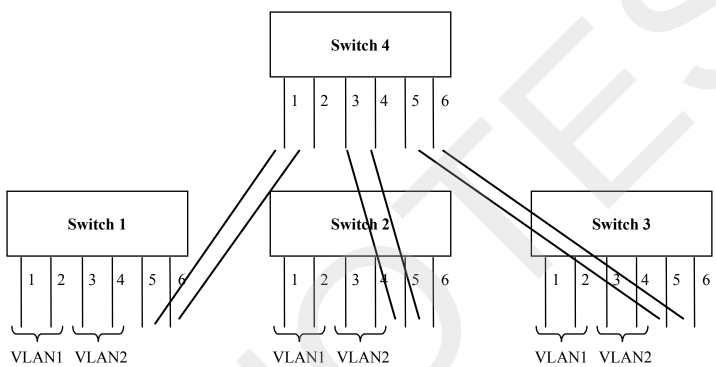
d. How many LAN frames are required to learn the possible routes?

Four frames.

55. Ports 1 through 4 in switches 1, 2, and 3, as shown in the figure below, provide port-based VLAN connectivity to stations attached to these ports. A network administrator needs to interconnect the other ports and configure the switches so that stations of the same VLAN connected to different switches can communicate. Use the association notation: switch a, VLAN n, port x, port y, …, port z, to indicate that ports x, y, …, z are associated to VLAN n in switch a, and the connectivity notation: (switch a, port x) to (switch b, port y) to indicate that port x in switch a is connected to port y in switch b. Show the association and connectivity for all the switches in the figure.



Solution:



Switch 1, VLAN 1, Port 1, Port 2, Port 5
Switch 1, VLAN 2, Port 3, Port 4, Port 6
Switch 2, VLAN 1, Port 1, Port 2, Port 5
Switch 2, VLAN 2, Port 3, Port 4, Port 6
Switch 3, VLAN 1, Port 1, Port 2, Port 5
Switch 3, VLAN 2, Port 3, Port 4, Port 6
Switch 4, VLAN 1, Port 1, Port 3, Port 5
Switch 4, VLAN 2, Port 2, Port 2, Port 6

Solutions to Chapter 7

7.1. Explain how a network that operates internally with virtual circuits can provide connectionless service. Comment on the delay performance of the service. Can you identify inefficiencies in this approach?

Solution:

The connection-oriented (co) network can present a network sublayer interface to the upper layer to make itself appear as a connectionless (cl) network. The upper layer sends packets in connectionless fashion, but the network resolves the packet destination's cl address into the corresponding co address (e.g. ATM address), and then establishes a virtual circuit between the source and destination. Any subsequent packets with the same source and destination are transmitted through this virtual circuit. The interface hides the internal operation of the network from the upper layer.

The cl-to-co address resolution is performed for the transmission of every packet, and hence incurs extra processing delay. Address caching can speed up the process, but cannot eliminate the delay. The first cl packet incurs extra delay because of the time required to set up the virtual circuit.

This cl-service-over-co-network approach is inefficient in that the QoS support of the connection-oriented network cannot be made available to the network layer.

ATM LAN emulation provides an example that involves providing connectionless service over a connection-oriented service.

7.2. Is it possible for a network to offer best-effort connection-oriented service? What features would such a service have, and how does it compare to best-effort connectionless service?

Solution:

Best-effort connection-oriented service would involve the transfer of packets along a pre-established path in a manner that does not provide mechanisms for dealing with the loss, corruption or misdelivery of packets. Best-effort connection-oriented service would require some means for establishing a path prior to the transfer of packets. Best-effort connectionless service would involve the transfer of packets in a datagram fashion, where routing decisions are made independently for each packet.

The path setup requirement makes connection-oriented service more complex than connectionless service. On the other hand, once a path is established, less processing is required to decide how a packet is to be forwarded. Connectionless service is more robust than connection-oriented service since connectionless service readily reroutes packets around a failure while VC service requires that new paths be established.

7.3. Suppose a service provider uses connectionless operation to run its network internally. Explain how the provider can offer customers reliable connection-oriented network service.

Solution:

To provide connection-oriented network service, an upper sublayer in the network layer at the edge of the network can establish logical connections across the connectionless network by setting up state information (for example, packet sequence number). The logical connection must be set up before packets can be transported, and each packet is assigned a sequence number. Using the sequence number, the upper sublayer entities can acknowledge received packets, determine and retransmit lost packets, delete duplicate packets, and rearrange out-of-order packets, hence providing reliable connection-oriented network service.

7.4. Where is complexity concentrated in a connection-oriented network? Where is it concentrated in a connectionless network?

Solution:

The complexity in connection-oriented networks revolves around the need to establish and maintain connections. Each node must implement the signaling required by the connection establishment process; each node must also maintain the state of the node in terms of connections already established and transmission resources available to accommodate new connections. End systems must be capable of exchanging signaling information with the network nodes to initiate and tear down connections. A connection oriented network must also include routing to select the paths for new connections.

Connectionless networks only require that nodes forward packets according to its routing tables. End systems only need to place network address information in the packet headers. The complexity of connectionless networks revolves around routing tables. Routing tables may be static and set up by a network administrator, or they may be dynamic and involve processing and exchange of link state information among nodes.

7.5. Comment on the following argument: Because they are so numerous, end systems should be simple and dirt cheap. Complexity should reside inside the network.

Solution:

This argument holds only if the computing resources are scarce and expensive as in the early days of computers. For example, telephone networks and terminal-oriented computer networks were designed based on this principle. Keeping the complexity inside the network is fine as long as the network operations are simple enough to keep the network manageable. But as the number of applications increases and the number of end systems grow, the complexity of the network increases drastically. In this case, keeping the complexity inside the network may not be scalable.

When the computing resources become plentiful and cheap, more applications and sophisticated communication services are demanded by the numerous end systems. This cannot be provided efficiently by one big monolithic network. Instead, by moving some of the complexity to the network edge and by utilizing the copious end-system computing resources, more and better network services can be deployed at a faster rate.

7.6. In this problem you compare your telephone demand behavior and your web demand behavior.

Solutions follow questions:

(a) Arrival rate: Estimate the number of calls you make in the busiest hour of the day; express this quantity in calls/minute. Service time: Estimate the average duration of a call in minutes. Find the load that is given by the product of arrival rate and service time. Multiply the load by 64 kbps to estimate your demand in bits/hour.

One possibility:

During the busiest hour of the day

5 calls/hour = 5 calls/hour

Average call duration = 7 minutes/call

5 x 7 = 35 minute/hour = 2100 sec/hour

Demand = 2100 sec/hour x 64 kbps = 134.4 Mbits/hour

(b) Arrival rate: Estimate the number of Web pages you request in the busiest hour of the day. Service time: Estimate the average length of a Web page. Estimate your demand in bits/hour.

One possibility:
During the busiest hour of the day:
60 Web pages / hour
Average Web page size = 20 kbytes
Demand = 20k x 8 x 60 = 9.6 Mbits/hour

Note that is web pages are heavy in terms of graphics content, say 200 kbytes/page, then the demand can approach that of telephone calls.

(c) Compare the number of call requests/hour to the number of web requests/hour. Comment on the connection setup capacity required if each web page request requires a connection setup. Comment on the amount of state information required to keep track of these connections.

The number of telephone call requests is much less than the number of Web requests per hour. The processing required to set up a call is much greater than that required for Web requests, especially because of the need to maintain billing records. On the other hand, Web requests are so much more frequent that its overall processing capability is much greater. Fortunately, the Web page request connection setup, for example, the TCP connection for each HTTP request, is implemented by the end systems in the network, a situation that is easier to scale than connection setups inside the network.

7.7. Apply the end-to-end argument to the question of how to control the delay jitter that is incurred in traversing a multi-hop network.

Solution:

Delay jitter is the variability in the packet delay experienced by a sequence of packets traversing the network. Many audio and video applications have a maximum tolerable bound on the delay jitter.

To control the delay jitter inside the network, every router must forward the packets within a certain time window by means of delaying the "early" packets or immediately forwarding the "late" packets. This requires significant amount of state information and processing at each router, and hence greatly complicates the network.

In the end-to-end approach, the applications at the end-systems have the knowledge of the tolerable range of jitter. A smoothing buffer can be used at the end system to collect the packets, and pass the packets to the application layer at a constant rate according to the application's specification. Consequently, the jitter is controlled while the network service remains simple.

7.8. Compare the operation of the layer 3 entities in the end systems and in the routers inside the network.

Solution:

All layer 3 (network layer) entities both in the end systems and inside the network perform packet routing. However, the layer 3 entities at the end systems are only concerned with transporting packets from one source to one destination (end-system to network or network to end-system). Routers perform both routing and forwarding of packets.

The layer 3 entities in the end systems in some networks regulate the input flow for congestion control. In networks with QoS support, they also police and shape input traffic according to the pre-negotiated contract between the end-system and the network. Layer 3 entities may also segment packets for the data link layer network, and assemble packets to pass to the upper layer at the end system.

The routers inside the network switch packets from multiple sources to multiple destinations. They perform traffic management functions (priority scheduling, buffer management, and shaping) to provide QoS to different traffic flows.

7.9. Consider a "fast" circuit-switching network in which the first packet in a sequence enters the network and triggers the setting up of a connection "on-the-fly" as it traces a path to its destination. Discuss the issues in implementing this approach in packet switching.

Solution:

There are two major issues. The first is the delay that the setup procedure introduces to the first packet. As a result, if the transmission of the rest of the packets starts after the first packet is sent, all these packets will be stuck behind the first packet as it sets the path and moves forward hop-by-hop. The proposed approach in fast circuit switching is to delay the packets for some time after the first packet is sent. The second issue is the possibility of failure in the path setup in a route which requires the choice an alternative route. If this happens all the packets behind the first packet will be lost because rerouting is done for the path setup packet and not the data packets following it.

7.10. Circuit switching requires that the resources allocated to a connection be released when the connection has ended. Compare the following two approaches to releasing resources: (a) use an explicit connection release procedure where the network resources are released upon termination of the connection and (b) use a time-out mechanism where the resources allocated to a connection are released if a "connection refresh" message is not received within a certain time.

Solution:

In the first approach once a connections has ended, a disconnect procedure is initiated during which control messages are sent to release the resources across the connection path. In this approach the resources are released as soon as the connection is ended. The cost associated with this approach is in the signaling that is required to terminate the connection and release the resources allocated to the connection.

In the second approach the resources are released if a connection refresh message is not received within a certain time. In this approach the refresh message needs to be sent periodically which consumes some of bandwidth. On the other hand, the connection management procedure is simplified by not having to explicitly release resources. Also, it takes some time after the connection has ended before the resources are released. And finally, there is the possibility that the refresh messages are lost during the connection, which results in the release of the resources in some sections of the route. This latter problem is addressed by setting the refresh timer to be a fraction of the connection reservation time to allow for the possibility of lost refresh messages.

7.11. The oversubscription ratio can be computed with a simple formula by assuming that each subscriber independently transmits at the peak rate of c when there is data to send and zero otherwise. If there are N subscribers and the average transmission rate of each subscribers is r, then the probability of k subscribers transmitting data at the peak rate simultaneously is given by

$$P_k = \binom{N}{k} (r/c)^k (1 - r/c)^{N-k}$$

The access multiplexer is said to be in the "overflow mode" if there are more than n subscribers transmitting data simultaneously, and the corresponding overflow probability is given by

$$Q_n = \sum_{k=n+1}^{N} \binom{N}{k} (r/c)^k (1 - r/c)^{N-k}$$

If a certain overflow probability is acceptable, then n is the smallest value that satisfy the given overflow probability. Find the oversubscription ratio if $N=50$, $r/c=0.01$, and $Q_n < 0.01$.

Solution:

$$Q_n = 1 - \sum_{k=0}^{n} \binom{N}{k} (r/c)^k (1 - r/c)^{N-k} \leq 0.01$$

$$Q_n = 1 - \sum_{k=0}^{n} \binom{50}{k} (.01)^k (.99)^{50-k} \leq 0.01$$

$$Q_n = 1 - (.99^{50}) \sum_{k=0}^{n} \binom{50}{k} (.01/.99)^k \leq 0.01$$

$n = 3$ → Q_n = 1 – (0.605006067) 1.65023771 = 1 – 0.998403827 = 0.001596173

The table below shows Q_n as a function of n. Thus if we have 50 users that are busy only 1% of the time, then the probability that more than 3 subscribers are busy is 0.0016. The probability that 2 or more subscribers are busy is 0.0148.

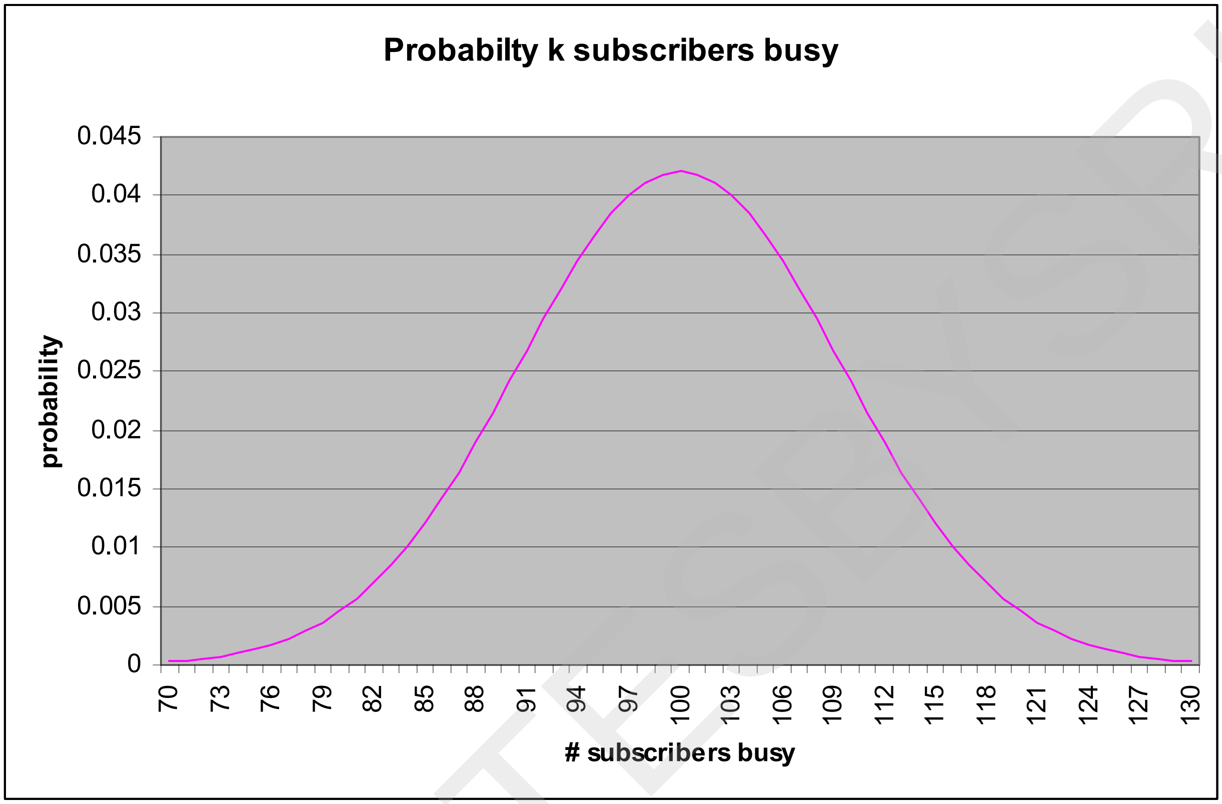
	N		
	50		
	p		
	0.01		Q_n
k		0	1
0	0.605006	0.605006	0.394994
1	0.305559	0.910565	0.089435
2	0.075618	0.986183	0.013817
3	0.012221	0.998404	0.001596
4	0.00145	0.999854	0.000146
5	0.000135	0.999989	1.09E-05
6	1.02E-05	0.999999	6.85E-07
7	6.48E-07	1	3.69E-08
8	3.52E-08	1	1.73E-09
9	1.66E-09	1	7.13E-11
10	6.87E-11	1	2.61E-12
11	2.52E-12	1	8.63E-14
12	8.29E-14	1	3.44E-15
13	2.45E-15	1	0

7.12. An access multiplexer serves $N = 1000$ subscribers, each of which has an activity factor $r/c = 0.1$. What is the oversubscription ratio if an overflow probability of 1 percent is acceptable? If $N\,p(1-p) >> 1$, you may use an approximation technique (called the DeMoivre-Laplace Theorem), which is given by

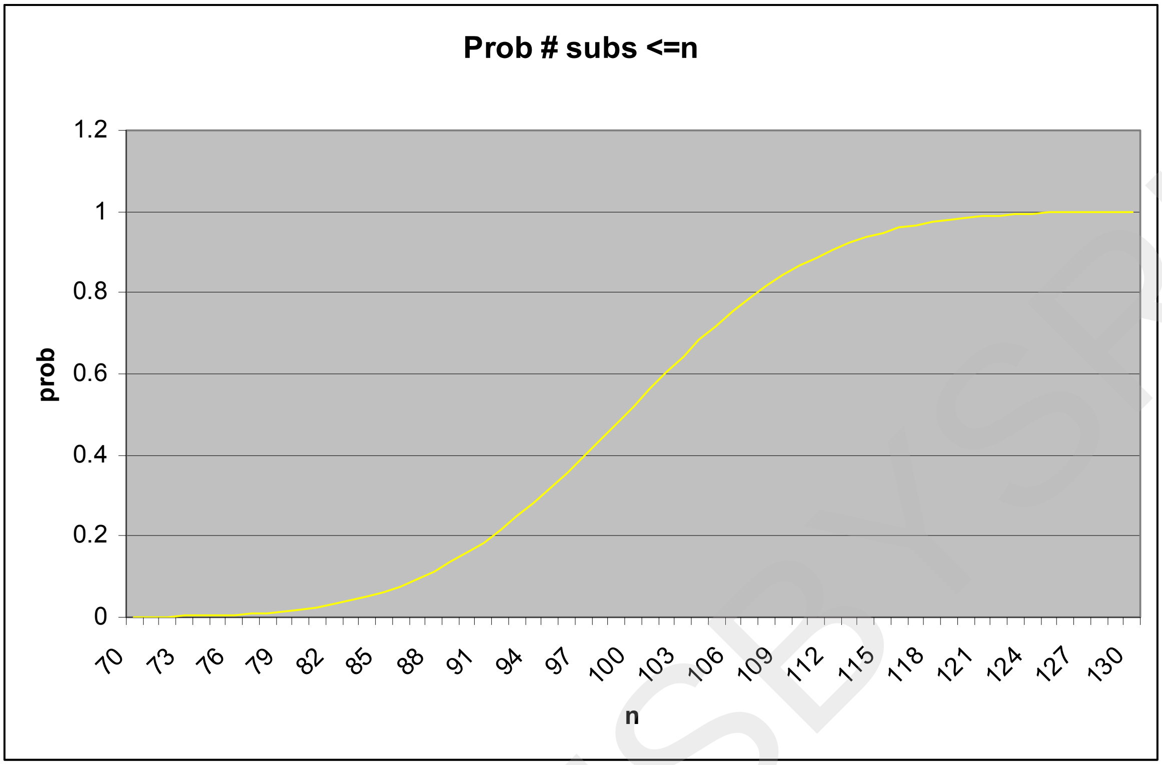
$$P_k = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{1}{\sqrt{2\pi Np(1-p)}} e^{-\frac{(k-Np)^2}{2Np(1-p)}}.$$

Solution:

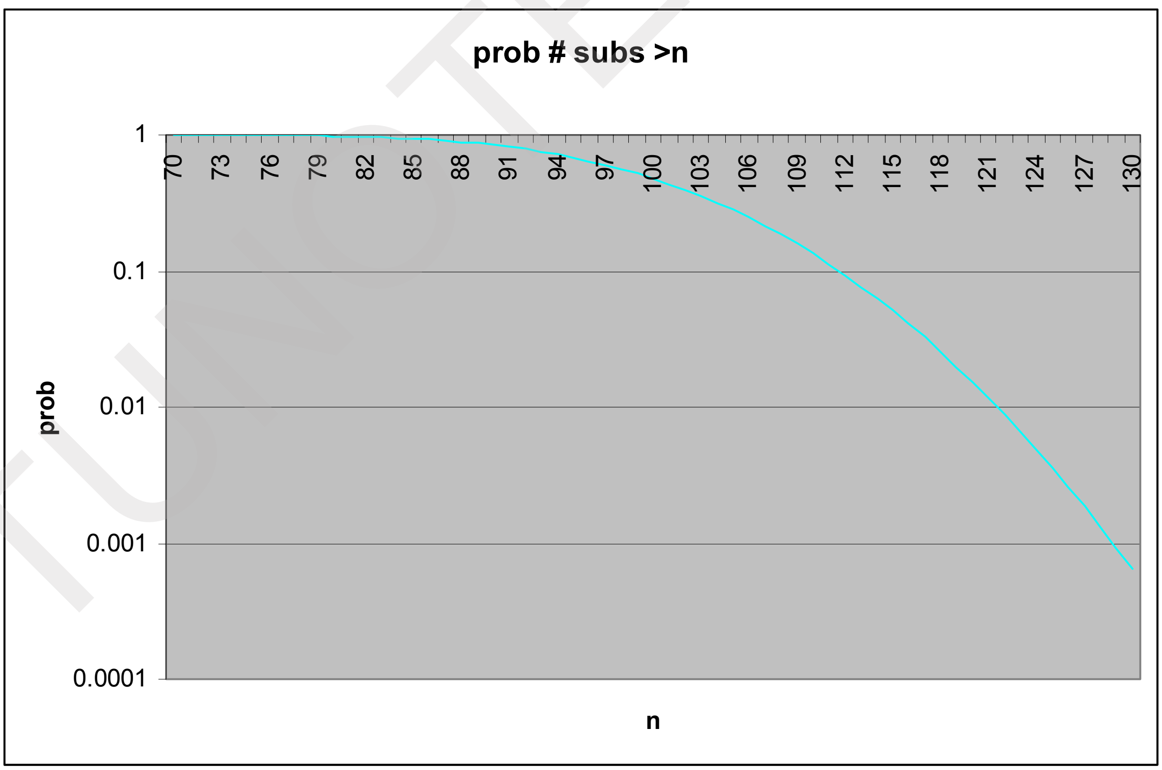
The average number of busy subscribers is *Np* = 1000*0.1 = 100. The graph below shows the probabilities in the vicinity of *k* = 100 using the above approximation:



By adding the probabilities up to a certain number of subscribers, say *n*, we can find the probability that the number of subscribers is less than or equal to *n*. The graph below shows the result.



The following plots the probability that more than n subscribers are busy in a logarithmic scale.



We see that the probability of exceeding 122 is about 1%.

7.13. In Figure 7.5 trace the transmission of IP packets from when a web page request is made to when the web page is received. Identify the components of the end-to-end delay.

Solutions follow questions:

(a) Assume that the browser is on a computer that is in the same departmental LAN as the server.

(Assume TCP connection to server has been established)
Creation of IP packet at workstation to carry Web page request
Frame transfer delay in the LAN
Receipt of IP packet at server
Processing of web page request
Creation of IP packet(s) to return Web page
Transfer of frame(s) across the LAN
Receipt of IP packet(s) at workstation

(b) Assume that the web server is in the central organization servers.

(Assume TCP connection to server has been established)
Creation of IP packet at workstation to carry Web page request
Frame transfer delay in the LAN
Receipt of IP packet at departmental router
Routing and forwarding of IP packet across campus backbone
Receipt of IP packet at server farm router
Routing and forwarding of IP packet to Web server
Processing of web page request
Creation of IP packet(s) to return Web page
Transfer of frame(s) across the server LAN to server farm router
Receipt of IP packet(s) at server router
Routing and forwarding of IP packet(s) across campus backbone
Receipt of IP packet(s) at departmental router
Routing and forwarding of IP packet(s) to workstation
Transfer of frame(s) across departmental LAN
Receipt of IP packet(s) at workstation

(c) Assume that the server is located in a remote network.

(Assume TCP connection to server has been established)
Creation of IP packet at workstation to carry Web page request
Frame transfer delay in the LAN
Receipt of IP packet at departmental router
Routing and forwarding of IP packet across campus backbone
Receipt of IP packet at gateway router
Routing and forwarding of IP packet across Internet to Web server
Processing of web page request
Creation of IP packet(s) to return Web page
Transfer of frame(s) across the Internet to gateway router
Receipt of IP packet(s) at gateway router
Routing and forwarding of IP packet across campus backbone
Receipt of IP packet(s) at departmental router
Routing and forwarding of IP packet(s) to workstation
Transfer of frame(s) across departmental LAN
Receipt of IP packet(s) at workstation

7.14. In Figure 7.5 trace the transmission of IP packets between two personal computers running an IP telephony application. Identify the components of the end-to-end delay.

Solutions follow questions:

(a) Assume that the two PCs are in the same departmental LAN.

IP packets flow simultaneously in both directions.
Consider the flow from PC1 to PC2:

Processing of sound to create digital signal
Creation of IP packet at PC1 to carry a voice packet
Frame transfer delay in the LAN
Receipt of IP packet at PC2
Processing of digitized signal to produce sound

The sources of delays are in the processing of the voice signal and in the transfer of frames across the LAN.

(b) Assume that the PCs are in different domains.

IP packets flow simultaneously in both directions.
Consider the flow from PC1 to PC2:

Processing of sound to create digital signal
Creation of IP packet at PC1 to carry a voice packet
Frame transfer delay in LAN1
Receipt of IP packet at departmental router
Routing and forwarding of IP packet across campus backbone
Receipt of IP packet at gateway router
Routing and forwarding of IP packet across Internet to PC2
Receipt of IP packet at PC2
Processing of digitized signal to produce sound

The sources of delays are in the processing of the voice signal, in the transfer of frames across LAN, and most significantly in the transfer across multiple routers in the Internet. The latter component can introduce large and variable delays.

7.15. In Figure 7.5 suppose that a workstation becomes faulty and begins sending LAN frames with the broadcast address. What stations are affected by this broadcast storm? Explain why the use of broadcast packets is discouraged in IP.

Solution:

All workstations in the extended (departmental) LAN are affected by the broadcast storm. Each LAN interface recognizes the frame as a broadcast frame and hence is compelled to process it. If a workstation sends out IP packets with broadcast address, the routers will forward the packets beyond the departmental LAN, to the campus backbone network, wide area network, and eventually the broadcast storm would affect the entire network. Therefore, the IP routers usually have the broadcast option disabled.

7.16. Explain why the distance in hops from your ISP to a NAP is very important. What happens if a NAP becomes congested?

Solution:

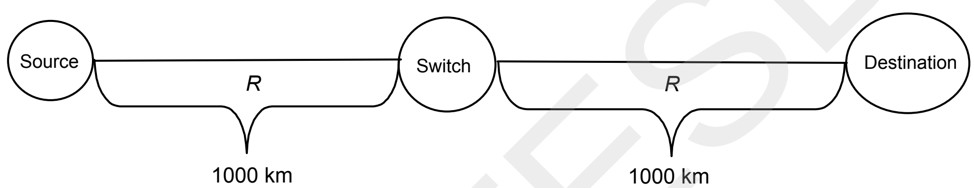
NAPs are the points where multiple networks interconnect to exchange IP traffic. Many IP transfers are therefore likely to involve traveling across a NAP. Thus the distance and hence the delay to other destinations is influenced strongly by the distance to a NAP.

If a NAP becomes congested, then the ISPs connected to it are limited in the traffic they can exchange. The ISPs have to route traffic through other NAPs and may cause further congestion elsewhere.

7.17. A 64-kilobyte message is to be transmitted over two hops in a network. The network limits packets to a maximum size of 2 kilobytes, and each packet has a 32-byte header. The transmission lines in the network are error free and have a speed of 50 Mbps. Each hop is 1000 km long. How long does it take to get the message from the source to the destination?

Solution:

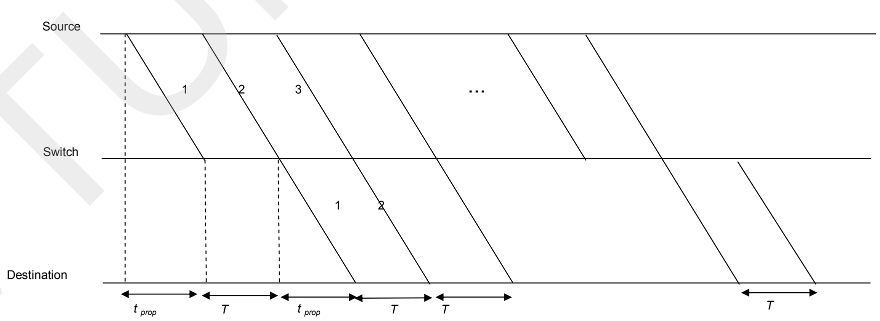
The situation is shown in the figure below.



First let's determine the number of packets required for the message. Given:

message size = 64 Kb
maximum packet size = 2 Kb, with a 32-byte header.
R = 50 Mbps
number of packets = 64 x 1024 / (2 x 1024 - 32) = 32.5 ≅ 33 packets.

Assume that the signal travels at the speed of light, or t_{prop} = 1000 km / (3 x 10^8 ms) = 3 1/3 ms. The transmission time of one packet is T = 2 Kb / 50 Mbps = 2 x 1024 x 8 / 50 x 10^6 = 327.68 μsec. The figure below shows the basic situation.

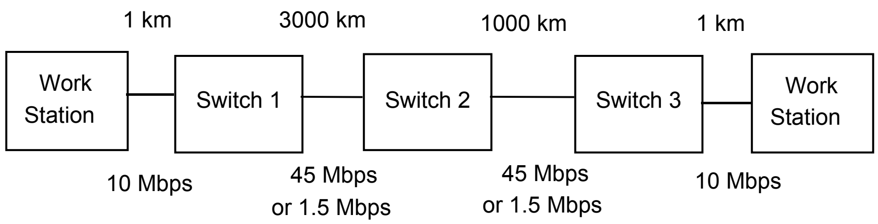


Assume that queueing and processing delays at the intermediate switch are negligible.

Total delay = 2 t_{prop} + T (number of packets +1)

= 2 (3 1/3 ms) + (327.68 μsec) (33 +1) = 17.8 ms

7.18. An audiovisual real-time application uses packet switching to transmit 32 kilobit/second speech and 64 kilobit/second video over the following network connection.



Two choices of packet length are being considered: In option 1 a packet contains 10 milliseconds of speech and audio information; in option 2 a packet contains 100 milliseconds of speech and audio information. Each packet has a 40-byte header.

Solutions follow questions:

(a) For each option find out what percentage of each packet is header overhead.

Option 1:

10 ms of speech gives 32 x 10 bits of information
10 ms of audio gives 64 x 10 bits of information
40-byte header produces 40 x 8 bits of header

Therefore, the packet size is 1280 bits, with an overhead of 25%.

Option 2:

100 ms of speech gives 32 x 10^2 bits of information
100 ms of audio gives 64 x 10^2 bits of information
40-byte header produces 40 x 8 bits of header

Therefore, the packet size is 9920 bits, with an overhead of 3.225%.

(b) Draw a time diagram and identify all the components of the end-to-end delay. Keep in mind that a packet cannot be sent until it has been filled and that a packet cannot be relayed until it is completely received (that is, store and forward). Assume that bit errors are negligible.

The end-to-end delay contains the following components: 1. a packetization delay P_s consisting of the time it take a source to generate enough information to fill a packet; the first byte in the packet is delayed by this amount: 2. queueing delay, frame transmission delay, and propagation delay at each hop.

We assume that a message is the same as a packet and that

$$\text{delay} = P_s + \underbrace{P_1 + P_2 + P_3 + P_4}_{\text{Propagation delays}} + \underbrace{T_1 + T_2 + T_3 + T_4}_{\text{Transmission through four hops}}$$

We sketch these delays in the following part.

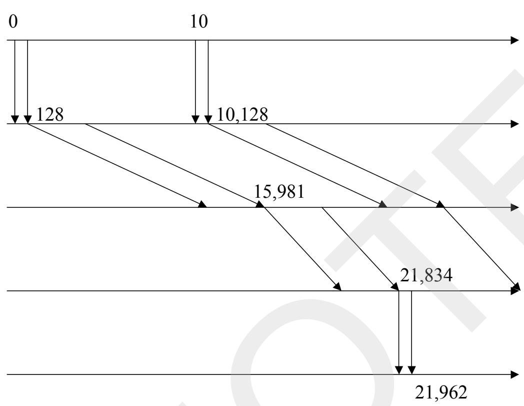
(c) Evaluate all the delay components for which you have been given sufficient information. Consider both choices of packet length. Assume that the signal propagates at a speed of 1 km /5 microseconds. Consider two cases of backbone network speed: 45 Mbps and 1.5 Mbps. Summarize your result for the four possible cases in a table with four entries.

Transmission Delays:

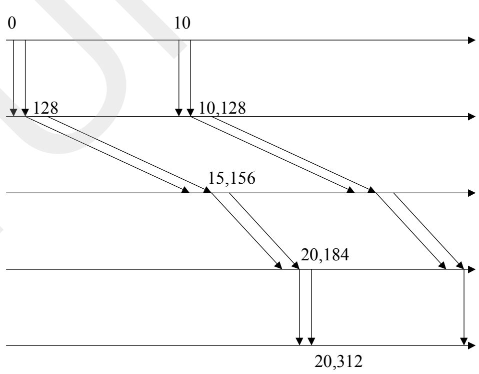
Bit Rate	1280 bits	9920 bits
1.5×10^6	853 μsec	6613 μsec
45×10^6	28,4 μsec	220.4 μsec

Propagation delays:
$P_1 = P_4 = 5\text{x}10^{-6}$ sec = .005 ms
P_2 = 15 ms
P_3 = 5 ms
Total = 20 ms

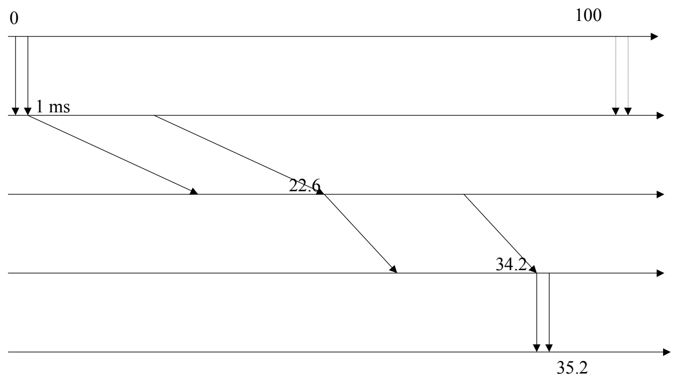
10 ms packetization, 1.5 Mbps: The packet transmission time is 1280/1.5Mbps=853 μsec. In this case, the long propagation delay dominates the overall delay.



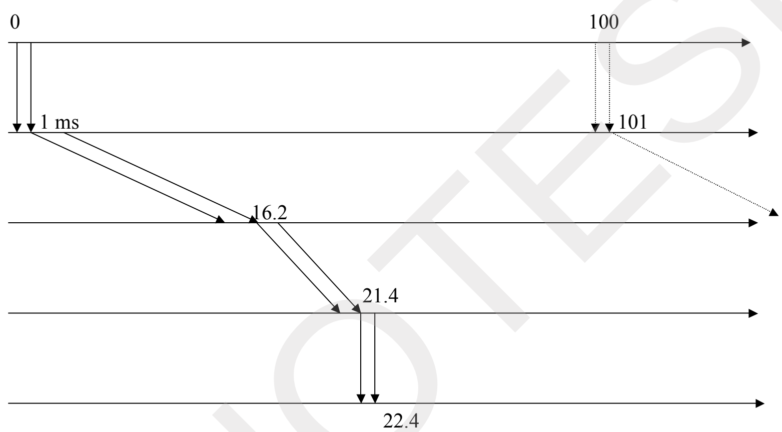
10 ms packetization, 45 Mbps: the packet transmission time is 1280/45Mbps=28.4 μsec. The high-speed in the backbone portion of the network does little to reduce the overall delay.



100 ms packetization, 1.5 Mbps: The packet transmission time is now 9920/1.5Mbps =6613 μsec. The long propagation delay and the long packet size provide comparable delays.



100 ms packetization, 45 Mbps: The high backbone speed makes the long packet transmission time negligible relative to the propagation delay.

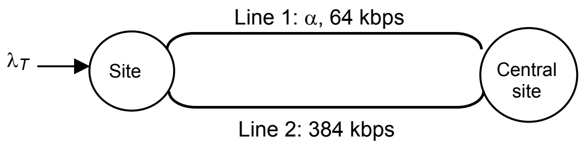


(d) Which of the preceding delay components would involve queueing delays?

The transfer in switches 1, 2, and 3 involve random queueing delays.

7.19. Suppose that a site has two communication lines connecting it to a central site. One line has a speed of 64 kbps, and the other line has a speed of 384 kbps. Suppose each line is modeled by an M/M/1 queueing system with average packet delay given by $E[D] = E[X]/(1 - \rho)$ where $E[X]$ is the average time required to transmit a packet, λ is the arrival rate in packets/second, and $\rho = \lambda E[X]$ is the load. Assume packets have an average length of 8000 bits. Suppose that a fraction α of the packets are routed to the first line and the remaining $1 - \alpha$ are routed to the second line.

Solutions follow questions:



(a) Find the value of α that minimizes the total average delay.

$E[L]$ = 8000 bits

$\lambda_1 = \alpha\lambda_T$, $\lambda_2 = (1 - \alpha)\lambda_T$

$E[X_1]$ = 8000 bits / 64 kbps = 0.125 sec

$E[X_2]$ = 8000 bits /384 kbps = 0.02083 sec

$E[D_T] = E[D_1]\,\alpha + E[D_2]\,(1 - \alpha)$, $\alpha < 1$

$$= \frac{1}{\frac{1}{E[X_1]} - \lambda_1}\alpha + \frac{1}{\frac{1}{E[X_2]} - \lambda_2}(1-\alpha)$$

$$= \frac{1}{8-\lambda_1}\alpha + \frac{1}{48-\lambda_2}(1-\alpha)$$

$$= \frac{1}{8-\lambda_T\alpha}\alpha + \frac{1}{48-\lambda_T(1-\alpha)}(1-\alpha)$$

Before doing the math to find the minimum in the above expression, it is worthwhile to consider a couple of extreme cases. The maximum capacity available to transfer packets across the combined links is 56 packets per second, 8 p/s across the first link and 48 p/s across the second link. Clearly when the arrival rate approaches capacity, we will send traffic across the two links. However consider the case when the arrival rate is very small. The delay across the first link is 1/8 and the delay across the second link is 1/48. Clearly at low traffic we want to send all the traffic on the fast link. As we increase the arrival rate, we eventually reach a point where it is worthwhile to route some packets along the slow link. Thus we expect α to be zero for some range of arrival rates, and then become nonzero after a specific value of arrival rate is reached.

Now take derivatives to find the minimum:

Let A be the first term in the delay expression and let B be the second term.

$$\frac{dE[D_T]}{d\alpha} = \frac{dA}{d\alpha} + \frac{dB}{d\alpha}$$

$$\frac{dA}{d\alpha} = \frac{d}{d\alpha}\frac{1}{8-\lambda_T\alpha}\alpha = (-1)(8-\lambda_T\alpha)^{-2}(-\lambda_T)\alpha + \frac{1}{8-\lambda_T\alpha}$$

$$= \frac{\lambda_T\alpha}{(8-\lambda_T\alpha)^2} + \frac{1}{8-\lambda_T\alpha} = \frac{\lambda_T\alpha + 8 - \lambda_T\alpha}{(8-\lambda_T\alpha)^2} = \frac{8}{(8-\lambda_T\alpha)^2}$$

Similarly, we have

$$\frac{dB}{d\alpha} = \frac{d}{d\alpha}\left[\frac{1}{48-\lambda_T(1-\alpha)}(1-\alpha)\right] = (-1)[48-\lambda_T+\lambda_T\alpha]^{-2}(\lambda_T)(1-\alpha) + \frac{-1}{48-\lambda_T(1-\alpha)}$$

$$= \frac{-\lambda_T(1-\alpha)}{(48-\lambda_T+\lambda_T\alpha)^2} + \frac{-1}{48-\lambda_T+\lambda_T\alpha} = \frac{\lambda_T(1-\alpha)-(48-\lambda_T+\lambda_T\alpha)}{48-\lambda_T+\lambda_T\alpha} = \frac{-48}{(48-\lambda_T+\lambda_T\alpha)^2}$$

To find the α that minimizes the delay, we set $\frac{dA}{d\alpha}+\frac{dB}{d\alpha}=0$. We then obtain:

$$\frac{8}{(8-\lambda_T\alpha)^2} + \frac{-48}{(48-\lambda_T+\lambda_T\alpha)^2} = 0$$

$$\frac{1}{(8-\lambda_T\alpha)^2} = \frac{6}{(48-\lambda_T+\lambda_T\alpha)^2}$$

$$6(8-\lambda_T\alpha)^2 = (48-\lambda_T+\lambda_T\alpha)^2$$

$$\sqrt{6}(8-\lambda_T\alpha) = 48-\lambda_T+\lambda_T\alpha$$

$$8\sqrt{6}-\sqrt{6}\lambda_T\alpha = 48-\lambda_T+\lambda_T\alpha$$

$$8\sqrt{6}-48+\lambda_T = \lambda_T\alpha+\sqrt{6}\lambda_T\alpha$$

$$\alpha = \frac{8\sqrt{6}-48+\lambda_T}{(1+\sqrt{6})\lambda_T} = \frac{\lambda_T-28.4}{3.45\lambda_T}$$

Thus we see that α must be zero for arrival rates less that 28.4.

(b) Compare the average delay in part (a) to the average delay in a single multiplexer that combines the two transmission lines into a single transmission line.

With a single transmission line the link speed will be 64 + 384 = 448 kbps. Therefore,

$E[X]$ = 8000 bits / 448 kbps = 0.0179 sec

$$E[D] = \frac{1}{\frac{1}{E[X]}-\lambda_T} = \frac{1}{\frac{1}{0.0179}-\lambda_T} = \frac{1}{56-\lambda_T}$$

The average delay in the single multiplexer is smaller than the average delay in the system served by two servers. This reduction is due to the fact that some of the capacity in the two-server system can be idle when there is only one packet in the system, whereas the single multiplexer system is always working at full rate.

20. A message of size m bits is to be transmitted over an L-hop path in a store-and-forward packet network as a series of N consecutive packets, each containing k data bits and h header bits. Assume that $m >> k + h$. The transmission rate of each link is R bits/second. Propagation and queueing delays are negligible.

Solutions follow questions:

We know from Figure 7.10 that if the message were sent as a whole, then an entire message propagation delay must be incurred at each hop. If instead we break the message into packets, then from Figure 7.12 a form of pipelining takes place where only a packet time is incurred in each hop. The point of this problem is to determine the optimum packet size that minimizes the overall delay.

We know that:

- message size = m bits = N packets
- one packet = k data bits + h header bits
- $m >> k + h$
- there are L hops of store-and-forward
- transmission rate = R bits/sec
- t_{prop} and t_{Qing} are negligible
- transmission time of one packet, T, equals the amount of bits in a packet / transmission rate, or $T = (k + h) / R$.

(a) What is the total number of bits that must be transmitted?

$N (k + h)$ bits are transmitted.

(b) What is the total delay experienced by the message (that is, the time between the first transmitted bit at the source and the last received bit at the destination)?

The total delay, D, can be expressed by

$$D = Lt_{prop} + LT + (N - 1)T.$$

Since the propagation delay is negligible, the first term in the above equation is zero. Substituting for T we get,

$$D = L\frac{k+h}{R} + (N-1)\frac{k+h}{R}$$

$$= (N - 1 + L)\frac{k+h}{R}$$

(c) What value of k minimizes the total delay?

We determined D in the previous equation. We also know that $N = \left\lceil \frac{m}{k} \right\rceil$, where m is the ceiling.

Therefore,

$$D \approx \left(\frac{m}{k} - 1 + L\right)\frac{k+h}{R} = \left(\frac{m}{k}\right)\frac{k+h}{R} + (L-1)\frac{k+h}{R}$$

Note that there are two components to the delay: 1. the first depends on the number of packets that need to be transmitted, *m/k*, as *k* increases this term decreases; 2. the second term increases as the size of the packets, *k* + *h*, is increased. The optimum value of *k* achieves a balance between these two tendencies.

$$\frac{dD}{dk} = \left[\frac{d}{dk}\left(\frac{m}{k} - 1 + L\right)\right]\frac{k+h}{R} + \left(\frac{m}{k} - 1 + L\right)\frac{d}{dk}\left(\frac{k+h}{R}\right)$$

$$= \frac{-m}{k^2} \times \frac{k+h}{R} + \left(\frac{m}{k} - 1 + L\right)\frac{1}{R}$$

If we set *dD/dk* to zero, we obtain:

$$\frac{-m}{k^2} \times \frac{k+h}{R} + \left(\frac{m}{k} - 1 + L\right)\frac{1}{R} = 0$$

$$\frac{m}{k^2}\left(\frac{k}{R} + \frac{h}{R}\right) = \frac{m}{kR} - \frac{1}{R} + \frac{L}{R}$$

$$\frac{m}{kR} + \frac{mh}{k^2 R} = \frac{m}{kR} + \frac{L-1}{R}$$

$$\frac{mh}{k^2} = L - 1$$

$$k^2 = \frac{mh}{L-1}$$

$$k = \sqrt{\frac{mh}{L-1}}$$

It is interesting to note that the optimum payload size is the geometric mean of the message size and the header, normalized to the number of hops (-1).

7.21. Suppose that a datagram packet-switching network has a routing algorithm that generates routing tables so that there are two disjoint paths between every source and destination that is attached to the network. Identify the benefits of this arrangement. What problems are introduced with this approach?

Solution:

There are several benefits to having two disjoint paths between every source and destination. The first benefit is that when a path goes down, there is another one ready to take over without the need of finding a new one. This reduces the time required to recover from faults in the network. A second benefit is that traffic from a source to a destination can be load-balanced to yield better delay performance and more efficient use of network resources. On the other hand, the computing and maintaining two paths for every destination increases the complexity of the routing algorithms. It is also necessary that the network topology be designed so that two disjoint paths are available between any two nodes.

7.22. Suppose that a datagram packet-switching network uses headers of length H bytes and a virtual-circuit packet-switching network uses headers of length h bytes. Use Figure 7.15 to determine the length M of a message for which the virtual-circuit switching delivers the packet in less time than datagram switching does. Assume packets in both networks are the same length.

Solution:

Assume the message M produces k packets of M/k bits each. With headers the datagram packet is then $H + M/k$ bits long, and the VC packet is $h + M/k$ bits long.

From Figure 7.16, the total delay for datagram packet switching (assuming a single path for all packets) is

$T = L\, t_{prop} + L\, P + (k\text{-}1)\, P$ where $P = (H + M/k)/R$.

For VC packet switching, we first incur a connection setup time of $2\, L\, t_{prop} + L\, t_{proc}$ for the processing and transmission of the signaling messages. The total delay to deliver the packet is then:

$T' = 2L\, t_{prop} + L\, t_{proc} + L\, t_{prop} + L\, P' + (k\text{-}1)\, P'$ where $P' = (h + M/k)/R$.

Datagram switching has higher total delay if $T > T'$, that is,

$L\, t_{prop} + L\, P + (k\text{-}1)\, P > 2L\, t_{prop} + L\, t_{proc} + L\, t_{prop} + L\, P' + (k\text{-}1)\, P'$

which is equivalent to:

$2L\, t_{prop} + L\, t_{proc} < (L+k\text{-}1)(H\text{-}h)/R$.

This condition is satisfied when the VC setup time is small and the difference between the lengths of the datagram and VC headers is large.

It is also interesting to consider VC packet switching with cut-through switching. In this case the overall delay is:

$T'' = 2L\, t_{prop} + L\, t_{proc} + L\, t_{prop} + L\, h + k\, P'$ where $P' = (h + M/k)/R$.

Now datagram switching will require more time if:

$L\, t_{prop} + L\, P + (k\text{-}1)\, P > 2L\, t_{prop} + L\, t_{proc} + L\, t_{prop} + L\, h + (k\text{-}1)\, P'$

which is equivalent to:

$2L\, t_{prop} + L\, t_{proc} < (k\text{-}1)(H\text{-}h)/R - L(P\text{-}h)$.

Note that if the VC setup time can be avoided (for example if the VC has been set up by previous messages and if the network caches the VC for some time), then the delay for the VC approach improves dramatically.

7.23. Consider the operation of a packet switch in a connectionless network. What is the source of the load on the processor? What can be done if the processor becomes the system bottleneck?

Solution:

There are two sources of load on the processor. The first involves the processing involved in determining the route and next-hop for each packet. The second source of load to the processor is the implementation of routing protocols. This involves the gathering of state information, the exchange of routing messages with other nodes, and the computation of routes and associated routing tables.

The first source of processing load can be addressed by implementing per-packet processing in hardware, and by distributing this processing to line cards in a switch. Distributing the computation of the routing tables in a switch between central processors and processors in the line cards can be done but is quite tricky.

7.24. Consider the operation of a packet switch in a connection-oriented network. What is the source of the load on the processor? What can be done if the processor becomes overloaded?

Solution:

The main source of load in the switch is in the processing required for the setting up and tearing down of connections. The processor must keep track of the connections that are established in the switch and of the remaining resources for accepting new connections. The processor must participate in the exchange of signals to setup connections and it must implement connection admission procedures to determine when a connection request should be accepted or rejected. The processor load increases with the calling rate. The processor must be upgraded when the calling rate approaches the call handling capacity of the processor. Alternatively, the processing can be distributed between central processors and processing in the line cards.

7.25. Consider the following traffic patterns in a banyan switch with eight inputs and eight outputs in Figure 7.21. Which traffic patterns below are successfully routed without contention? Can you give a general condition under which a banyan switch is said to be non-blocking (that is, performs successful routing without contention)?

Solutions:

(a) Pattern 1: Packets from inputs 0, 1, 2, 3, and 4 are to be routed to outputs 2, 3, 4, 5, and 7, respectively.

(b) Pattern 2: Packets from inputs 0, 1, 2, 3, and 4 are to be routed to outputs 1, 2, 4, 3, and 6, respectively.

(c) Pattern 3: Packets from inputs 0, 2, 3, 4, and 6 are to be routed to outputs 2, 3, 4, 5, and 7, respectively.

Pattern a is routed without contention. Patterns b and c have contention. A banyan switch is nonblocking if the input packets are sorted according to their destination port numbers and are given to the switch entries starting from top with no gaps or duplicates.

7.26. Suppose a routing algorithm identifies paths that are "best" in the sense that: (1) minimum number of hops, (2) minimum delay, or (3) maximum available bandwidth. Identify conditions under which the paths produced by the different criteria are the same? are different?

Solution:

The first criterion ignores the state of each link, but works well in situations were the states of all links are the same. Counting number of hops is also simple and efficient in terms of the number of bits required to represent the link. Minimum hop routing is also efficient in the use of transmission resources, since each packet consumes bandwidth using the minimum number of links.

The minimum delay criterion will lead to paths along the route that has minimum delay. If the delay is independent of the traffic levels, e.g. propagation delay, then the criterion is useful. However, if the delay is strongly dependent on the traffic levels, then rerouting based on the current delays in the links will change the traffic on each link and hence the delays! In this case, not only does the current link delay need to be considered, but also the derivative of the delay with respect to traffic level.

The maximum available bandwidth criterion tries to route traffic along pipes with the highest “cross-section” to the destination. This approach tends to spread traffic across the various links in the network. This approach is inefficient relative to minimum hop routing in that it may use longer paths.

At very low traffic loads, the delay across the network is the sum of the transmission times and the propagation delays. If all links are about the same length and bit rate, then minimum hop routing and minimum delay routing will give the same performance. If links vary widely in length, then minimum hop routing may not give the same performance as minimum delay routing.

Minimum hop routing will yield the same paths as maximum available bandwidth routing if links are loaded to about the same levels so that the available bandwidth in links is about the same. When link utilization varies widely, maximum available bandwidth routing will start using longer paths.

7.27. Suppose that the virtual circuit identifiers (VCIs) are unique to a switch, not to an input port. What is traded off in this scenario?

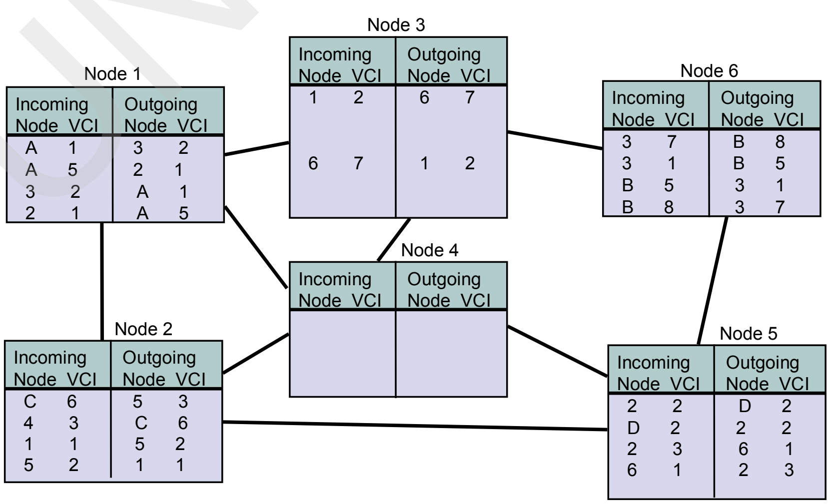
Solution:

When the VCIs are unique only within an input port, then the VCI numbers can be reused at all ports in the switch. Thus the number of VCIs that can be supported increases with the number of ports. In addition, VCI allocation is decentralized and can be handled independently by each input port. When the VCIs are unique only to the switch, then the number of VCIs that can be handled is reduced drastically. In addition VCI allocation must now be done centrally. Thus the per-switch VCI allocation approach may only be useful in small switch, small network scenarios.

7.28. Consider the virtual-circuit packet network in Figure 7.23. Suppose that node 4 in the network fails. Reroute the affected virtual circuits and show the new set of routing tables.

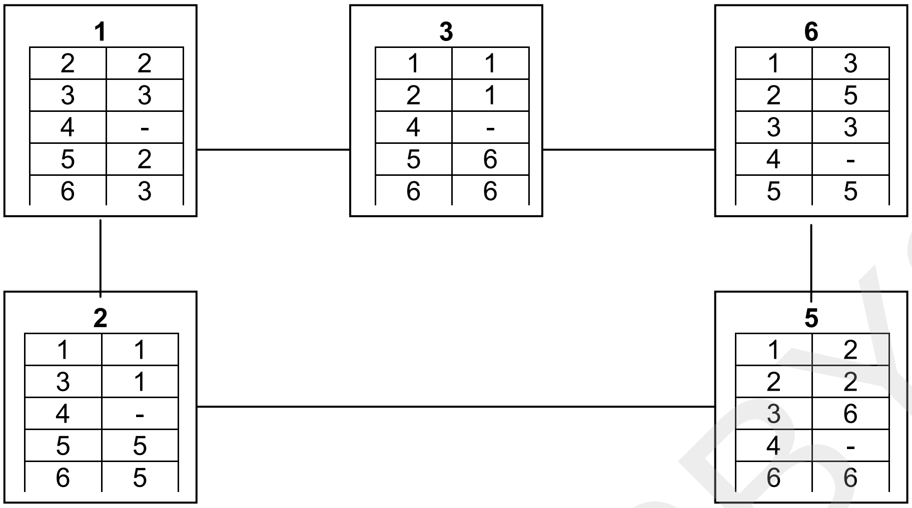
Solution:

When node 4 fails, all the affected connections need to be re-routed. In the example, the VC from A to D needs to be rerouted. One possible new route is A-1-2-5-D. The routing tables at nodes 1, 2, and 5 need to be modified. The VC from C to B also needs to be rerouted. A possible new route is C-2-5-6-D. The routing tables in 2, 5, and 6 need to be modified. The routing tables in node 3 have to be modified to remove the connections that no longer traverse it.



7.29. Consider the datagram packet network in Figure 7.25. Reconstruct the routing tables (using minimum hop routing) that result after node 4 fails. Repeat if node 3 fails instead.

Solution:



1

2	2
3	3
4	-
5	2
6	3

3

1	1
2	1
4	-
5	6
6	6

6

1	3
2	5
3	3
4	-
5	5

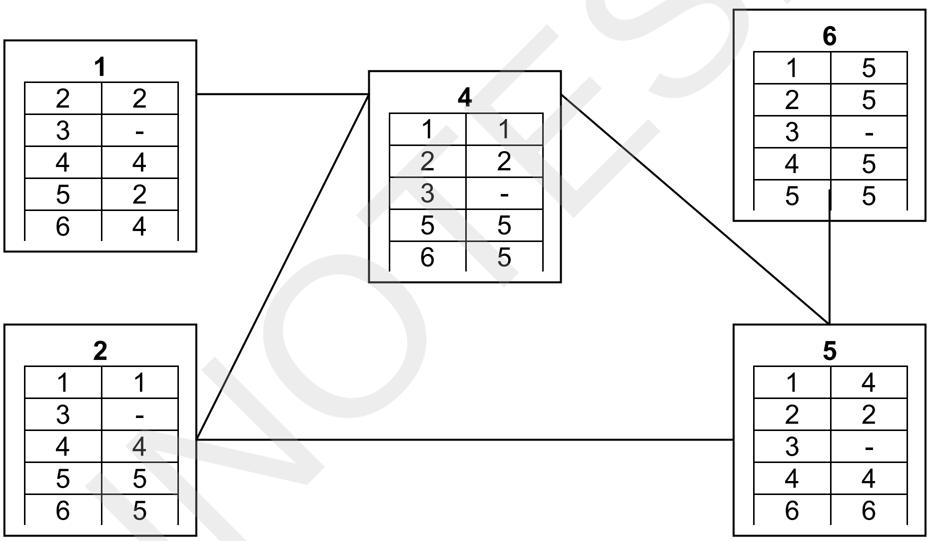
2

1	1
3	1
4	-
5	5
6	5

5

1	2
2	2
3	6
4	-
6	6

If node 3 fails:



1

2	2
3	-
4	4
5	2
6	4

4

1	1
2	2
3	-
5	5
6	5

6

1	5
2	5
3	-
4	5
5	5

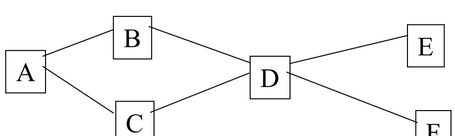
2

1	1
3	-
4	4
5	5
6	5

5

1	4
2	2
3	-
4	4
6	6

7.30. Consider the following six-node network. Assume all links have the same bit rate *R*.



Solutions follow questions:

(a) Suppose the network uses datagram routing. Find the routing table for each node, using minimum hop routing.

Node A:

destination	next node
B	B
C	C
D	B
E	B
F	B

Node B:

destination	next node
A	A
C	D
D	D
E	D
F	D

Node C:

destination	next node
A	A
B	A
D	D
E	D
F	D

Node D:

destination	next node
A	B
B	B
C	C
E	E
F	F

Node E:

destination	next node
A	D
B	D
C	D
D	D
F	D

Node F:

destination	next node
A	D
B	D
C	D
D	D
E	D

(b) Explain why the routing tables in part (a) lead to inefficient use of network bandwidth.

Minimum-hop based routing causes all traffic flow through the minimum hop path, while the bandwidth of other paths of equal and longer distance are unused. It is inefficient in its usage of network bandwidth and likely to cause congestion in the minimum-hop paths.

For example, in the routing table of node A, the traffic flows from A to E and A to F both pass through node B, and leave the bandwidth of link AC and CD unused.

(c) Can VC routing give better efficiency in the use of network bandwidth? Explain why or why not.

Yes. During the VC setup, the links along the path can be examined for the amount of available bandwidth, if it below a threshold, the connection will be routed over alternate path. VC routing allows more even distribution of traffic flows and improves the network bandwidth efficiency.

(d) Suggest an approach in which the routing tables in datagram network are modified to give better efficiency. Give the modified routing tables.

Assign a cost to each link that is proportional to the traffic loading (number of connections) of the link. A higher cost is assigned to more congested links. Thus the resulting routing table avoids congested links and distributes traffic more evenly.

The changes in the routing table are in bold and italicized.

Node A:

destination	next node
B	B
C	C
D	B
E	***C***
F	***C***

Node B:

destination	next node
A	A
C	***A***
D	D
E	D
F	D

Node C:

destination	next node
A	A
B	A
D	D
E	D
F	D

Node D:

Destination	next node
A	B
B	B
C	C
E	E
F	F

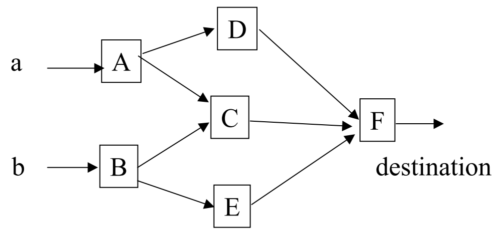
Node E:

destination	next node
A	D
B	D
C	D
D	D
F	D

Node F:

destination	next node
A	D
B	D
C	D
D	D
E	D

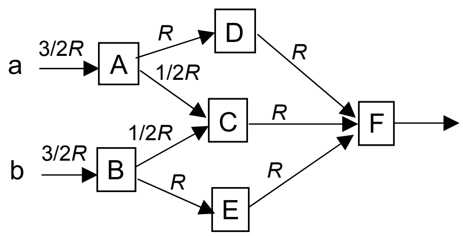
7.31. Consider the following six-node unidirectional network where flows a and b are to be transferred to the same destination. Assume all links have the same bit rate $R = 1$.



Solutions follow questions:

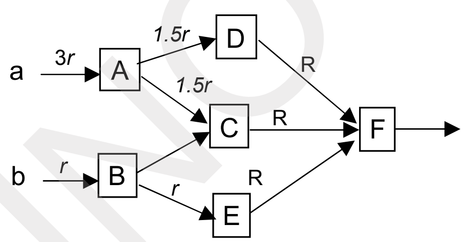
(a) If the flows a and b are equal, find the maximum flow that can be handled by the network.

The links to node F are the bottleneck links. A maximum flow of 3*R* can be handled.



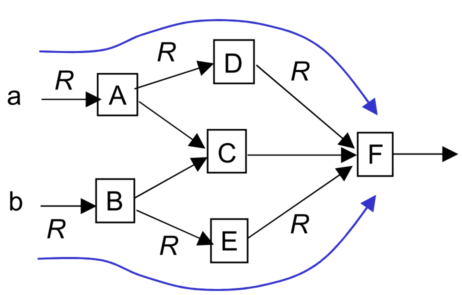
(b) If flow a is three times as large as flow b, find the maximum flow that can be handled by the network.

Let b = *r* and a = 3*r*. Flow a sees a maximum cross-section of 2*R* flow across the network, and flow b sees a cross-section of 2*R* as well. Since flow a has the larger flow, its flow of 3*r* is limited to a maximum value of 2*R*, that is, *r* = 2*R*/3. It then follows that b must be given by *r* = 2*R*/3.

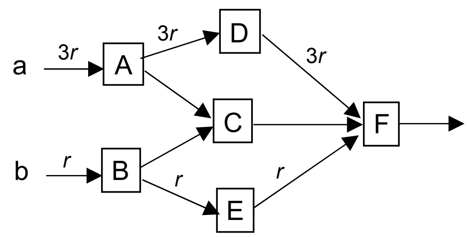


(c) Repeat (a) and (b) if the flows are constrained to use only one path.

Part (a): To have maximum flow, flow a must take path A → D → F. Similarly, flow b must take the path B → E → F. In this case, a = *R* and b = *R*. So the total maximum flow is 2*R*.



Part (b): $R = 3r$ and $r = 1/3R$. Therefore a = R and b = $1/3R$. The total maximum flow is $4/3R$.



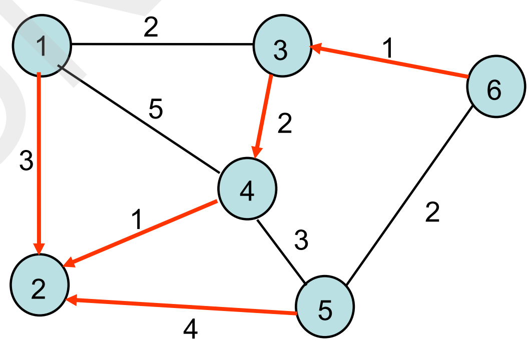
7.32. Consider the network in Figure 7.30.

Solutions follow questions:

(a) Use the Bellman-Ford algorithm to find the set of shortest paths from all nodes to destination node 2.

Iteration	Node 1	Node 3	Node 4	Node 5	Node 6
Initial	(-1 , ∞)	(-1 , ∞)	(-1 , ∞)	(-1 , ∞)	(-1 , ∞)
1	(2 , 3) (3 , ∞) (4 , ∞)	(1 , ∞) (4 , ∞) (6 , ∞)	(1 , ∞) (2 , 1) (3 , ∞) (5 , ∞)	(2 , 4) (4 , ∞) (6 , ∞)	(3 , ∞) (5 , ∞)
2	(2 , 3) (3 , ∞) (4 , 6)	(1 , 5) (4 , 3) (6 , ∞)	(1 , 8) (2 , 1) (3 , ∞) (5 , 7)	(2 , 4) (4 , 4) (6 , ∞)	(3 , ∞) (5 , 6)
3	(2 , 3) (3 , 5) (4 , 6)	(1 , 5) (4 , 3) (6 , 7)	(1 , 8) (2 , 1) (3 , 5) (5 , 7)	(2 , 4) (4 , 4) (6 , 8)	(3 , 4) (5 , 6)
4	(2 , 3) (3 , 5) (4 , 6)	(1 , 5) (4 , 3) (6 , 5)	(1 , 8) (2 , 1) (3 , 5) (5 , 7)	(2 , 4) (4 , 4) (6 , 6)	(3 , 4) (5 , 6)

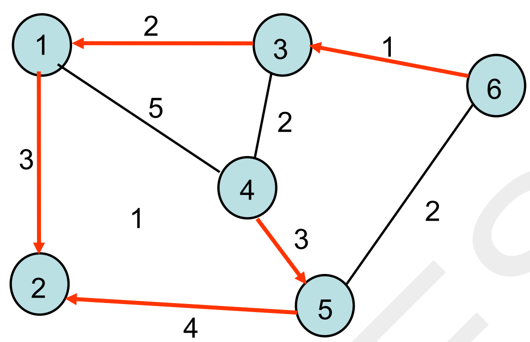
The set of paths to destination 2 are shown below:



Now continue the algorithm after the link between node 2 and 4 goes down.

Iteration	Node 1	Node 3	Node 4	Node 5	Node 6
Before break	(2 , 3)	(4 , 3)	(2 , 1)	(2 , 4)	(3 , 4)
1	(2 , 3) (3 , 5) (4 , 6)	(1 , 5) (4 , 3) (6 , 5)	(1 , 8) (3, 5) (5 , 7)	(2 , 4) (4 , 4) (6 , 6)	(3 , 4) (5 , 6)
2	(2 , 3) (3 , 5) (4 , 10)	(1 , 5) (4 , 7) (6 , 5)	(1 , 8) (3 , 5) (5 , 7)	(2 , 4) (4 , 8) (6 , 6)	(3 , 4) (5 , 6)
3	(2 , 3) (3 , 7) (4 , 10)	(1 , 5) (4 , 7) (6 , 5)	(1 , 8) (3 , 7) (5 , 7)	(2 , 4) (4 , 8) (6 , 6)	(3 , 6) (5 , 6)
4	(2 , 3) (3 , 7) (4 , 12)	(1 , 5) (4 , 9) (6 , 7)	(1 , 8) (3 , 7) (5 , 7)	(2 , 4) (4 , 10) (6 , 8)	(3 , 6) (5 , 6)

The new set of paths are shown below:

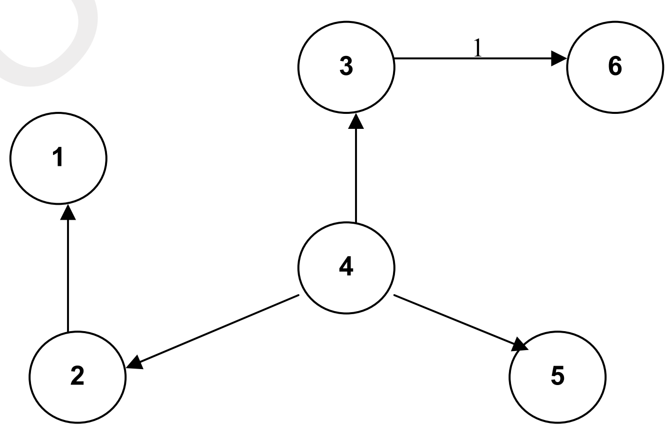


7.33. Consider the network in Figure 7.30.

Solutions follow questions:

(a) Use the Dijkstra algorithm to find the set of shortest paths from node 4 to other nodes.

Iteration	N	D_1	D_2	D_3	D_5	D_6
Initial	{ 4 }	5	1	2	3	∞
1	{ 2, 4 }	4	**1**	2	3	∞
2	{ 2, 3, 4 }	4	1	**2**	3	3
3	{ 2, 3, 4, 5 }	4	1	2	**3**	3
4	{ 2, 3, 4, 5, 6 }	4	1	2	3	**3**
5	{1, 2, 3, 4, 5, 6}	**4**	1	2	3	3



(b) Find the set of associated routing table entries.

Destination	Next Node	Cost
1	2	4
2	2	1
3	3	2
5	5	3
6	3	3

7.34. Compare source routing with hop-by-hop routing with respect to (1) packet header overhead, (2) routing table size, (3) flexibility in route selection, and (4) QoS support, for both connectionless and connection-oriented packet networks.

Solution:

Packet header overhead is higher in source routing in both connectionless and connection-oriented networks because the ID of the nodes across the path is included in the header in source routing while in hop-by-hop routing only the destination in connectionless networks or the connection number in connection-oriented networks is specified in the header.

Source routing requires no routing table lookup in intermediate nodes while hop-by-hop routing requires a routing table with one entry per destination in connectionless networks or one entry per connection in connection-oriented networks. Source nodes however must have sufficient routing information to determine the path across the network, essentially the same information required in link state routing.

Source routing is more flexible because it can consider all possible routes from the source to destination while in hop-by-hop routing each node can choose only its neighbors. However, hop-by-hop routing can respond more quickly to changes in the network.

Source routing can incorporate QoS considerations more flexibly as it can consider all possibilities in routing, while hop-by-hop routing can incorporate the QoS considerations at each hop but not universally.

7.35. Suppose that a block of user information that is *L* bytes long is segmented into multiple cells. Assume that each data unit can hold up *P* bytes of user information, that each cell has a header that is *H* bytes long, and the cells are fixed in length, and padded if necessary. Define the efficiency as the ratio of the *L* user bytes to the total number of bytes produced by the segmentation process.

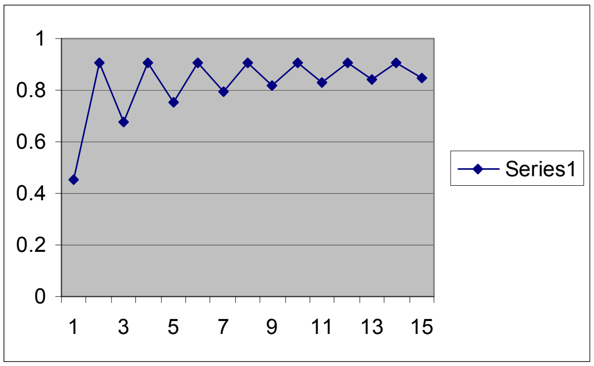
Solutions follow questions:

(a) Find an expression for the efficiency as a function of *L, H*, and *P*. Use the ceiling function *c(x)* which is defined as the smallest integer larger or equal to *x*.

$$eff = \frac{L}{L + H \times c\left(L/P\right)} = \frac{1}{1 + \left(H/L\right) \times c\left(L/P\right)}$$

(b) Plot the efficiency for the following ATM parameters: $H = 5$, $P = 48$, and $L = 24k$ for $k = 0, 1, 2, 3, 4, 5$, and 6.

0	0	0
1	0.452830189	0.45283
2	0.905660377	0.90566
3	0.679245283	0.679245
4	0.905660377	0.90566
5	0.754716981	0.754717
6	0.905660377	0.90566
7	0.79245283	0.792453
8	0.905660377	0.90566
9	0.81509434	0.815094
10	0.905660377	0.90566
11	0.830188679	
12	0.905660377	
13	0.84097035	
14	0.905660377	
15	0.849056604	



7.36. Consider a videoconferencing application in which the encoder produces a digital stream at a bit rate of 144 kbps. The packetization delay is defined as the delay incurred by the first byte in the packet from the instant it is produced to the instant when the packet if filled. Let *P* and *H* be defined as they are in problem 7.35.

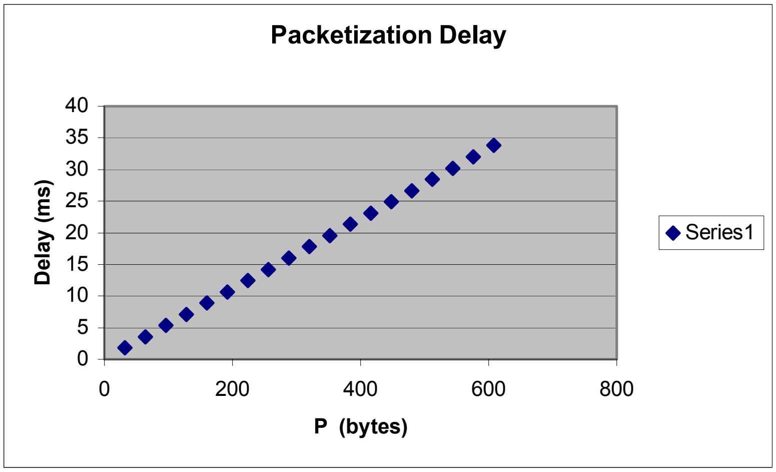
Solutions follow questions:

(a) Find an expression for the packetization delay for this video application as a function of *P*.

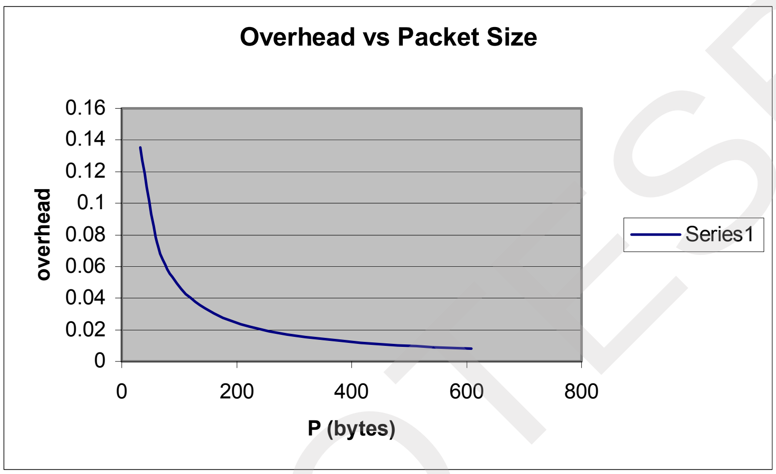
packetization delay = *P*8* / (144 kbps)

(b) Find an expression for the efficiency as a function of *P* and *H*. Let $H = 5$ and plot the packetization delay and the efficiency versus *P*.

efficiency = *P* / (*P* + *H*) = *1 / (1 + H/P*)



overhead = 1 – efficiency



The two graphs show that there is a tradeoff between efficiency and packetization delay. The efficiency improves as *P* is increased; however the packetization delay also increases with *P*. An overly stringent delay requirement can lead to bandwidth inefficiency.

7.37. Suppose an ATM switch has 16 ports each operating at SONET OC-3 transmission rate, 155 Mbps. What is the maximum possible throughput of the switch?

Solution:

The maximum possible throughput of the switch is achieved when all of the input ports operate at maximum speed:

155 Mbps $\times$ 16 = 2.48 Gbps.

This calculation assumes that the switch fabric is capable of switching the maximum amount of traffic that can arrive at the line cards. Achieving a capacity of 2.48 Gbps is trivial today, but at much higher rates (say, a terabit-per-second) many switch designs do not have the ability to switch all the traffic that can arrive at the input ports.

The student is warned that the networking industry literature tends to quote the capacity of a switch as the sum of the input and output bit rates of all the line cards. This practice gives a capacity that is at best twice the actual maximum throughput. As indicated above, it is not unusual for a switch fabric to not have enough capacity to transfer all the traffic that can arrive at its input ports.

7.38. Refer to the virtual-circuit packet-switching network in Figure 7.23. How many VCIs does each connection in the example consume? What is the effect of the length of routes on VCI consumption?

Solution:

Each connection uses the same number of VCIs as the number of hops in its path. However, since each VCI is defined local to each link, the use of a VCI on one hop does not deplete the available VCIs in another link.

7.39. Generalize the hierarchical network in Figure 7.26 so that 2^K nodes are interconnected in a full mesh at the top of the hierarchy, and so that each node connects to 2^L nodes in the next lower level in the hierarchy. Suppose there are four levels in the hierarchy.

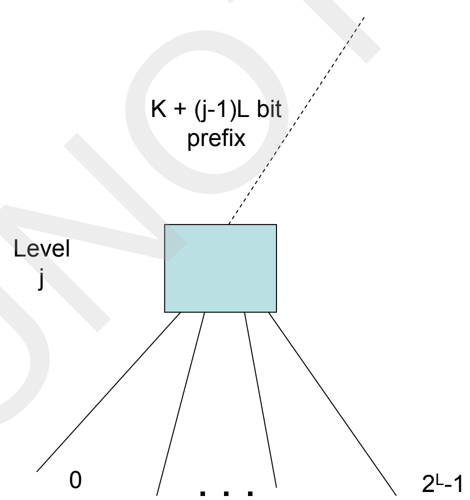
Solutions follow questions:

(a) How many nodes are there in the hierarchy?

$2^K \times 2^L \times 2^L \times 2^L$

(b) What does a routing table look like at level j in the hierarchy, j = 1, 2, 3, and 4?

Each router is placed at a node in the tree network. The routes available either go up a single link in the hierarchy or down 2^L possible links down in the hierarchy. Packets that go down in the hierarchy have prefixes that correspond to the index of the node. The first L-bit symbol after said index indicates which link is to be taken down the hierarchy. All other packets go up the hierarchy by default.



(c) What is the maximum number of hops between nodes in the network?

6 hops: maximum 3 hops to the top level and maximum 3 hops to the lowest level

7.40. Assuming that the earth is a perfect sphere with radius 6400 km, how many bits of addressing is required to have a distinct address for every 1 cm x 1 cm square on the surface of the earth?

Solution:

Area of sphere = $4\pi r^2$, r is the radius.

$$\log_2\left[\frac{4\times\pi\times\left(6400\times10^5\right)^2}{1\times1}\right]=63 \text{ bits}$$

This is fewer bits than the 128 bits of addressing used in IP version 6 discussed in Chapter 8.

7.41. Suppose that 64 kbps PCM coded speech is packetized into a constant bit rate ATM cell stream. Assume that each cell holds 48 bytes of speech and has a 5 byte header.

Solutions follow questions:

(a) What is the interval between production of full cells?

$$\frac{48\times8\text{ bits}}{64\text{ Kb/sec}}=6\text{ milliseconds}$$

(b) How long does it take to transmit the cell at 155 Mbps?

$$\frac{(48+5)\times8\text{ bits}}{155\text{ Mbps}}=2.73\text{ microseconds}$$

(c) How many cells could be transmitted in this system between consecutive voice cells?

$$\frac{.006}{2.73\times10^{-6}}=2197 \text{ cells}$$

7.42. Suppose that 64 kbps PCM coded speech is packetized into a constant bit rate ATM cell stream. Assume that each cell holds 48 bytes of speech and has a 5 byte header. Assume that packets with silence are discarded. Assume that the duration of a period of speech activity has an exponential distribution with mean 300 ms and that the silence periods have a duration that also has an exponential distribution but with mean 600 ms. Recall that if T has an exponential distribution with mean $1/\mu$, then $P[T>t] = e^{-\mu t}$.

Solutions follow questions:

(a) What is the peak cell rate of this system?

$$\text{peak cell rate} = \frac{64\text{ kbps}}{48\times8\text{ bits}}\approx167\text{ cells/sec}$$

(b) What is the distribution of the burst of packets produced during an active period?

A speech activity period produces k packets if the duration of the burst is between $(k-1)\, t_{cell}$ and $k\, t_{cell}$, where t_{cell} = 6 ms. Therefore the probability of a burst of size k is given by:

$$P[k]=P[(k-1)0.006\le T\le k0.006]=e^{-6(k-1)/300}-e^{-6k/300}=(1-e^{-6/300})(e^{-6/300})^{k-1}$$

This is a geometric distribution with parameter $p = e^{-6/300}$.

(c) What is the average rate at which cells are produced?

$$\frac{167 \text{ cells/sec} \times 300 \text{ ms}}{900 \text{ ms}} = 56 \text{ cells/sec}$$

7.43. Suppose that a data source produces information according to an on/off process. When the source is on, it produces information at a constant rate of 1 Mbps; when it is off, it produces no information. Suppose that the information is packetized into an ATM cell stream. Assume that each cell holds 48 bytes of data and has a 5 byte header. Assume that the duration of an on period has a Pareto distribution with parameter $\alpha = 1$. Assume that the off period is also Pareto but with general parameter α.. If T has a Pareto distribution with parameter α, then $P[T > t] = t^{-\alpha}$ for $t > 1$. If $\alpha > 1$, then $E[T] = \alpha/(\alpha - 1)$, and if $0 < \alpha < 1$, then $E[T]$ is infinite.

Solutions follow questions:

(a) What is the peak cell rate of this system?

peak cell rate = $\frac{1 \text{ Mbps}}{48 \times 8 \text{ bits}} \approx 2604 \text{ cells/sec}$

(b) What is the distribution of the burst of packets produced during an on period?

The time to produce a cell is given by t_{cell} = 1/2604 = 0.384 ms

$$\begin{aligned} P[k] &= P[(k-1)0.000384 \le T \le k0.000384] = 1-(k.000384)^{-\alpha} - (1-((k-1).000384)^{-\alpha} \\ &= (2604/(k-1))^{\alpha} - (2604/k)^{\alpha} \end{aligned}$$

If $\alpha = 1$, then

$$P[k] = (2604/(k-1)) - (2604/k)$$

(c) What is the average rate at which cells are produced?

If $\alpha > 1$, then the average length of a burst is $\alpha/(\alpha-1)$. Suppose that the off time has a Pareto distribution with parameter $\beta > 1$, and average length $\beta/(\beta-1)$. Thus the average rate at which cells are produced is the peak cell rate multiplied by the average time that the data source produces information:

$$\frac{\alpha/(\alpha-1)}{\alpha/(\alpha-1)+\beta/(\beta-1)} \times 2604$$

The Pareto distribution is characterized by high variability. For example the variance is infinite for $\alpha < 2$. Thus the on and off periods will exhibit high variability and are therefore very difficult to deal with in traffic management.

When α is less than 1, then the mean time in the on state is infinite. In this case, the source can stay in the on state forever and cells will arrive continuously at the peak rate.

7.44. An IP packet consists of 20 bytes of header and 1500 bytes of payload. Now suppose that the packet is mapped into ATM cells that have 5 bytes of header and 48 bytes of payload. How much of the resulting cell stream is header overhead?

Solution:

Total payload for ATM: 1520 bytes
This implies 32 ATM frames
Total ATM header bytes: 160
Total Header bytes: 180
Total bytes transmitted: 1696

Header overhead = 180 / 1696 = 10.61%

7.45. Suppose that virtual paths are set up between every pair of nodes in an ATM network. Explain why connection set up can be greatly simplified in this case.

Solution:

When two nodes need to communicate, each switch in the path does not have to be involved in the connection set up. Instead the switches at the ends of the VP assign an end-to-end VCI to each connection. When a cell traverses the network, only the VPI is used to forward it. The VCI is used only to direct the cell to the appropriate VC as the cell exits the network. The problem with this approach is that the number of virtual paths required grows as the square of the number of nodes in the ATM network.

7.46. Suppose that the ATM network concept is generalized so that packets can be variable in length. What features of ATM networking are retained? What features are lost?

Solution:

The capability of forwarding frames end-to-end across a virtual circuit using label switching is retained even if packets are variable in length. The ability to police, schedule and shape packet flows is also retained, but the fine degree of control afforded by the small and constant ATM cells is lost. The fast switching afforded by fixed-length blocks need not be lost since a switch can internally switch fixed-length blocks while still appearing as a variable length switch to the rest of the network. The header becomes more complex because of the variable packet length. On the other hand, the header processing rate is reduced because the time between header arrivals increases. For example a line card that handles ATM cells at 10 Gbps must process a cell every $53 \times 8 / 10^{10} = 42.4$ nanoseconds. The same line card handling 1500 byte MPLS packets must process a cell every $1500 \times 8 / 10^{10} = 1200$ nanoseconds, which is less demanding by a factor of nearly 30 times.

7.47. Explain where priority queueing and fair queueing may be carried out in the generic switch/router in Figure 7.19.

Solution:

Queueing and scheduling are used whenever there is contention for resources. In the ideal case, called "output-buffered queueing", the interconnection fabric has ample capacity to transfer all arriving packets virtually instantly from the input port to the output port. In this case, each output line card appears like a multiplexer that must exercise priority and fair queueing to decide the order in which packets are transmitted. Most switch designs depart from the ideal case and are limited in the rate at which packets can be transferred from input line cards to output line cards. Consequently, packets can accumulate in the input line card. In this case, scheduling is required to decide the order in which packets must be transferred across the fabric.

7.48. Consider the head-of-line priority system in Figure 7.42. Explain the impact on the delay and loss performance of the low-priority traffic under the following conditions:

Solutions follow questions:

The presence of high-priority packets prevents low-priority packets from accessing the transmission line. The periods of times when the high-priority buffer is non-empty are periods during which the low-priority buffer builds up without relief. The delay and loss performance of the low-priority traffic is therefore affected strongly by the pattern of bandwidth usage by the high-priority stream.

If high priority packets can preempt the transmission of low-priority packets, then low priority traffic has no effect whatsoever on high priority traffic. If high priority packets cannot preempt low priority traffic, then arriving high priority packets must wait for the completion of service of the low priority packet found upon arrival. If low priority packets can be long, then this impact can be significant.

(a) The high-priority traffic consists of uniformly spaced, fixed-length packets.

Packets of the high-priority queue have constant arrival rate and service rate. Thus the low priority traffic is deprived of access to transmission bandwidth during periodically recurring time intervals. To the extent that the transmission line is sufficiently available to handle the low priority stream, the performance of low priority traffic is relatively steady and predictable.

(b) The high-priority traffic consists of uniformly spaced, variable-length packets.

The high priority traffic has constant arrival rate but variable service rate due to the variable-length packets. The periods during which low priority traffic has access to transmission is now less predictable. Consequently, there is increased variability in delay and increased average loss and delay.

(c) The high-priority traffic consists of highly bursty, variable-length packets.

The availability of bandwidth to the low priority traffic is now highly unpredictable. The delay and loss performance of the low priority traffic is the worst among all three cases.

7.49. Consider the head-of-line priority system in Figure 7.42. Suppose that each priority class is divided into several subclasses with different "drop" priorities. For each priority subclass there is a threshold that if exceeded by the queue length results in discarding of arriving packets from the corresponding subclass. Explain the range of delay and loss behaviors that are experienced by the different subclasses.

Solution:

Each priority class will experience delay performance better than that of lower priority classes and worse than that of higher priority classes.

Within each class, packets with higher drop preference will experience higher packet loss probability than packets from the same class but of lower drop preference. Packets with higher drop preference that are admitted into the buffer will also experience lower average delay than packets of lower drop precedence because on average they well find a smaller number of packets in the buffer.

7.50. Incorporate some form of weighted fair queueing in the head-of-line priority system in Figure 7.42 so that the low-priority traffic is guaranteed to receive r bps out of the total bit rate R of the transmission link. Explain why this may be a desirable feature. How does this impact the performance of the high-priority traffic?

Solution:

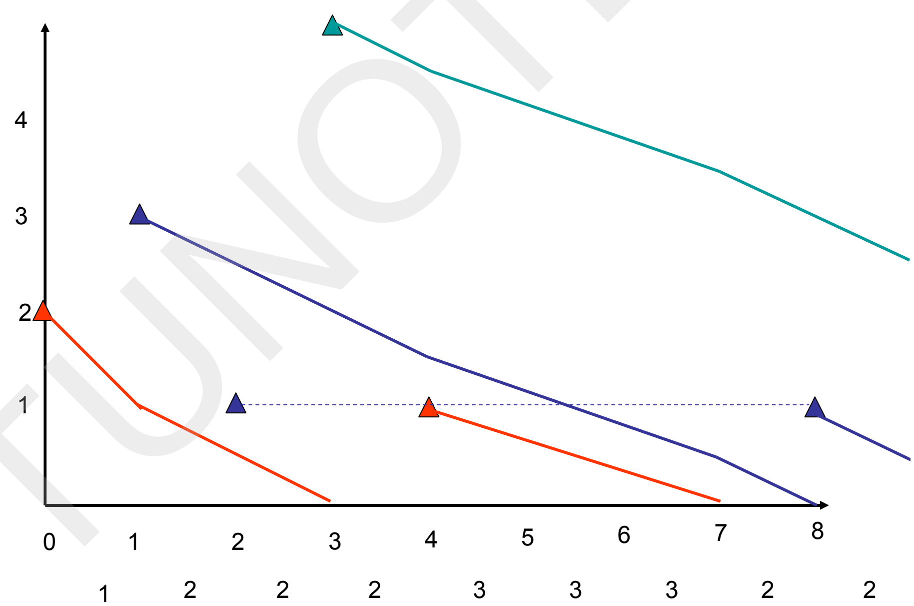
The trouble with the strict priority system in Figure 7.43 is that high priority traffic can starve the low priority traffic if no limits are placed on the volume of high priority traffic that is allowed into the system. An effective method for preserving the priority structure but preventing starvation of low priority traffic is to shape the high priority stream so that the average rate does not exceed some value, say $R - r$. In this case, r bps is guaranteed to become available to the low priority traffic. The performance of the high priority traffic is affected only insofar as the shaping introduces delay when packets need to be delayed to meet the shaping constraint. The guarantee of r bps to the low priority traffic can significantly improve the delay performance of the low priority traffic.

7.51. Consider a packet-by-packet fair queueing system with three logical buffers and with service rate of one unit/second. Show the sequence of transmissions for this system for the following packet arrival pattern. Buffer 1: arrival at time $t = 0$, length 2; arrival at $t = 4$, length 1. Buffer 2: arrival at time $t = 1$, length 3; arrival at $t = 2$, length 1. Buffer 3: arrival at time $t = 3$, length 5.

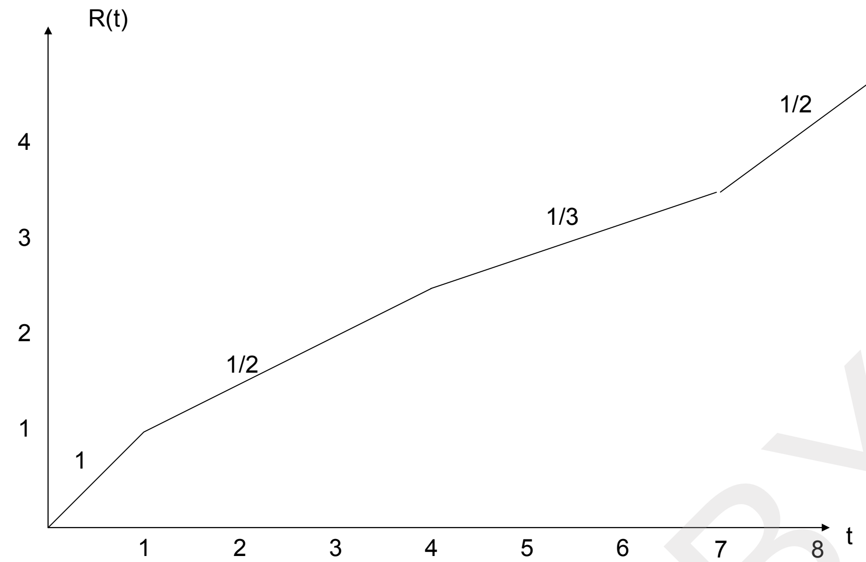
Solution:

The packet-by-packet fair queueing system selects the order of packet transmission according to the finish times in an emulated fluid flow system. The first figure below shows the times at which the various packets arrive and the length of the packets. The red triangles indicate packet arrivals at buffer 1; blue at buffer 2; and green at buffer 3. The rate at which packets are serviced is the reciprocal of the number of active buffers at the time. The second figure below shows how the round number $R(t)$ grows as a function of time. The slope of $R(t)$ is the reciprocal of the number of active stations. A given packet finishes service in the fluid flow system when the round number reaches the finish time computed for the given packet.

First Figure: packet arrivals and fluid flow service:



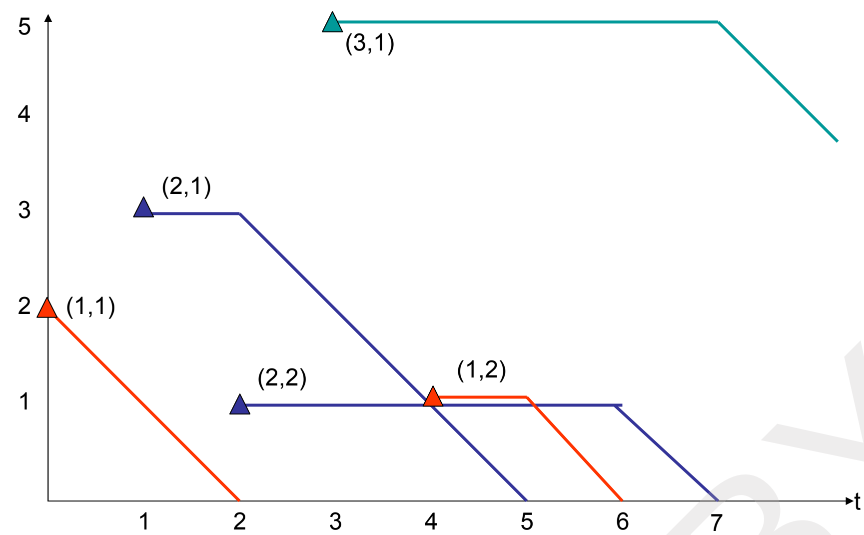
Second Figure: Round number vs. time



The table below gives the finish times for the various packet arrivals.

Buffer		*F*(buffer no., packet no.)	Order of transmission
B1	(1) $t = 0, L = 2$ (2) $t = 4, L = 1$	$F(1,1) = R(0) + 2 = 2$ $F(1,2) = R(4) + 1 = 3.5$	1 3
B2	(1) $t = 1, L = 3$ (2) $t = 2, L = 1$	$F(2, 1) = R(1) + 3 = 4$ $F(2,2) = \max\{F(2,1), 2\} + 1 = 5$	2 4
B3	(1) $t = 3, L = 5$	$F(3,1) = R(3) + 5 = 7$	5

The figure below shows the actual packet completion times for each of the packets.

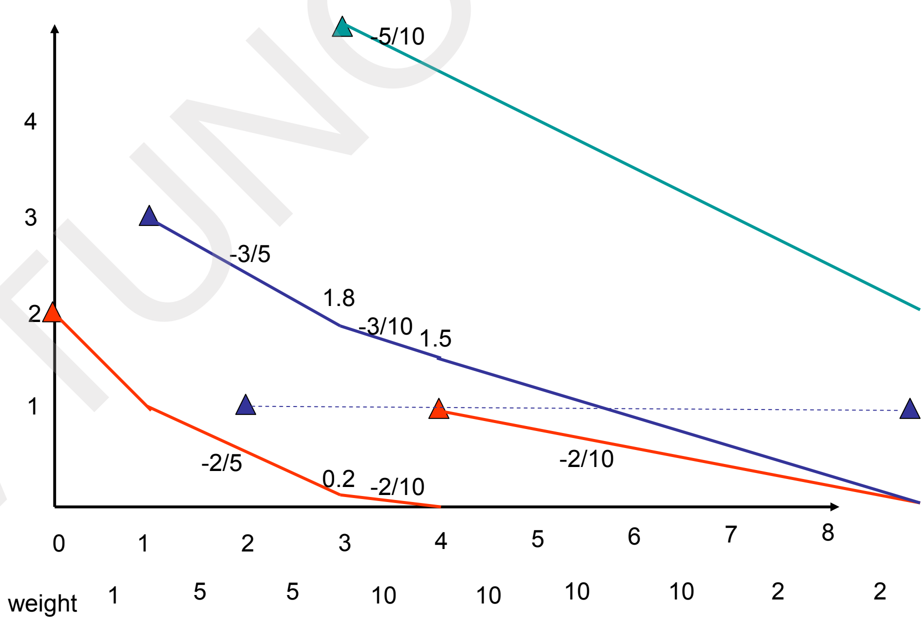


Note the difference in the order of packet completion times in the ideal fluid flow system and the packetized system. The first packet in buffer 2 is able to "sneak in" ahead of the second packet from buffer 1 in the time interval between 2 and 3 seconds.

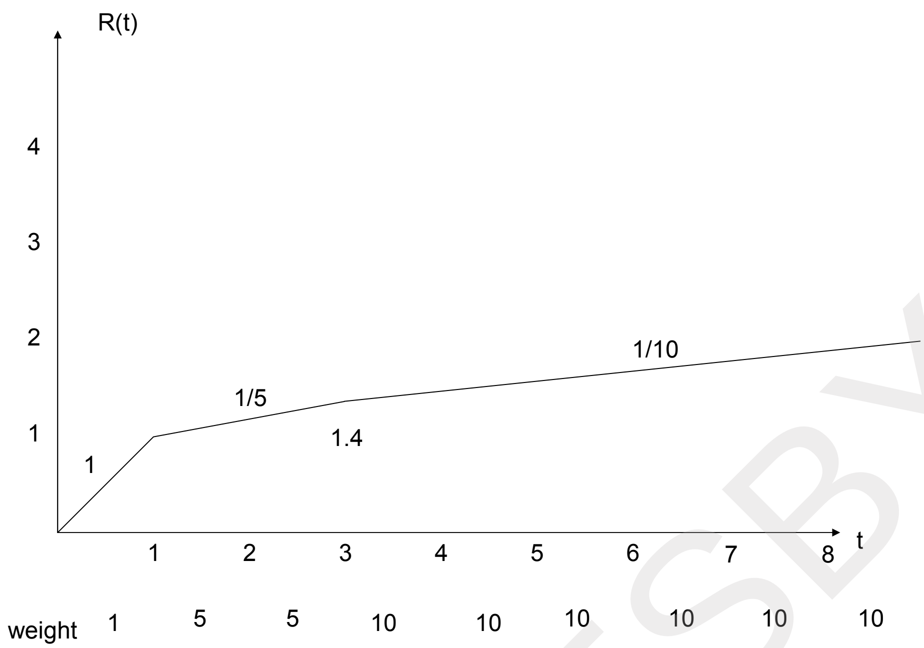
7.52 Repeat problem 7.51 if buffers 1, 2, and 3 have weights 2, 3, and 5, respectively.

Solution:

The following figure shows how the various packet arrivals are serviced in the fluid flow system. It can be seen that the packets from the first and second buffers take much longer to complete service because the lower weighting relative to the packet from buffer 3.



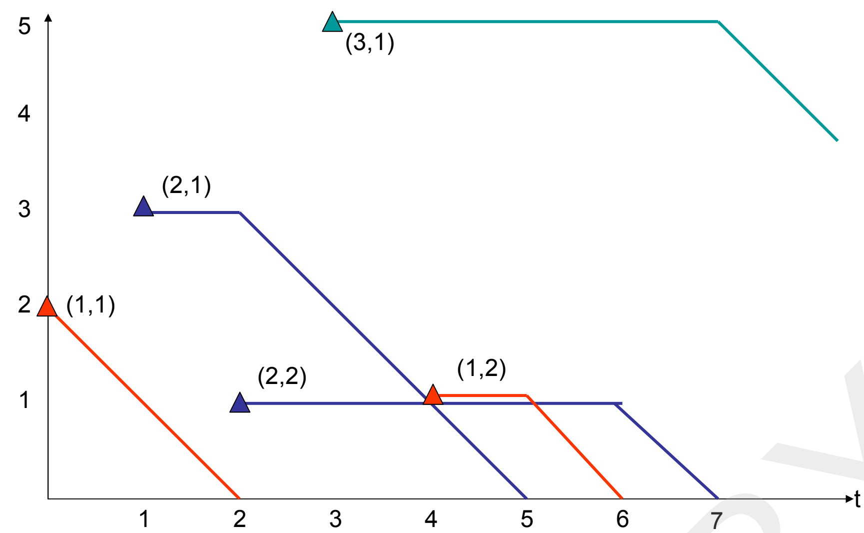
The generalization of the round number in the weighted fair queueing case has a slope that is the reciprocal of the sum of the weights of the active buffers at a given time. The round number vs. time is shown in the next figure.



The packet finish times are given in the table below:

Buffer		*F*(buffer no., packet no.)	Order of transmission
B1 $w = 2$	(1) $t = 0$, $L = 2$ (2) $t = 4$, $L = 1$	$F(1,1) = R(0) + (2/2) = 1$ $F(1,2) = R(4) + (1/2) = 2.$	1 3
B2 $w = 3$	(1) $t = 1$, $L = 3$ (2) $t = 2$, $L = 1$	$F(2, 1) = R(1) + (3/3) = 2$ $F(2,2) = \max\{2, 2\} + (1/3) = 2\ 1/3$	2 4
B3 $w = 5$	(1) $t = 3$, $L = 5$	$F(3,1) = R(3) + (5/5) = 2.4$	5

The following figure shows the actual packet completion times.



7.53. Suppose that in a packet-by-packet weighted fair-queueing system, a packet with finish tag *F* enters service at time *t*. Is it possible for a packet to arrive at the system after time *t* and have a finish tag less than *F*? If yes, give an example. If no, explain why.

Solution:

Yes, the finish tag is a function of the arriving time of the packet to the system, but also the packet length. The early arriving packet may have a very long packet length while the late-arriving packet has a very short packet length. The late-arriving packet would then have a smaller *F* than the earlier packet.

In Problem 7.51, the first packet that arrives at buffer 2 ends up with a later finish time than the second packet that arrives at buffer 1. By the time the latter packet arrives, the former packet has already entered service in the real system.

7.54. Deficit round-robin is a scheduling scheme that operates as follows. The scheduler visits the buffers in round-robin fashion. A deficit counter is maintained for each buffer. When the scheduler visits a buffer, the scheduler adds a quantum of service to the deficit counter, and compares the resulting value to the length of the packet at the head of the line. If the counter is larger, the packet is served and the counter is reduced by the packet length. If not, the deficit is saved for the next visit. Suppose that a system has four buffers and that these contain packets of length 16, 10, 12 and 8 and that the quantum is 4 units. Show the deficit counter at each buffer as a function of time and indicate when the packets are transmitted.

Solution:

	R1	**R2**	**R3**	**R4**
BQ1: *L* = 16 *C* = 0	*C* = 4	*C* = 8	*C* = 12	*L* = 0, Xmit *C* = 0
B2: *L* = 10 *C* = 0	*C* = 4	*C* = 8	*L* = 0, Xmit *C* = 2	*L* = 0 *C* = 0
B3: *L* = 12 *C* = 0	*C* = 4	*C* = 8	*L* = 0, Xmit *C* = 0	*L* = 0 *C* = 0
B4: *L* = 8 *C* = 0	*C* = 4	*L* = 0, Xmit *C* = 0	*L* = 0 *C* = 0	*L* = 0 *C* = 0

The DRR scheduler visits each buffer in turn, adding a quantum of credits to the deficit if the buffer is nonempty. When a buffer has enough credits to transmit a packet, the scheduler sends the packet for transmission. The scheduler resumes visiting the buffers from this point when the packet has completed transmission.

7.55. Should packet-level traffic management be performed in the core of the network where the packet streams have been aggregated to multi-Gigabit/second bit rates? Discuss the alternative of using TDM circuits to carry packet streams in the core network.

Solution:

The purpose of packet-level traffic management is to ensure that packets are provided with the appropriate level of QoS, which typically involves packet delay and packet loss performance. As packet traffic is aggregated into higher bit-rate streams, it becomes more difficult and costly to do packet-level traffic management in the core. However, the much higher bit rates mean that the packet transmissions times are much shorter, and hence queueing delays are shorter even if the number of packets in queue increases. Similarly, larger buffer sizes are affordable for higher bit rates so loss can be kept at relatively low levels. A reasonable approach is to multiplex packet streams into larger streams before entering the core of the network and to then transfer the streams efficiently switched using TDM or optical path circuits, thus eliminating the need for packet-level traffic management.

7.56. Queue management with random early detection (RED):

Solutions follow questions:

(a) Explain why RED helps prevent TCP senders from detecting congestion and slowing down their transmission rates at the same time.

By randomly dropping the packets when the threshold is passed, only a subset of the sources is induced to reduce their traffic rates. Thus as congestion sets in, the community of TCP senders is gradually and progressively directed to slow down their transmission rates.

(b) Discuss the effect of RED on network throughput.

RED maintains a relatively stable network throughput because it drops packets before congestion occurs and because it avoids the oscillations in throughput that can result when a large number of senders change their sending rate in unison. How close this throughput is to the network capacity is dependent to some extent on the min_{th} and max_{th} values set for RED.

(c) Discuss the implementation complexity of the RED algorithm.

In the RED implementation, every queue must maintain two variables: min_{th} and max_{th}. A processing module that continuously takes the running average (to allow for occasional traffic bursts) of the queue length for every queue is also needed. A random number generator of some sort is required to drop packets with probabilities that depend on the running average of the queue length.

(d) Discuss what would happen if instantaneous queue length were used instead of average queue length.

Using average queue length allows RED to tolerate small fluctuations around the threshold level. The use of instantaneous queue length would cause normal buffer fluctuations to cause RED to be initiated more frequently resulting in inefficiencies in the network.

(e) Explore ways to find reasonable values for the RED parameters (i.e., min_{th}, max_{th}, and the packet drop probability when the average queue length reaches max_{th}).

The values of the RED parameters depend on a number of factors. The min_{th} and max_{th} selection depends on available buffer sizes. The difference between the two thresholds should be larger than the typical increase in average length in one round-trip time (RTT). The difference is also related to the transient burst sizes that should be tolerated when congestion occurs. A recommended setup for max_{th} is 2 to 3 times min_{th}. In general min_{th} should be lower if RTT is higher. The dependence on RTT relates the selection of the thresholds to link speed and propagation delay. The selection of min_{th} has an impact on average delay as well. For smaller delays, min_{th} should be smaller however the link utilization may be compromised if the threshold is too small.

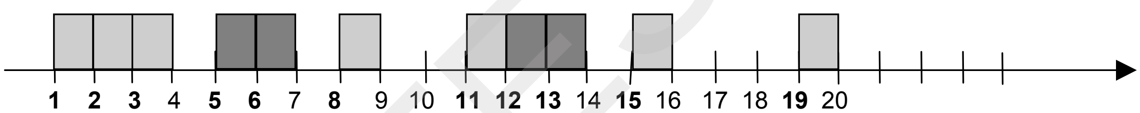
The selection of the packet drop probability should be related to the steady state packet drop in the network.

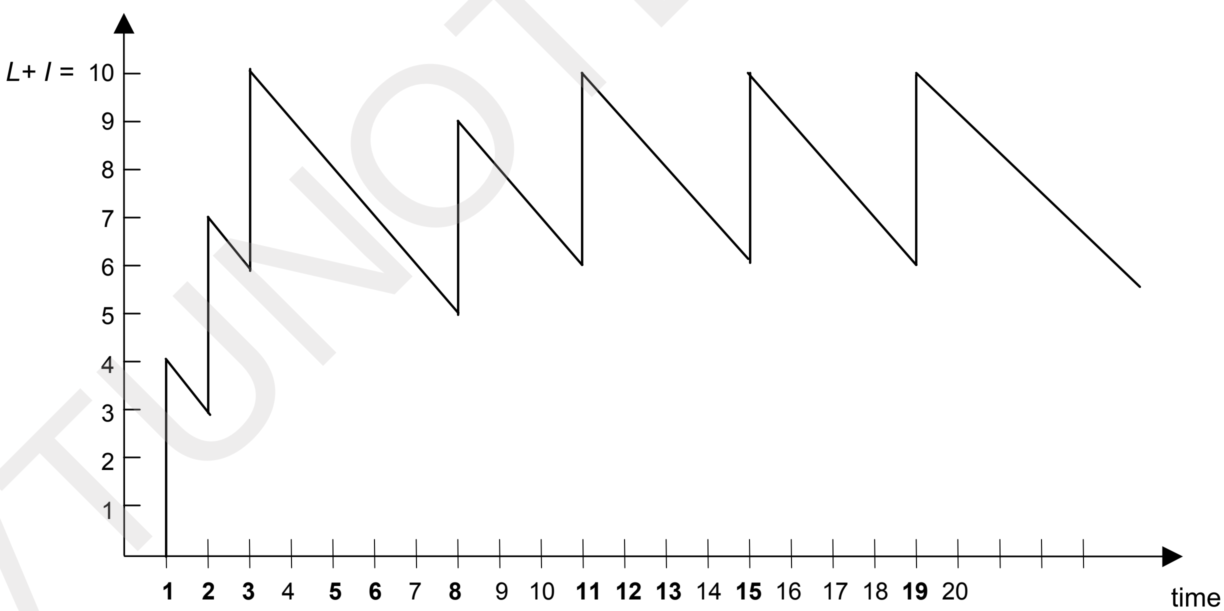
In weighted RED, the weight determines the time constant for the averaging for the average queue size. If the weight is too small the average queue size will respond too slowly. A large weight on the other hand will cause the average to follow the instantaneous queue size more closely and will result I fluctuations.

7.57. Suppose that ATM cells arrive at a leaky bucket policer at times t = 1, 2, 3, 5, 6, 8, 11, 12, 13, 15, and 19. Assuming the same parameters as the example in Figure 7.54, plot the bucket content and identify any nonconforming cells. Repeat if L is reduced to 4.

Solution:

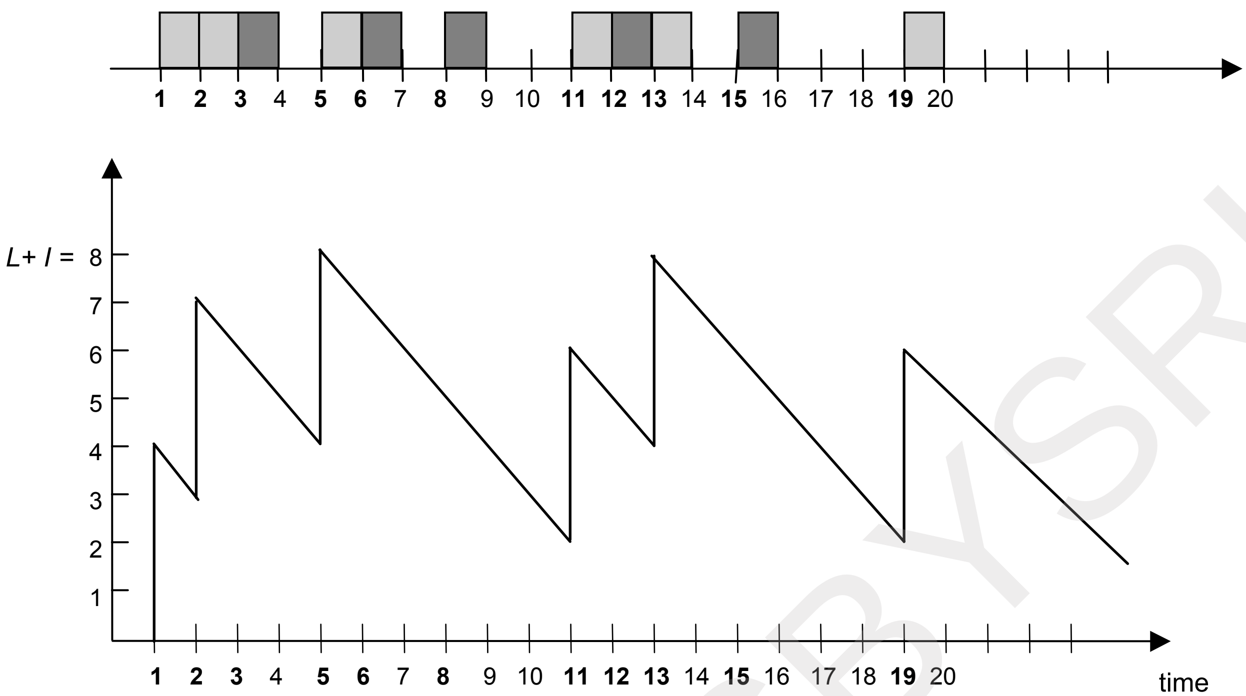
1) I = 4 and L = 6. Drain rate = 1 unit/plot-time.





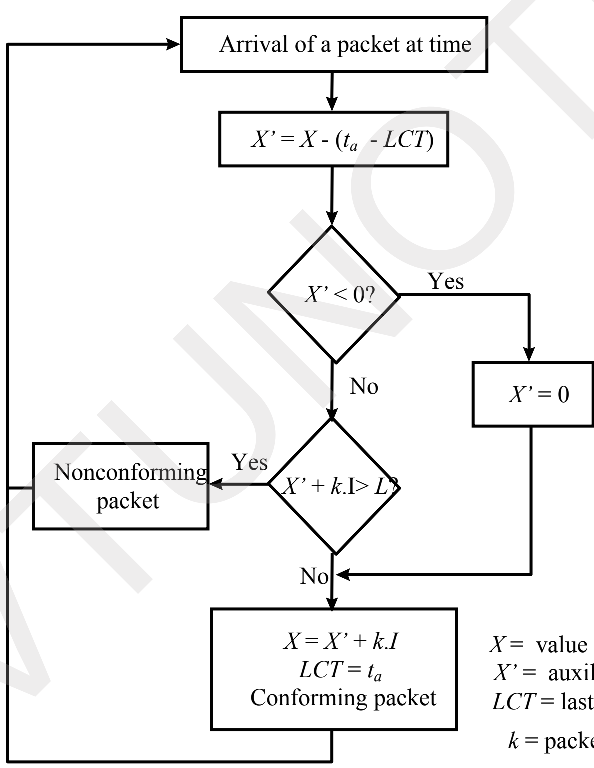
Cells 5, 6, 12, and 13 are nonconforming.

2) I = 4 and L = 4. Drain rate = 1 unit/plot-time.



7.58. Modify the leaky bucket algorithm in Figure 7.53 if packet length is variable.

Solution:



X = value of the leaky bucket counter
X' = auxiliary variable
LCT = last conformance time
k = packet length in terms of service units

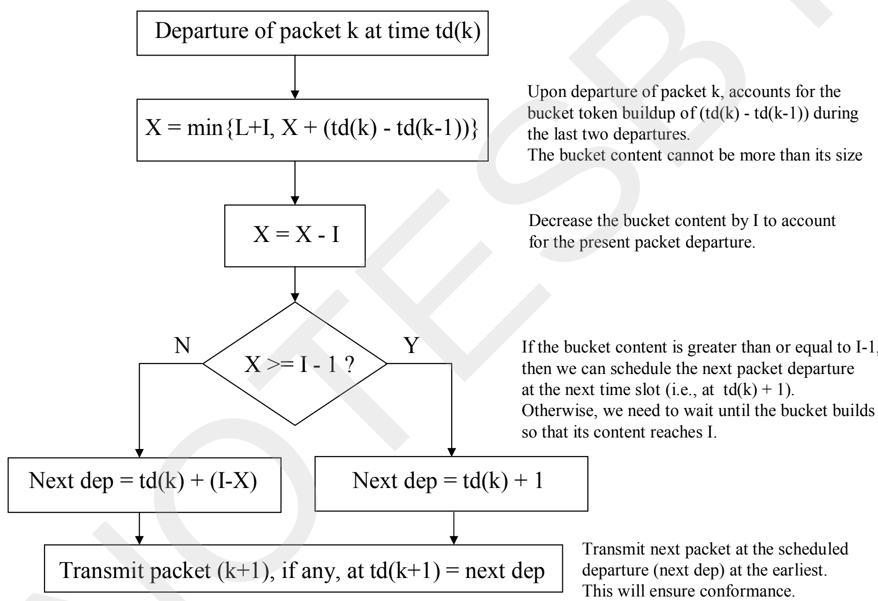
7.59. Explain the difference between the leaky bucket traffic shaper and the token bucket traffic shaper.

Solution:

In the leaky bucket the packet output rate is always constant. In the token bucket, the output rate can be variable and is controlled at one packet per token. If there are tokens in the bucket, the packets are sent out as soon as they arrive. The depth of the token bucket limits the traffic burstiness. If there are no tokens in the bucket, the packet has to wait for the arrival of a token, and the token arrival rate becomes the packet output rate. In the case of the empty token bucket where the tokens arrive at a constant rate, the token bucket essentially becomes a leaky bucket.

7.60. Show the algorithm for token bucket traffic shaper using a flow chart similar to the one shown in Figure 7.53 for policing. The flow chart begins with a departure of a packet k at time $t_d(k)$ and calculates the departure time for packet $k + 1$. Define the necessary parameters and assume that packet length is fixed.

Solution:



7.61. Explain where the policing device and the traffic shaping device should be located in the generic packet switch in Figure 7.19.

Solution:

The traffic policing device should be placed at the input port of the ingress line card. As packets arrive their usage history is updated and the packet if tagged and/or dropped if necessary. The traffic shaping device should be placed in the output port of egress line card, prior to transfer to another switch or network.

7.62. Which of the parameters in the upper bound for the end-to-end delay (Equation 7.24) are controllable by the application? What happens as the bit rate of the transmission links becomes very large?

Solution:

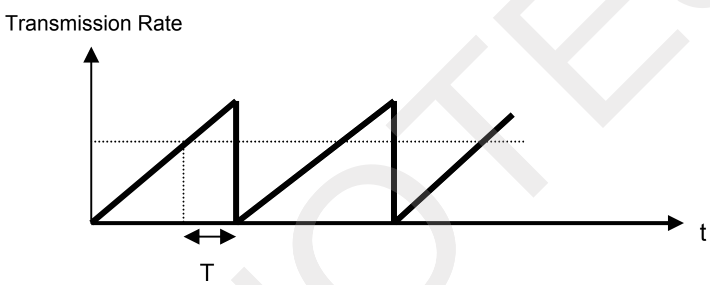
$$D \le \frac{b}{R} + \frac{(H-1)m}{R} + \sum_{j=1}^{H} \frac{M}{R_j}$$

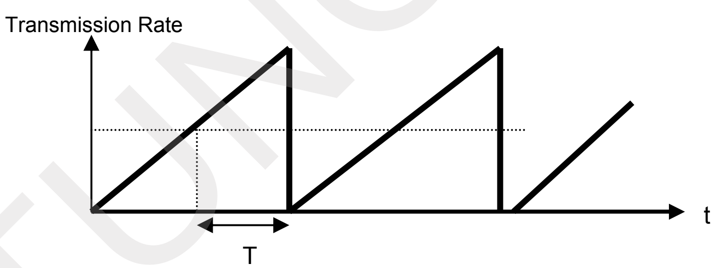
The maximum packet size of the given flow, *m*, can be controlled by the application. The reserved bandwidth allocation *R* for the flow and the maximum burst size *b* can be negotiated. If source routing is used the number of hops can be controlled to a limited extent.

As the transmission link speeds become very large, the last term becomes negligible and the performance is determined by *b, H, R,* and *m.*

7.63. Suppose a source with an unlimited amount of information to transmit uses a closed-loop control to regulate the transmission rate according to feedback information. If the feedback information indicates that there is no congestion, the source continuously increases its transmission rate in a linear fashion. If the feedback information indicates congestion along the path, the source sets its transmission rate to zero and then repeats the cycle by continuously increasing its transmission rate until congestion is detected once again. Assume that it takes *T* seconds for the feedback information to reach the source after congestion occurs. Sketch the transmission rate at the source versus time trajectory for a low and a high value of *T*. Explain how propagation delay *T* plays a role in closed-loop control.

Solution:



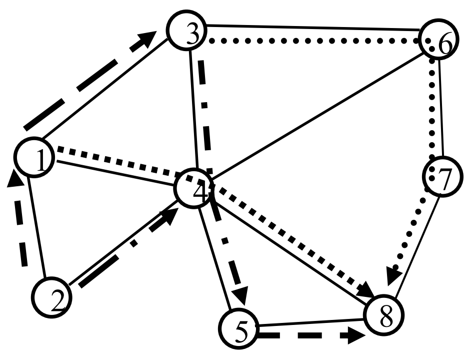


The rate at the source exhibits a sawtooth behavior. The rate starts at 0 and increases linearly until it reaches a peak and drops to 0 again. This is repeated periodically with a frequency which is proportional to the T, as the mechanism has no means of adapting to the available bandwidth. With a larger T the drop in the rate is delayed further from the time that congestion is detected, and so the source cannot respond quickly to the onset to congestion, and indeed may contribute to congestion as well.

7.64. Consider the network in Figure 7.63. Suppose that paths need to be set up in the following order: nodes 5 to 8, 1 to 8, 2 to 4, 3 to 8, 3 to 5, 2 to 1, 1 to 3, 3 to 6, 6 to 7, and 7 to 8. Assume that each link has a capacity of one unit and each path requires bandwidth of one unit.

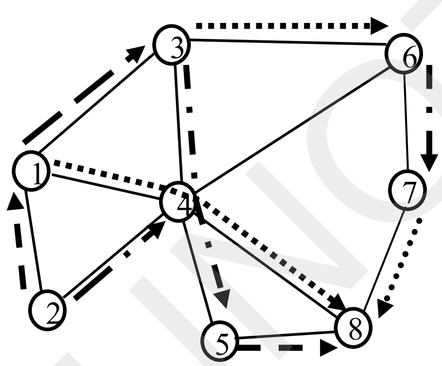
Solutions follow questions:

(a) Use shortest-path routing to set up as many paths as possible. How many paths are blocked?



One connection (3 to 8) is blocked.

(b) Use constraint shortest-path routing to set up as many paths as possible. How many paths are blocked?



Three connections (3 to 6, 6 to 7, and 7 to 8) are blocked.

(c) Can you suggest an improvement to constraint shortest-path routing?

The number of connections that are blocked by each connection is tracked and if it is realized that the connection is blocking too many other connections the connection is rerouted.

Solutions to Chapter 8

8.1. The IP header checksum only verifies the integrity of IP header. Discuss the pros and cons of doing the checksum on the header part versus on the entire packet.

Solution:

Error checking in the header is more important because the packet is routed according to the header information. In addition, the delivery of the data at the destination to the higher layers also requires the header information. Thus error checking of the header protects against misdelivery of the information. Restricting the error checking to the header also simplifies the implementation in the nodes, requires less checksum bits, and prevents unnecessary packet discard. Some higher layers can tolerate some data errors, and higher layers also have the option of performing retransmission.

8.2. Identify the address class of the following IP addresses: 200.58.20.165; 128.167.23.20; 16.196.128.50; 50.156.10.10; 250.10.24.96.

Solution:

An IP address has a fixed length of 32 bits, where the most significant bits identify the particular class. Therefore, to identify the address class we need to convert the dotted-decimal notation back into its binary counterpart, and compare the binary notation to the class prefixes shown in Figure 8.5 in the text. (Recall that the dotted-decimal notation was devised to communicate addresses more readily to other people. In this notation, the 32 bits are divided into four groups of 8 bits – separated by periods – and then converted to their decimal counterpart.) The first few bits (shown in red) of the address can be used to determine the class.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

200.58.20.165
11001000.00111010.00010100.10100101
Class C

128.167.23.20
10000000.10100111.00010111.00010100
Class B

16.196.128.50
00010000.11000100.10000000.00110010
Class A

150.156.10.10
10010110.10011100.00001010.00001010
Class B

250.10.24.96
11111010.00001010.00011000.01100000
Class E

8.3. Convert the IP addresses in Problem 8.2 to their binary representation.

Solution:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

200.58.20.165
11001000.00111010.00010100.10100101

128.167.23.20
10000000.10100111.00010111.00010100

16.196.128.50
00010000.11000100.10000000.00110010

150.156.10.10
10010110.10011100.00001010.00001010

250.10.24.96
11111010.00001010.00011000.01100000

8.4. Identify the range of IPv4 addresses spanned by Class A, Class B, and Class C.

Solution:

The range of IPv4 addresses spanned by each class is:
Class A: 1.0.0.0 to 127.255.255.255
Class B: 128.0.0.0 to 191.255.255.255
Class C: 192.0.0.0 to 223.255.255.255

8.5. What are all the possible subnet masks for the Class C address space? List all the subnet masks in dotted-decimal notation, and determine the number of hosts per subnet supported for each subnet mask.

Solution:

255.255.255.128 supports 126 hosts (not including the broadcast address)

255.255.255.192 supports 62 hosts

255.255.255.224 supports 30 hosts

255.255.255.240 supports 14 hosts

255.255.255.248 supports 7 hosts

255.255.255.252 supports 3 hosts

255.255.255.254 and 255.255.255.255 are not practically usable.

8.6. A host in an organization has an IP address 150.32.64.34 and a subnet mask 255.255.240.0. What is the address of this subnet? What is the range of IP addresses that a host can have on this subnet?

Solution:

Address: 10010110 00100000 01000000 00100010

Mask: 11111111 11111111 11110000 00000000

Subnet: **10010110 00100000 0100**0000 00000000

Host:

From: 10010110 00100000 0100**0000 00000001**

To: 10010110 00100000 0100**1111 11111110**

8.7. A university has 150 LANs with 100 hosts in each LAN.

Solutions follow questions:

(a) Suppose the university has one Class B address. Design an appropriate subnet addressing scheme.

A Class B address has 14 bits for the network ID and 16 bits for the host ID. To design an appropriate subnet addressing scheme we need to decide how many bits to allocate to the host ID versus the subnet ID. We can choose either 7 bits or 8 bits to identify the hosts.

If we allocate 8 bits for to identify the host, as shown below, then there are sufficient subnet-id bits to cover up to 2^8=256 LANs and enough host-id bits to cover up to 256 hosts for each LAN. The subnet mask in this case is 255.255.255.0

1	**0**	Network-id	Subnet-id	Host-id
0	1	15	16 23	24 31

Subnet mask: 255.255.255.0

If we allocate 7 bits for to identify the host, as shown below, then there are sufficient subnet-id bits to cover up to 2^9=512 LANs and enough host-id bits to cover up to 128 hosts for each LAN. The subnet mask in this case is 255.255.255.128.

The choice between 7 or 8 bits to represent the hosts depends on which is likely to grow more, the number of subnets or the number of hosts in a LAN. Alternatively a variable-length prefix scheme using 7-bit host addresses, and grouping these form larger subnets provides greater flexibility in accommodating future changes.

(b) Design an appropriate CIDR addressing scheme.

CIDR addressing scheme involves devising a prefix length that indicates the length of the network mask. In this case, 8 bits are required to identify each LAN (since 127 < 150 < 255) and 7 bits are required to identify each host in each LAN (since 63 < 100 < 127). Therefore a CIDR address would use a 17-bit prefix, and thus have an address of the form address/17.

8.8. A small organization has a Class C address for seven networks each with 24 hosts. What is an appropriate subnet mask?

Solution:

A Class C address requires 21 bits for its network ID, leaving 8 bits for the host ID and subnet ID to share. One possible scheme would assign 4 bits to the host and 4 to the subnet ID, as shown below. The number of bits assigned to the host can be increased to 5 as well.

Network-id	Subnet-id	Host-id
0 … 23	24 … 27	28 … 31

Subnet mask: 255.255.255.224

8.9. A packet with IP address 150.100.12.55 arrives at router R1 in Figure 8.8. Explain how the packet is delivered to the appropriate host.

Solution:

The packet with IP address 150.100.12.55 arrives from the outside network. R1 has to know the next-hop router or host to send the packet to. The address corresponds to the binary string 10010110.01100100.00001100.00110111. R1 knows that a 9 bit subnet field is in use so it applies the following mask to extract the subnetwork address from the IP address.
11111111.11111111.11111111.10000000

The resulting IP address is 10010110.01100100.00001100.00000000 and corresponds 150.100.12.0. This indicates that the host is in subnet 150.100.12.0, so the router transmits the IP packet on this (attached) LAN.

8.10. In Figure 8.8 assign a physical layer address 1, 2, … to each physical interface starting from the top row, moving right to left, and then moving down. Suppose H4 sends an IP packet to H1. Show the sequence of IP packets and Ethernet frames exchanged to accomplish this transfer.

Solution:

1. Send IP packet from H4 to R1:
Source address 150.100.12.55 to destination IP address 150.100.12.176
Source Ethernet 4 to Receive Ethernet 6
2. Forward IP packet from R1 to H1
Source address 150.100.12.55 to destination IP address 150.100.12.176
Source Ethernet 3 to Receive Ethernet 2

8.11. ARP is used to find the MAC address that corresponds to an IP address; RARP is used to find the IP address that corresponds to a MAC address. True or false?

Solution:

True, ARP is used to find the MAC address for a given IP address.

True, Reverse ARP is used by a device to find its IP address given its MAC address.

8.12. Perform CIDR aggregation on the following /24 IP addresses: 128.56.24.0/24; 128.56.25.0./24; 128.56.26.0/24; 128.56.27.0/24.

Solution:

128.56.24.0/$_{22}$ = 10000000.00111000.00011000.00000000
128.56.25.0/$_{22}$ = 10000000.00111000.00011001.00000000
128.56.26.0/$_{22}$ = 10000000.00111000.00011010.00000000
128.56.27.0/$_{22}$ = 10000000.00111000.00011011.00000000
mask = 11111111.11111111.11111100.00000000
The resulting prefix is 128.56.24.0/22.

8.13. Perform CIDR aggregation on the following /24 IP addresses: 200.96.86.0/24; 200.96.87.0/24; 200.96.88.0/24; 200.96.89.0/24.

Solution:

200.96.86.0/$_{20}$ = 11001000.01100000.01010110.00000000
200.96.87.0/$_{20}$ = 11001000.01100000.01010111.00000000
200.96.88.0/$_{20}$ = 11001000.01100000.01011000.00000000
200.96.89.0/$_{20}$ = 11001000.01100000.01011001.00000000
mask = 11111111.11111111.11110000.00000000
The resulting prefix is 200.96.80.0/20.

8.14. The following are estimates of the population of major regions of the world: Africa 900 million; South America 500 million; North America 400 million; East Asia 1500 million; South and Central Asia 2200 million; Russia 200 million; Europe 500 million.

Solutions follow questions:

(a) Suppose each region is to be assigned 100 IP addresses per person. Is this possible? If not, how many addresses can be assigned per person? Repeat for IPv6.

The total number of IPv4 addresses is: 2^{32} which is approximately 4.29 billion. The above world population estimate totals 6.2 billion, so it is not possible to assign an individual address to each person. IPv6 is required to provide 100 addresses per person.

(b) Design an appropriate CIDR scheme to provide the addressing in part (a).

	(millions)	(millions)	IPV6 128 bits	
Regions	**Population**	**IP/Person**	**Net-id**	**Host-id**
Africa	900	90000	91	37
South America	500	50000	92	36
North America	400	40000	92	36
East Asia	1500	150000	90	38
South C. Asia	2200	220000	90	38
Russia	200	20000	93	35
Europe	500	50000	92	36

8.15. Suppose four major ISPs were to emerge with points of presence in every major region of the world. How should a CIDR scheme treat these ISPs in relation to addressing for each major region?

Solution:

The networks of these ISPs will span across national and geographical boundaries and will connected in a non-hierarchical manner. These large global ISPs constitute major transit routing domains, so it makes sense to assign them blocks of unique IP addresses and to require that domains attached to them should begin with the transit domain's prefix. The blocks of addresses should be managed so that fragmentation of the CIDR block does not take place. For example, when a customer of the ISP changes to another ISP, the customer is required to return the block of addresses. This policy enables CIDR to be effective in controlling the size of routing tables.

8.16. Discuss the difficulties with using actual time in the TTL field.

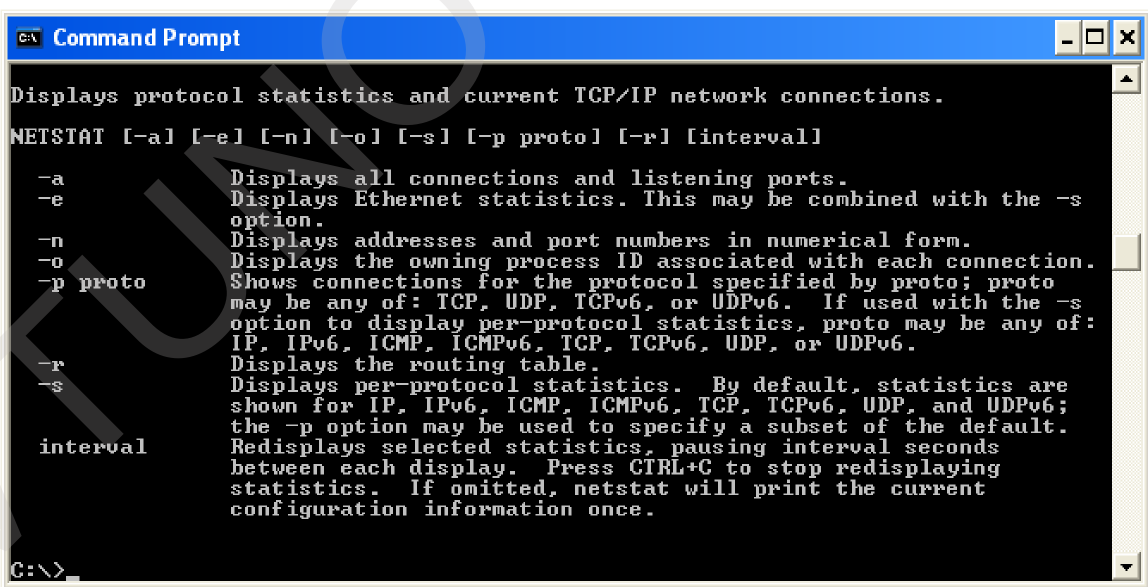
Solution:

Unlike the number of hops, which is predictable if a packet is routed correctly, the actual time that it takes to go through the route is not predictable. Therefore the amount of time that a packet stays in the network is not necessarily an indication of misrouting. To allow an upper limit for delay across the network TTL field would become a very large number. It is also more complex to track and update the TTL according to actual time as the packet traverses the network.

8.17. Lookup the `netstat` command in the manual for your system. Use the command to display the routing table in your host. Try the different command options.

Solution:

The exact command depends on your computing environment. In a DOS (Windows) environment, the command may be `netstat -r`. In a SUN/UNIX environment, the command may be `netstat -a`. See the answer to problem 45 in Chapter 1 for an example routing table. The figure below shows a screen capture of the parameters for the command in Windows XP.



8.18. Suppose a router receives an IP packet containing 600 data bytes and has to forward the packet to a network with maximum transmission unit of 200 bytes. Assume that the IP header is 20 bytes long. Show the fragments that the router creates and specify the relevant values in each fragment header (i.e., total length, fragment offset, and more bit).

Solution:

Given:

IP packet = 600 data bytes
MTU = 200 bytes
IP header = 20 header bytes
Maximum possible data length per fragment = MTU – IP header = 200 – 20 = 180 bytes.
The data length of each fragment must be a multiple of eight bytes; therefore the maximum number of data bytes that can be carried per fragment is 22*8=176.

The data packet must be divided into 4 frames, as shown by the following calculations:

176 + 176 + 176 + 72 = 600
<u>20 + 20 + 20 + 20</u>
196 196 196 92

The sequence of frames and packet headers is shown below:

	Total length	Id	Mf	Fragment Offset
Original Packet	620	x	0	0
Fragment 1	196	x	1	0
Fragment 2	196	x	1	22
Fragment 3	196	x	1	44
Fragment 4	92	x	0	66

8.19. Design an algorithm for reassembling fragments of an IP packet at the destination IP.

Solution:

I. Set data = Null
II. Verify that all fragments for id = x have arrived
III. Sort fragments in ascending order based on fragment offset
IV. For each fragment starting with fragment offset = 0, move data = data + data-in-fragment$_I$
V. Data contains the reassembled information

8.20. Does it make sense to do reassembly at intermediate routers? Explain.

Solution:

No, because the packet may be de-fragmented again, and all the time required to wait for all fragments and to reassemble the packet will be wasted. Also it is not guaranteed that all fragments go through the same path and arrive at the same node in a datagram network such as IP.

8.21. Describe the implementation issues of IPv4 packet processing.

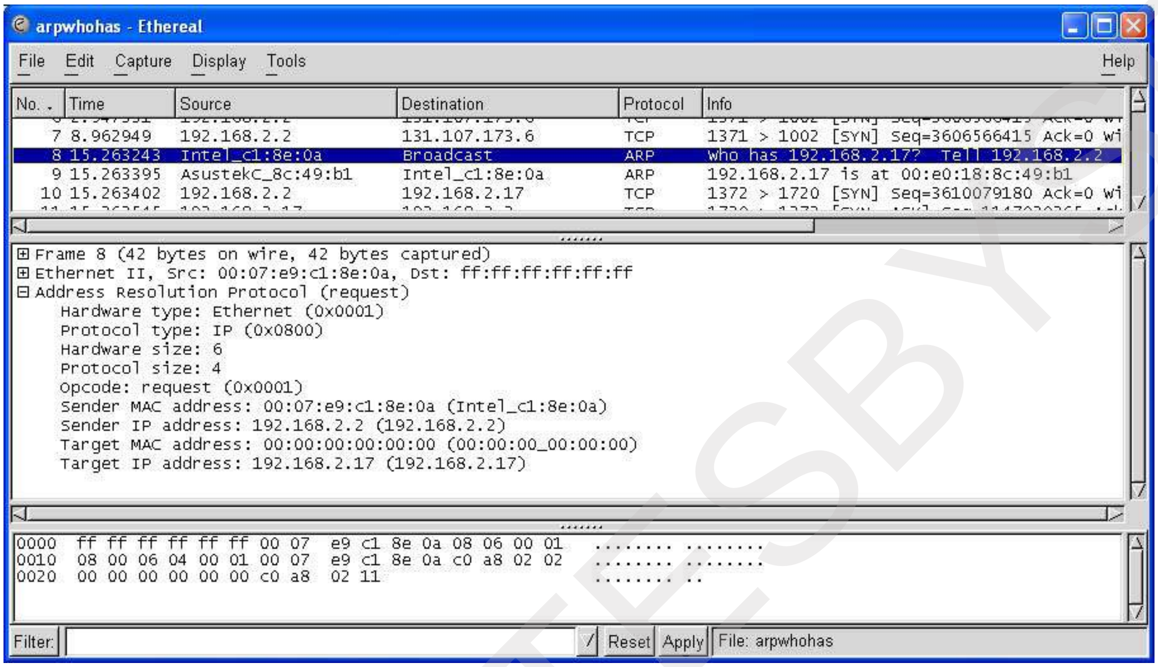
Solution:

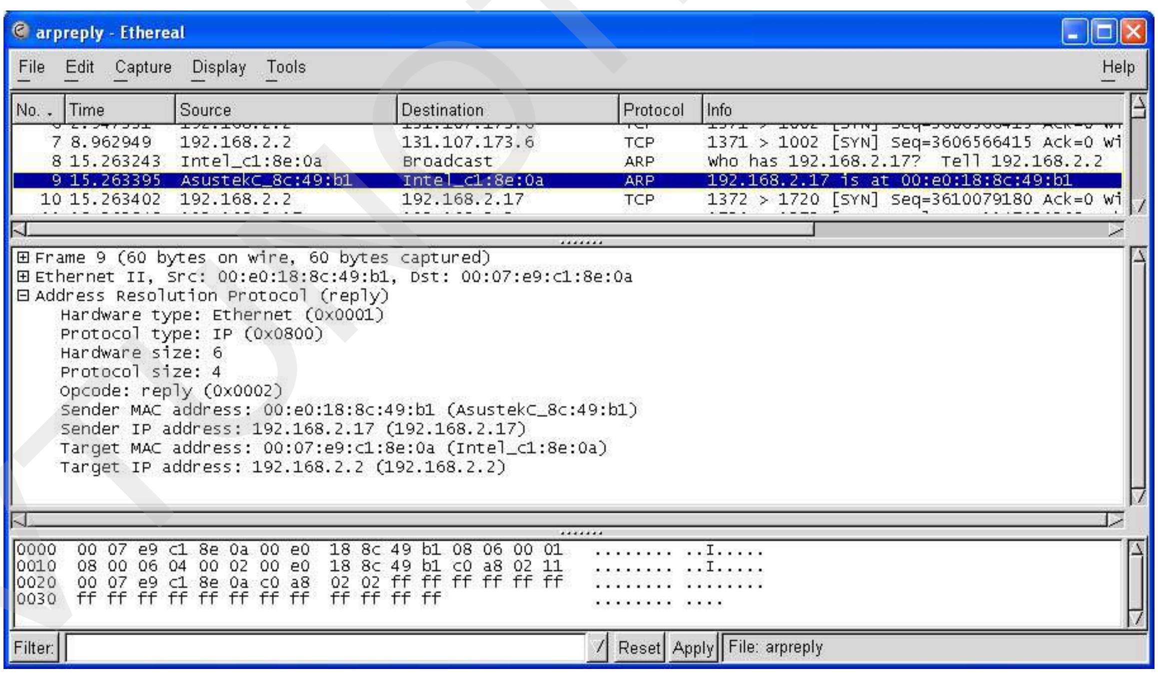
The router performs error checking: the header checksum is computed; the version and total length fields are checked for valid values. The router finds the next-hop by consulting its routing table. The router then updates various fields, including the TTL and checksum fields. The packet is then forwarded.

8.22. Use Ethereal to capture ARP packets to find the MAC addresses in a LAN.

Solution:

The following screen captures show and ARP request and the corresponding ARP reply.





8.23. Abbreviate the following IPv6 addresses:

Solutions follow questions:

(a) 0000:0000:0F53:6382:AB00:67DB:BB27:7332

::F53:6382:AB00:67DB:BB27:7332

(b) 0000:0000:0000:0000:0000:0000:004D:ABCD

::4D:ABCD

(c) 0000:0000:0000:AF36:7328:0000:87AA:0398

::AF36:7328:0:87AA:398

(d) 2819:00AF:0000:0000:0000:0035:0CB2:B271

2819:AF::35:CB2:B271

8.24. What is the efficiency of IPv6 packets that carry 10 ms of 64 kbps voice? Repeat if an IPv6 packet carries 1 frame of 4 Mbps MPEG2 video, assuming a frame rate of 30 frames/second.

Solution:

10 ms of 64 kbps voice = $10 \times 10^{-3} \times 64 \times 10^{3}$ = 640 bits = 80 bytes
Header = 40 bytes; Efficiency = 80/(80 +40) = 2/3 = 0.6666 = 66.7%.

1 frame of video is: $4 \times 10^{6}/30$ = 133,333 bits = 16666 bytes
Efficiency = 16666/(16666+40) = 99.76%.

8.25. Why does IPv6 allow fragmentation at the source only?

Solution:

The task of fragmenting a packet uses processing resources in a router. By requiring that all fragmentation be done at the source, routers are relieved of the fragmentation processing load, and hence they can operate faster on the basic routing task.

8.26. Assuming the population estimates in problem 8.14, how many IP addresses does IPv6 provide per capita?

Solution:

Based on the estimates in problem 10, there are $6{,}200 \times 10^{6}$ humans which in turn means 5.4×10^{28} IPv6 addresses per capita.

8.27. Suppose that IPv6 is used over a noisy wireless link. What is the effect of not having header error checking?

Solution:

The transmission over the noisy wireless link will introduce errors in the transmitted frames. If the frames do not contain error-checking, then assuming the frame is recognizable, erroneous packets may be passed to the router and unpredictable behavior may ensue. However, error checking (and retransmission) is included in most noisy wireless links; thus the effect of transmission errors is to trigger retransmission of frames in the link layer. Only packets that arrive in frames that pass error-checking are transferred to the IP layer.

8.28. Explain how the use of hierarchy enhances scalability in the following aspects of Internet:

Solutions follow questions:

(a) Domain name system

The use of hierarchy helps to speed up the translation of a domain name into an internet address. The search starts from the highest level (for example, .com, .org, .net) and eventually down to the specific hostname information. The hierarchy also helps to organize the database architecture of the DNS. Moreover, with a hierarchical system, it is not necessary for each DNS server to contain every single domain name in the network. There are different levels of DNS servers each containing the essential information for its own domain.

(b) IP addressing

Classful IP addressing uses hierarchy to arrange the address space in several discrete classes of addresses that correspond to networks of different sizes. CIDR IP addressing uses a variable-length prefix and a subnet mask to represent networks at a finer granularity of network size. In doing so, CIDR addressing increases the utilization of the address space. When combined with address allocation policies that aggregate routes, CIDR makes it possible to reduce the size of the routing tables required in router.

(c) OSPF routing

OSPF uses a two-level hierarchy that allows an AS to be partitioned into several groups called areas each interconnected by a central backbone area. This localization reduces the amount of routing information that needs to be maintained by individual routers. It also reduces the number of routing messages that need to be exchanged within the network.

(d) Interdomain routing

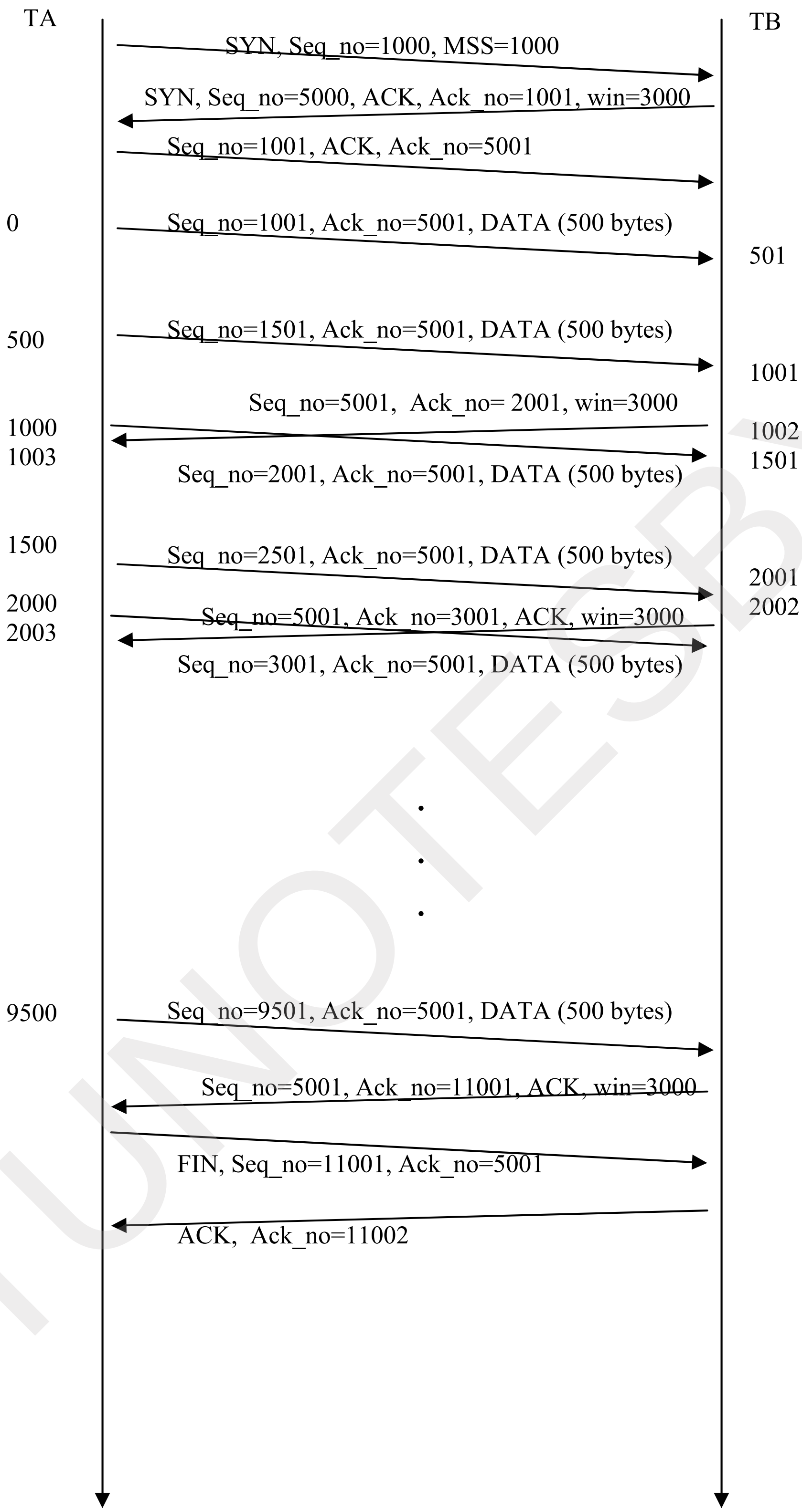
Interdomain routing uses the Border Gateway Protocol (BGP) to exchange routing information between AS's that in turn allows IP packets to flow across the AS border. Thus interdomain routing enables the scalability of the overall Internet by enabling various AS's to become interconnected.

8.29. The TCP in station A sends a SYN segment with ISN = 1000 and MSS = 1000 to station B. Station B replies with a SYN segment with ISN = 5000 and MSS = 500. Suppose station A has 10,000 bytes to transfer to B. Assume the link between stations A and B is 8 Mbps and the distance between them is 200 m. Neglect the header overheads to keep the arithmetic simple. Station B has 3000 bytes of buffer available to receive data from A. Sketch the sequence of segment exchanges, including the parameter values in the segment headers, and the state as a function of time at the two stations under the following situations:

Solutions follow questions:

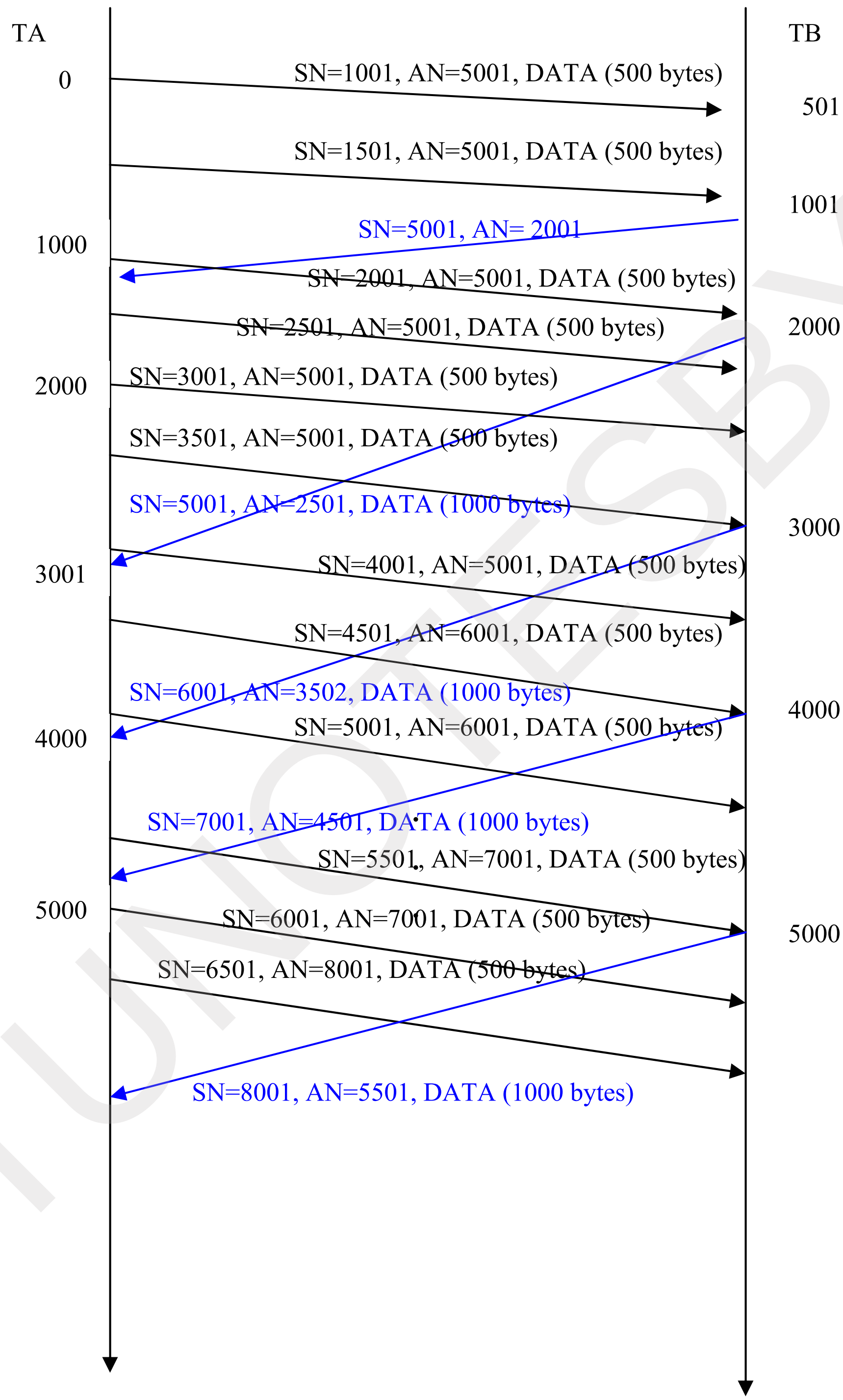
(a) Station A sends its first data segment at $t = 0$. Station B has no data to send and sends an ACK segment every other frame.

At a transmission rate of 8 megabits per second, a single byte has a transmission time of 8 bits / 8×10^6 bits/second = 1 microsecond. A distance of 200 meters in optical fiber has a propagation time of 200 meters / 2×10^8 meters/second = 1 microsecond. Therefore a segment of 500 bytes requires 501 microseconds to arrive completely at the receiver. In the following we also assume that the send window is replenished by the receiver as soon as it receives a segment. The time scale below is in microseconds.


TA
TB
SYN, Seq_no=1000, MSS=1000
SYN, Seq_no=5000, ACK, Ack_no=1001, win=3000
Seq_no=1001, ACK, Ack_no=5001
0
Seq_no=1001, Ack_no=5001, DATA (500 bytes)
501
500
Seq_no=1501, Ack_no=5001, DATA (500 bytes)
1001
Seq_no=5001, Ack_no= 2001, win=3000
1000
1003
1002
1501
Seq_no=2001, Ack_no=5001, DATA (500 bytes)
1500
Seq_no=2501, Ack_no=5001, DATA (500 bytes)
2001
2000
2003
Seq_no=5001, Ack_no=3001, ACK, win=3000
2002
Seq_no=3001, Ack_no=5001, DATA (500 bytes)
9500
Seq_no=9501, Ack_no=5001, DATA (500 bytes)
Seq_no=5001, Ack_no=11001, ACK, win=3000
FIN, Seq_no=11001, Ack_no=5001
ACK, Ack_no=11002

(b) Station A sends its first data segment at t=0. Station B has 6000 bytes to send, and it sends its first data segment at t = 2 ms.

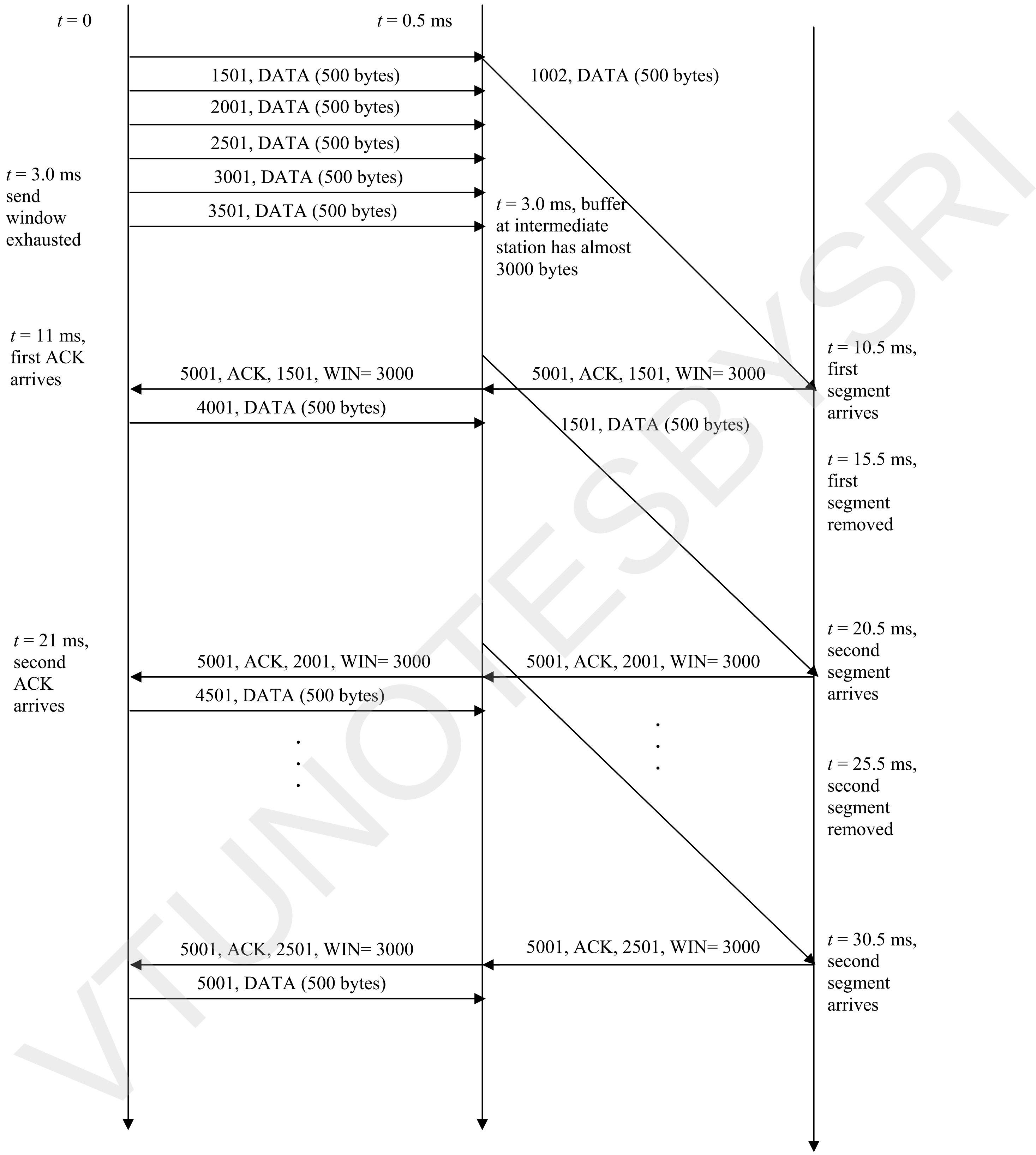
The main feature of this problem is that the acknowledgments are delayed longer because of the long segments that are transmitted from B to A.



8.30. Suppose that the TCP in station A sends information to the TCP in station B over a two-hop path. The data link in the first hop operates at a speed of 8 Mbps, and the data link in the second hop operates at a speed of 400 kbps. Station B has a 3 kilobyte buffer to receive information from A, and the application at station B reads information from the receive buffer at a rate of 800 kbps. The TCP in station A sends a SYN segment with ISN = 1000 and MSS = 1000 to station B. Station B replies with a SYN segment with ISN = 5000 and MSS = 500. Suppose station A has 10,000 bytes to transfer to B. Neglect the header overheads to keep the arithmetic simple. Sketch the sequence of segment exchanges, including the parameter values in the segment headers, and the state as a function of time at the two stations. Show the contents of the buffers in the intermediate switch as well as at the source and destination stations.

Solution:

It takes 500 microseconds to transmit 500 bytes from station A to the intermediate station, but it takes 10 milliseconds to send the same 500 bytes from the intermediate station to station B. Hence, segments will accumulate at the intermediate station until station A exhausts its send window of 3000 bytes. Eventually station A will receive acknowledgments that allow it to resume transmission. Note that the rate at which acknowledgments are returned to station A is controlled by the rate at which segments arrive at station B from the bottleneck at the intermediate node.


t = 0
t = 0.5 ms
1501, DATA (500 bytes)
1002, DATA (500 bytes)
2001, DATA (500 bytes)
2501, DATA (500 bytes)
t = 3.0 ms send window exhausted
3001, DATA (500 bytes)
3501, DATA (500 bytes)
t = 3.0 ms, buffer at intermediate station has almost 3000 bytes
t = 11 ms, first ACK arrives
t = 10.5 ms, first segment arrives
5001, ACK, 1501, WIN= 3000
5001, ACK, 1501, WIN= 3000
4001, DATA (500 bytes)
1501, DATA (500 bytes)
t = 15.5 ms, first segment removed
t = 21 ms, second ACK arrives
t = 20.5 ms, second segment arrives
5001, ACK, 2001, WIN= 3000
5001, ACK, 2001, WIN= 3000
4501, DATA (500 bytes)
t = 25.5 ms, second segment removed
5001, ACK, 2501, WIN= 3000
5001, ACK, 2501, WIN= 3000
t = 30.5 ms, second segment arrives
5001, DATA (500 bytes)

8.31. Suppose that the delays experienced by TCP segments traversing the network is equally likely to be any value in the interval [50 ms, 75 ms]. (See Equations 5.17 to 5.20.)

Solutions follow questions:

(a) Find the mean and standard deviation of the delay.

The delay lies between interval [50ms, 75ms] and is a uniform random variable. The mean is:

$E[X] = (50 + 75) / 2 = 62.5$ ms.

The standard deviation of delay is:

$STD[x] = VAR[x]^{1/2} = [(75 - 50)^2 / 12]^{1/2} = 7.217$

(b) Most computer languages have a function for generating uniformly distributed random variables. Use this function in a short program to generate random times in the above interval. Also, calculate t_{RTT} and d_{RTT} and compare to part (a).

See below for sample program written in C.

```
/* Communication Networks - Chapter 8    */
/* Question 25 (b)                       */
/* Description - Generate a random value */
/* between 50 to 75 ms.  Calculate t_RTT */
/* and d_RTT                             */
/* The min, max and avg value of t_RTT   */
/* and d_RTT are also recorded           */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main (void)
{
  int i;
  float temp, t_n, t_rtt_new, t_rtt_old;
  float d_rtt_new, d_rtt_old;
  float t_rtt_min, t_rtt_sum, t_rtt_max;
  float d_rtt_min, d_rtt_sum, d_rtt_max;

  const float alpha = 0.875;
  const float beta = 0.25;

  srand (time(NULL));
  t_rtt_old = 0;
  d_rtt_old = 0;

  t_rtt_sum = t_rtt_max = 0;
  d_rtt_sum = d_rtt_max = 0;

  t_rtt_min = d_rtt_min = 500;

  for (i = 0; i < 500; i++)
  {

    /* Generate a random value between 0 to 1 */
    temp = (float) rand() / RAND_MAX;

    /* Scale the random value to fit between 50 to 75 */
    t_n = temp * 25 + 50;
```

```
    /* Calculate t_RTT and d_RTT */
    t_rtt_new = (alpha * t_rtt_old) + ((1 - alpha) * t_n);
    d_rtt_new = (beta * d_rtt_old) + ((1 - beta) * fabs (t_n - t_rtt_old));

    if (t_rtt_new < t_rtt_min)
      t_rtt_min = t_rtt_new;
    if (t_rtt_new > t_rtt_max)
      t_rtt_max = t_rtt_new;

    if (d_rtt_new < d_rtt_min)
      d_rtt_min = d_rtt_new;
    if (d_rtt_new > d_rtt_max)
      d_rtt_max = d_rtt_new;

    t_rtt_sum += t_rtt_new;
    d_rtt_sum += d_rtt_new;

    printf ("t_RTT: %f d_RTT: %f\n", t_rtt_new, d_rtt_new);
    t_rtt_old = t_rtt_new;
    d_rtt_old = d_rtt_new;
  }
  printf ("t_RTT min: %f t_RTT max: %f t_RTT avg: %f\n",
           t_rtt_min, t_rtt_max, (t_rtt_sum / 500.0));
  printf ("d_RTT min: %f d_RTT max: %f d_RTT avg: %f\n",
           d_rtt_min, d_rtt_max, (d_rtt_sum / 500.0));
}
```

We ran the preceding program and obtained the average values of t_{RTT} = 61.6924 and d_{RTT} = 7.1139. These values are averaged from a sample of 500 values.

8.32. Suppose that the advertised window is 1 Mbyte long. If a sequence number is selected at random from the entire sequence number space, what is the probability that the sequence number falls inside the advertised window?

Solution:

If the sequence number field is 32 bits in length and the advertised window is 1Mbyte long, the probability that the sequence number falls inside the advertised window is:

$P = (1 \times 10^6) / 2^{32} = 2.33 \times 10^{-4}$

8.33. Explain the relationship between advertised window size, RTT, delay-bandwidth product, and the maximum achievable throughput in TCP.

Solutions follow questions:

(a) Plot the maximum achievable throughput versus delay-bandwidth product for an advertised window size of 65,535 bytes.

First consider delay-bandwidth product, $DBP = R*2t_p$. Here delay $2t_p$ is the propagation time that elapses from when a bit is sent by a source to the destination to when the bit can be returned back to the source. This is the minimum time that elapses from when a packet leaves a source to when the acknowledgment is received. The delay-bandwidth product *DBP* is then the number of bits (or bytes) that are in the network when the source transmits continuously at the maximum rate and when the bits return immediately back to the source.

The round-trip time *RTT* is the time that actually elapses from when a packet is sent to when its acknowledgment is received. *RTT* includes not only the propagation delay, but also queueing and processing delays. The advertised window, *W,* places a limit on the amount of information that a source can have outstanding in the network.

Consider the time from when a byte leaves the source to when its acknowledgment is received (that is, consider a *RTT*). In that time, the source will have transmitted at most a window-full of bytes into the network. Therefore the window size divided by the *RTT* places a limit on the throughput *r*, that is, the rate at which information can be transmitted into the network: $r < W/RTT$.

The throughput cannot exceed the maximum bit rate $R = DBP/2t_p$ that is available for the source to transmit into the network. Therefore, the throughput increases as the window size is increased, but cannot exceed the bit rate *R*:

Throughput = $r = \min\{R, W/RTT\} = \min\{DBP/2{*}t_p, W/RTT\}$

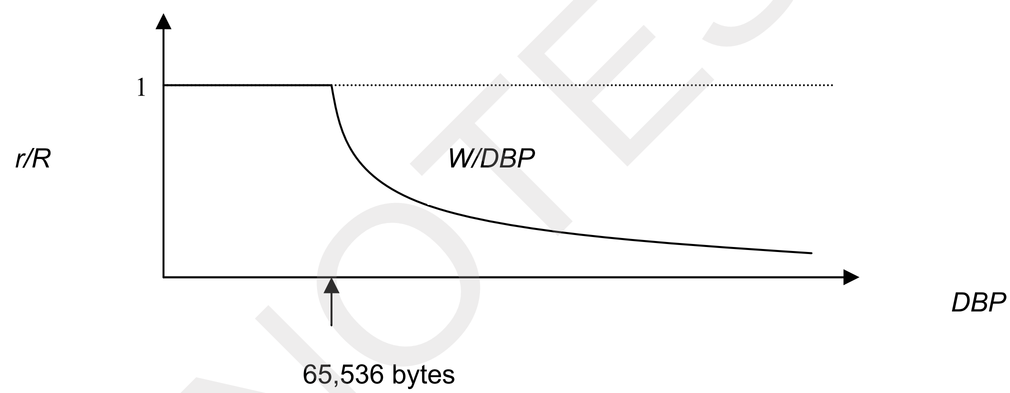
Suppose that the window size is less than the delay-bandwidth product. We then expect that the source cannot transmit at the maximum bit rate *R*. Indeed, we have that:

$r < W/RTT < W/2t_p$.

Therefore we have that:

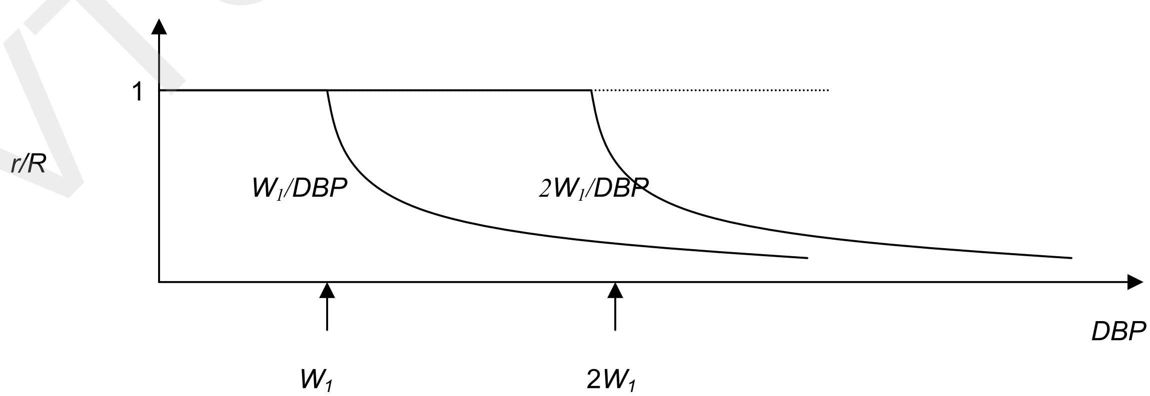
$r/R < W/(R{*}2t_p) = W/DBP$.

We conclude that the ratio of the maximum achievable throughput to *R* is less than the ratio of the window size to the *DBP*, as shown in the figure below.



(b) In the above plot include the maximum achievable throughput when the above window size is scaled up by a factor of $2K$, where $K = 4, 8, 12$.

The following figure shows the case where the window size is doubled.



(c) Place the following scenarios in the plot obtained in part (b): Ethernet with 1 Gbps and distance 100 meters; 2.4 Gbps and distance of 6000 km; satellite link with speed of 45 Mbps and RTT of 500 ms; 40 Gbps link with distance of 6000 km.

Case	***DBP* implied**
Ethernet R = 1 Gbps D = 100 m	$T_p = 100 / 2.5\text{x}10^8$ $T_p = 4\text{x}10^{-7}$ $DBP = 2{*}T_p{*}R$ $DBP = 8\text{x}10^{-7} * 1\text{x}10^9$ $DBP = 8 \text{ x } 10^{2}$ ***DBP* = 800 bits**
Link R = 2.4 Gbps D = 6000 km	$T_p = 6 \text{ x } 10^3 / 2.5\text{x}10^5$ $T_p = 2.4 \text{ x } 10^{-2}$ $DBP = 2{*}T_p{*}R$ $DBP = 2 * 2.4 \text{ x}10^{-2} * 2.4\text{x}10^9$ $DBP = 11.52 \text{ x } 10^7$ ***DBP* = 115.2 Mbits (14.4 Mbytes)**
Satellite link R = 45 Mbps RTT = 500 ms ($5\text{x}10^{-1}$ sec)	$DBP = RTT{*}R$ $DBP = 5\text{x}10^{-1} * 45\text{x}10^6$ $DBP = 225 \text{ x } 10^5$ bits ***DBP* = 22.5 Mbits (2.85Mbytes)**
link R = 40 Gbps D = 6000 km ($6\text{x}10^6$ m)	$T_p = 6 \text{ x } 10^3 / 2.5\text{x}10^5$ $T_p = 2.4 \text{ x } 10^{-2}$ $DBP = 2{*}T_p{*}R$ $DBP = 2 * 2.4 \text{ x}10^{-2} * 40\text{x}10^9$ $DBP = 192 \text{ x } 10^7$ ***DBP* = 1.92 Gbits (240 Mbytes)**

8.34. Consider the three-way handshake in TCP connection setup.

Solutions follow questions:

(a) Suppose that an old SYN segment from station A arrives at station B, requesting a TCP connection. Explain how the three-way handshake procedure ensures that the connection is rejected.

In a three-way handshake procedure, one must ensure the selection of the initial sequence number is always unique. If station B receives an old SYN segment from A, B will acknowledge the request based on the old sequence number. When A receives the acknowledgment segment from B, A will find out that B received a wrong sequence number. A will discard the acknowledgment packet and reset the connection.

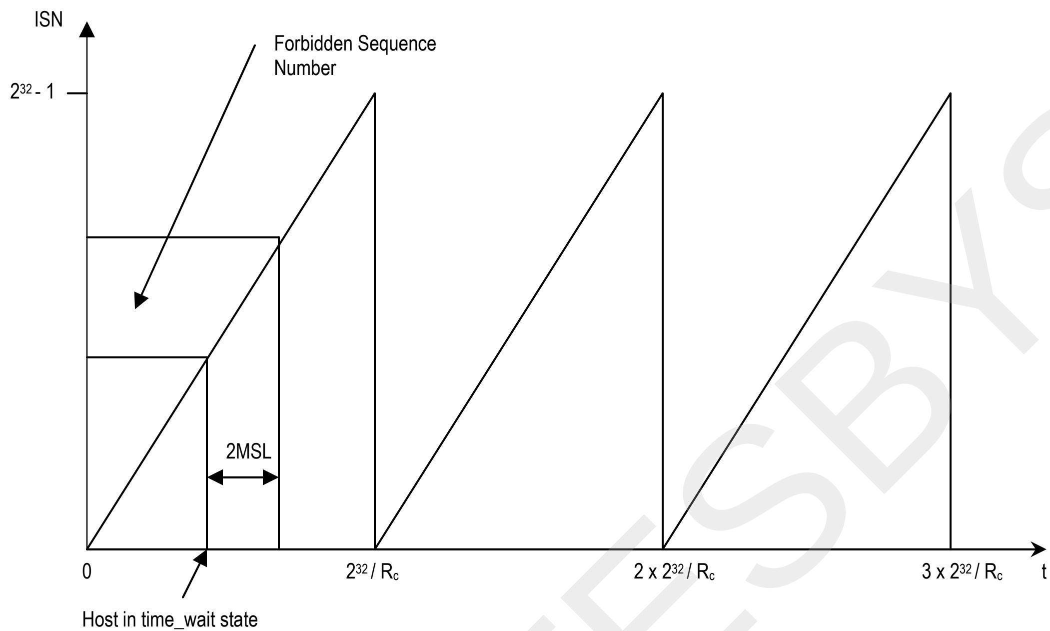
(b) Now suppose that an old SYN segment from station A arrives at station B, followed a bit later by an old ACK segment from A to a SYN segment from B. Is this connection request also rejected?

If an old SYN segment from A arrives at B, followed by an old ACK segment from A to a SYN segment from B, the connection will also be rejected. Initially, when B receives an old SYN segment, B will send a SYN segment with its own distinct sequence number set by itself. If B receives the old ACK from A, B will notify A that the connection is invalid since the old ACK sequence number does not match the sequence number previously defined by B. Therefore, the connection is rejected.

8.35. Suppose that the Initial Sequence Number (ISN) for a TCP connection is selected by taking the 32 low-order bits from a local clock.

Solutions follow questions:

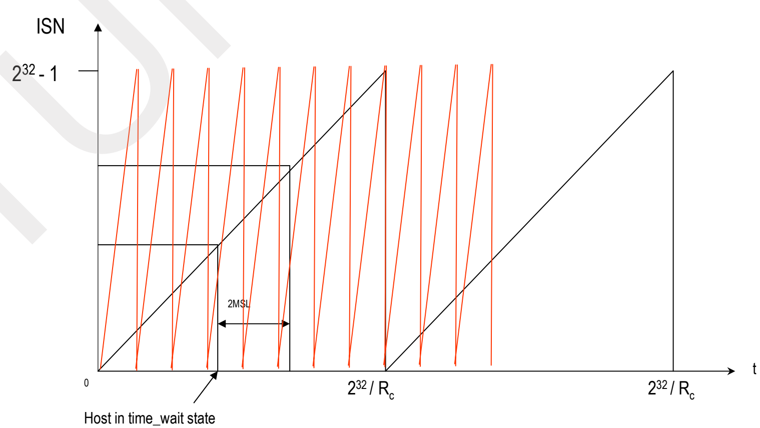
(a) Plot the ISN versus time assuming that the clock ticks forward once every $1/R_c$ seconds. Extend the plot so that the sequence numbers wrap around.



(b) To prevent old segments from disrupting a new connection, we forbid sequence numbers that fall in the range corresponding to 2MSL seconds prior to their use as an ISN. Show the range of forbidden sequence numbers versus time in the plot from part (a).

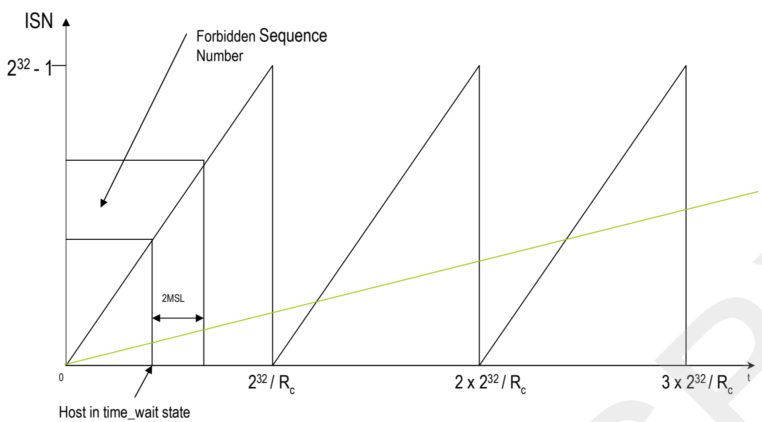
See above graph.

(c) Suppose that the transmitter sends bytes at an average rate $R > R_c$. Use the plot from part (b) to show what goes wrong.



If the transmitter sends data at an average rate $R > R_c$, the ISN will lag behind the transmitter's sequence number. In particular, if the source uses all the sequence numbers in less than 2 MSL, then all the sequence numbers would be forbidden as ISNs for the next connection.

(d) Now suppose that the connection is long-lived and that bytes are transmitted at a rate R that is much lower than R_c. Use the plot from part (b) to show what goes wrong. What can the transmitter do when it sees that this problem is about to happen?

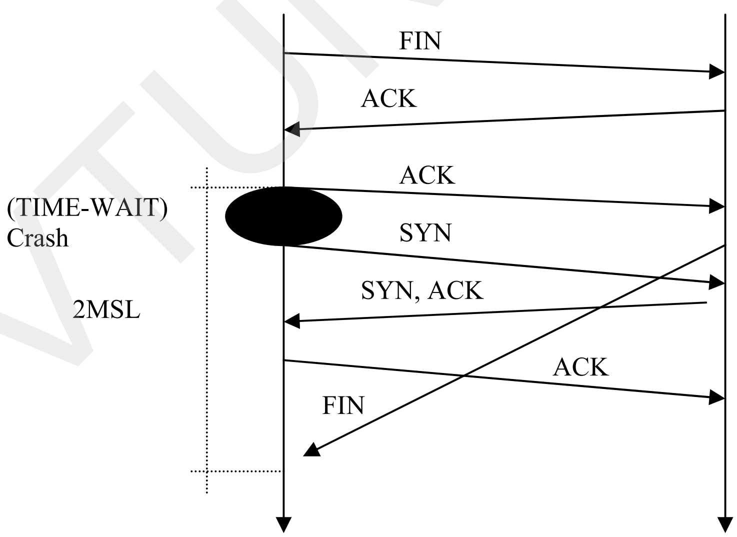


When the connection is long-lived and $R < R_c$, the ISN sequence number will wrap around at a much faster rate than the transmitter sequence number. When a new connection is established, it is possible for the ISN to be selected within the range of sequence numbers used by the slow connection.

8.36. Suppose that during the TCP connection closing procedure, a machine that is in the TIME_WAIT state crashes, reboots within MSL seconds, and immediately attempts to reestablish the connection using the same port numbers. Give an example that shows that delayed segments from the previous connections can cause problems. For this reason RFC 793 requires that for MSL seconds after rebooting TCP is not allowed to establish new connections.

Solution:

A delayed FIN from the earlier connection causes the new connection to be closed prematurely.



8.37. Are there any problems if the server in a TCP connection initiates an active close?

Solution:

As TCP is defined, no problems should arise if the server initiates an active close. Recall from Figure 8.36 that the side that does the active close (by issuing the first FIN segment) will enter the TIME_WAIT state. The side that does the passive close does not. Therefore, when the server does the active close, it will go into the TIME_WAIT state and hence will not be able to be restarted with its same (well-known) port number, because this is part of the parameters that are set aside during the 2MSL wait.

8.38. Use a network analyzer to capture the sequence of packets in a TCP connection. Analyze the contents of the segments that open and close the TCP connection. Estimate the rate at which information is transferred by examining the frame times and the TCP sequence numbers. Do the advertised windows change during the course of the connection?

Solution:

The following sequence of packet captures was obtained by connecting to www.yahoo.com using telnet. The TCP open and close can be observed at the beginning and end of the packet sequence. Advertised window sizes, acknowledgments, and sequence numbers are also shown. (The data was obtained from Ethereal using the print-to-file option.)

```
No. Time      Source          Destination     Protocol Info
 1 0.000000   Intel_c1:8e:0a  Broadcast       ARP    Who has 192.168.2.3?  Tell 192.168.2.2
 2 12.992506  192.168.2.18    192.168.2.1     DNS    Standard query A www.yahoo.com
 3 13.001008  192.168.2.1     192.168.2.18    DNS    Standard query response CNAME www.yahoo.akadns.net A 216.109.125.79 A
216.109.125.72 A 216.109.125.69 A 216.109.117.205 A 216.109.125.78 A 216.109.125.71 A 216.109.125.64 A 216.109.118.64
 4 13.001678  192.168.2.18    216.109.125.79  TCP    2498 > http [SYN] Seq=147142992 Ack=0 Win=8192 Len=0
 5 13.039151  216.109.125.79  192.168.2.18    TCP    http > 2498 [SYN, ACK] Seq=2183346772 Ack=147142993 Win=65535 Len=0
 6 13.039221  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142993 Ack=2183346773 Win=8760 Len=0
 7 18.472270  192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142993 Ack=2183346773 Win=8760 Len=1
 8 18.600842  00000000.0001031dccf7 00000000.Broadcast    IPX SAP  General Query
 9 18.622879  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142994 Win=65535 Len=0
10 18.734094  192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142994 Ack=2183346773 Win=8760 Len=1
11 18.905241  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142995 Win=65535 Len=0
12 18.924462  192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142995 Ack=2183346773 Win=8760 Len=1
13 19.078556  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142996 Win=65535 Len=0
14 19.244070  192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142996 Ack=2183346773 Win=8760 Len=1
15 19.369356  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142997 Win=65535 Len=0
16 19.692438  192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142997 Ack=2183346773 Win=8760 Len=2
17 19.755278  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142999 Win=65535 Len=1460
18 19.756467  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183348233 Ack=147142999 Win=65535 Len=1460
19 19.756515  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183349693 Win=8760 Len=0
20 19.758513  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183349693 Ack=147142999 Win=65535 Len=1460
21 19.843349  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183351153 Ack=147142999 Win=65535 Len=1460
22 19.843467  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183352613 Win=8760 Len=0
23 19.845612  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183352613 Ack=147142999 Win=65535 Len=1460
24 19.846719  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183354073 Ack=147142999 Win=65535 Len=1460
25 19.846764  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183355533 Win=8760 Len=0
26 19.899221  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183355533 Ack=147142999 Win=65535 Len=1460
27 19.900426  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183356993 Ack=147142999 Win=65535 Len=1460
28 19.900508  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183358453 Win=8760 Len=0
29 19.902280  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183358453 Ack=147142999 Win=65535 Len=1460
30 19.911624  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183359913 Ack=147142999 Win=65535 Len=1460
31 19.911745  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183361373 Win=8760 Len=0
32 19.916041  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183361373 Ack=147142999 Win=65535 Len=1460
33 19.917275  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183362833 Ack=147142999 Win=65535 Len=1460
34 19.917335  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183364293 Win=8760 Len=0
35 19.942667  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183364293 Ack=147142999 Win=65535 Len=1460
36 19.943834  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183365753 Ack=147142999 Win=65535 Len=1460
37 19.943914  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183367213 Win=8760 Len=0
38 19.959414  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183367213 Ack=147142999 Win=65535 Len=1460
39 19.962971  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183368673 Ack=147142999 Win=65535 Len=1460
40 19.963078  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183370133 Win=8760 Len=0
41 19.964934  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183370133 Ack=147142999 Win=65535 Len=1460
42 19.965690  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183371593 Ack=147142999 Win=65535 Len=1460
43 19.965733  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183373053 Win=8760 Len=0
44 19.987210  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183373053 Ack=147142999 Win=65535 Len=1460
45 19.994098  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183374513 Ack=147142999 Win=65535 Len=1460
46 19.994170  192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183375973 Win=8760 Len=0
47 20.013318  216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183375973 Ack=147142999 Win=65535 Len=1460
```

```
48 20.016425   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183378893 Ack=147142999 Win=65535 Len=1460
49 20.016538   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
50 20.018301   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183380353 Ack=147142999 Win=65535 Len=1460
51 20.018342   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
52 20.034934   216.109.125.79  192.168.2.18  TCP  http > 2498 [FIN, PSH, ACK] Seq=2183381813 Ack=147142999 Win=65535 Len=697
53 20.035032   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
54 21.275248   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183377433 Ack=147142999 Win=65535 Len=1460
55 21.275415   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183382510 Win=8760 Len=0
56 21.346468   216.109.125.79  192.168.2.18    TCP    http > 2498 [FIN, ACK] Seq=2183382510 Ack=147142999 Win=65535 Len=0
57 21.346585   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183382511 Win=8760 Len=0
58 23.239449   192.168.2.18    216.109.125.79  TCP    2498 > http [FIN, ACK] Seq=147142999 Ack=2183382511 Win=8760 Len=0
59 23.273437   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183382511 Ack=147143000 Win=65535 Len=0
```

8.39. Devise an experiment to use a network analyzer to observe the congestion control behavior of TCP. How would you obtain Figure 8.37 empirically? Run the experiment and plot the results.

Solution:

Congestion involves the buildup of packets in a buffer and can be triggered by the sustained arrivals of packets from a high speed network, e.g. a LAN, to a router feeding a slow-speed network, e.g. a dialup modem to an ISP. Thus one way of observing congestion is to send a stream of packets from a home LAN onto the Internet via a dialup modem. Congestion can also occur when multiple users send packets on multiple inputs to the same output port on a router. Thus a second way of generating congestion is to have several machines simultaneously send a stream of packets to the Internet via a home router that connects to a DSL or cable modem. A packet capture tool such as Ethereal can be used to track the evolution of sequence numbers and segment retransmissions over time. However, the congestion window is controlled by TCP which operates in the kernel of the OS. To obtain traces of `cwnd` versus time a tool such as `tcpdump` with debug option, or TCP instrumentation tools (such as provided by the Web100 project for Linux) need to be used.

8.40. A fast typist can do 100 words a minute, and each word has an average of 6 characters. Demonstrate Nagle's algorithm by showing the sequence of TCP segment exchanges between a client, with input from our fast typist, and a server. Indicate how many characters are contained in each segment sent from the client. Consider the following two cases:

Solutions follow questions:

The typist types 100 words per minutes, averaging 6 characters per word. This is equivalent to 600 characters per minute or 10 characters per second. Therefore, the typist can type a character every 100 ms.

(a) The client and server are in the same LAN and the RTT is 20 ms.

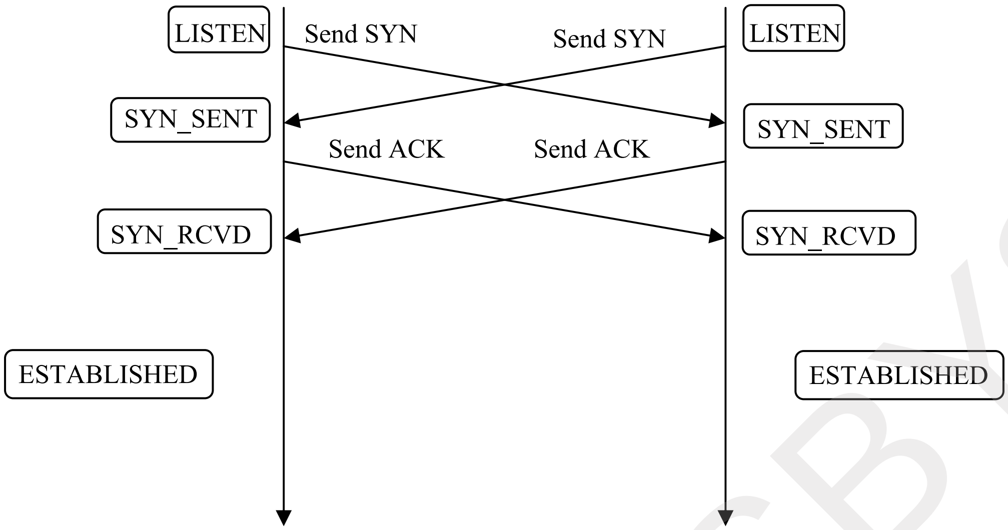
In this case Nagle's Algorithm is not activated since acknowledgments arrive before the next character is typed. Each client segment is 41 bytes long assuming IP and TCP headers are 20 bytes each.

(b) The client and server are connected across a WAN and the RTT is 100 ms.

In this case one or two characters are typed before an acknowledgment is received. Therefore, segments are either 41 or 42 bytes long.

8.41. *Simultaneous Open*. The TCP state transition diagram allows for the case where the two stations issue a SYN segment at nearly the same time. Draw the sequence of segment exchanges and use Figure 8.36 to show the sequence of states that are followed by the two stations in this case.

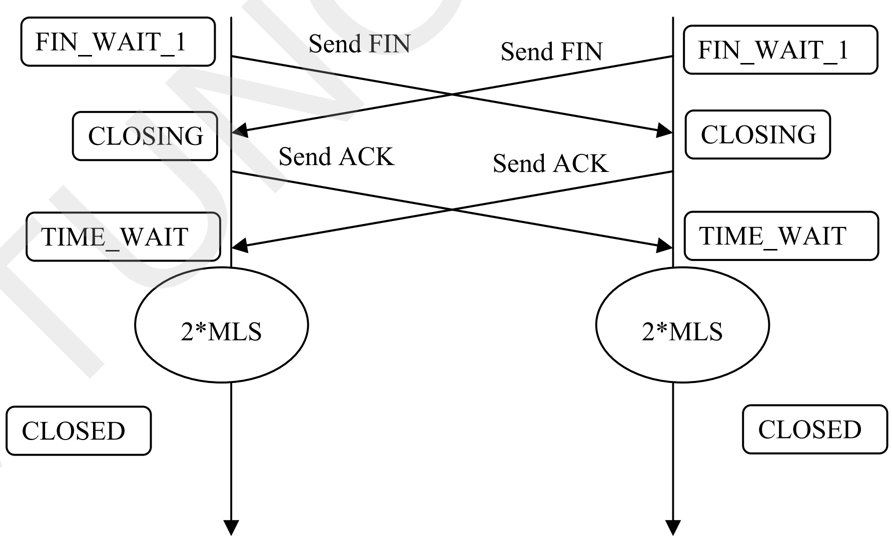
Solution:



8.42. *Simultaneous Close*. The TCP state transition diagram allows for the case where the two stations issue a FIN segment at nearly the same time. Draw the sequence of segment exchanges and use Figure 8.36 to show the sequence of states that are followed by the two stations in this case.

Solution:

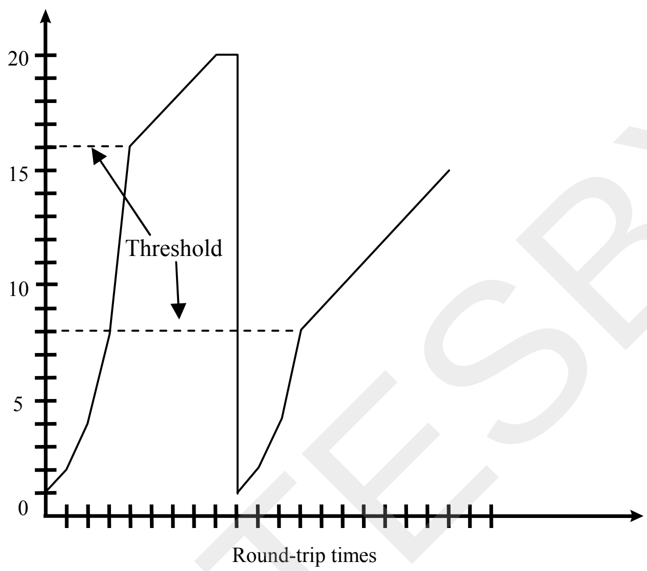
The sequence of the state transition will be the same for two hosts. After FIN sent, it moves from ESTABLISHED to FIN_WAIT_1. Once the host receives the FIN from other host, it sends an ACK and move from FIN_WAIT_1 to CLOSING. Finally, when the host received ACK from each other, it moves from CLOSING to TIME_WAIT. After 2MSL, both hosts transit back to CLOSED state.

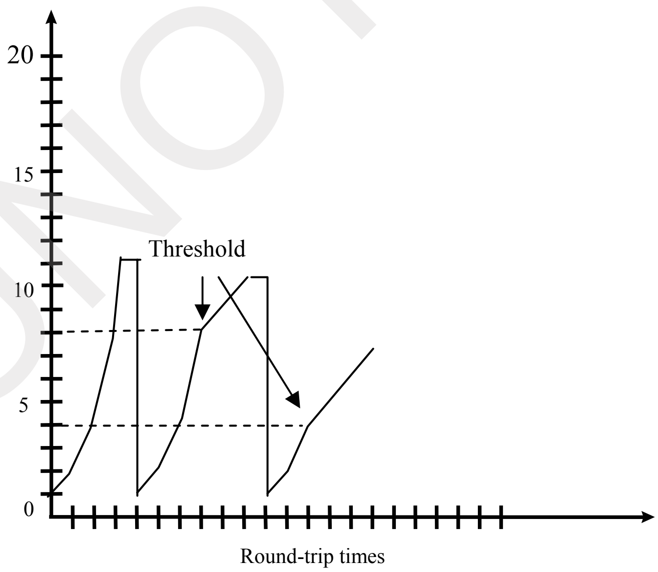


8.43. Suppose that a TCP source (with unlimited amount of information to transmit) begins transmitting onto a link that has 1 Mbps in available bandwidth. Sketch congestion window versus time trajectory. Now suppose that another TCP source (also with unlimited amount of information to transmit) begins transmitting over the same link. Sketch the congestion window versus the time for the initial source.

Solution:

Initially the TCP source has about 1 Mbps of available bandwidth, so its congestion avoidance behavior will begin as the rate approaches 1 Mbps. When the available bandwidth drops to 500 kbps, the congestion window behavior will begin sooner. The following two figures show the corresponding congestion window trajectories.





8.44. What is the maximum width of a RIP network?

Solution:

The maximum width of an RIP network is 15 nodes. Since the maximum number of hops in RIP network is 15, node 16 represents infinity.

8.45. Let's consider the bandwidth consumption of the RIP protocol.

Solutions follow questions:

(a) Estimate the number of messages exchanged per unit time by RIP.

A router implementing RIP sends an update message every 30 seconds. Assume that the typical node has *D* neighbors and that the number of nodes in the network is *N*. The number of message exchanges is then *ND/30* messages per second.

(b) Estimate the size of the messages exchanged as a function of the size of the RIP network.

A RIP message consists of a four-byte header plus 20-byte per entry and up to 25 entries per message. Therefore, each node will send out a message of at most $20(N – 1) + 4$ bytes every 30 seconds.

(c) Estimate the bandwidth consumption of a RIP network.

The bandwidth consumption is at most $[ND [20 (N – 1) + 4]] / 30$ bytes per second.

8.46. RIP runs over UDP, OSPF runs over IP, and BGP runs over TCP. Compare the merits of operating a routing protocol over TCP, UDP, IP.

Solution:

RIP is a protocol in which the routers operate in a highly distributed fashion following a distance vector algorithm. Message exchanges occur only between neighbors and at periodic intervals or triggered by specific events. UDP is suitable for the exchange of individual messages but without delivery guarantees. However the operation of RIP makes allowances for the lack of such guarantees.

OSPF relies on the use of a reliable flooding procedure to distribute link-state information to all the routers. This reliable flooding procedure requires close coordination with the operation of the routers, and hence direct operation over IP instead of over a transport layer protocol is preferred.

BGP peers exchange the entire BGP routing table initially and incremental updates are sent instead of periodic updates to reduce the bandwidth consumption. A small periodic KEEPALIVE message is used to determine that the BGP peers are alive. Reliable delivery of the routing information is required to minimize the bandwidth consumption. For this reason, TCP is chosen to provide the reliable delivery required by BGP.

8.47. Compare RIP and OSPF with respect to convergence time and the number of messages exchanged under several trigger conditions, that is, link failure, node failure, link coming up.

Solution:

Link failure – OSPF has a faster convergence time than RIP. When a link fails, the corresponding OSPF routers send a link-state update to all peer routers. The peer routers then update their databases. This process allows the link failure state information to be propagated to other routers quickly. In contrast, when a link fails in RIP, the corresponding RIP router updates its own distance vector and sends the link update message to its neighbor. The neighboring nodes update their own routing tables, calculate the new distance vector values and send the updated distance vector to their neighbors. The processing and distribution overhead in RIP slows down the convergence time and routing loops may be created while the algorithm is converging. The faster convergence time of OSPF is at the cost of flooding the network with update messages. RIP is based on the exchange of messages between adjacent nodes only.

Node failure – OSPF converges faster than RIP. OSPF sends a HELLO packet every 10 seconds, compared to RIP's update message every 30 seconds. In case of node failure, OSPF can detect failure of a node within the range of 10 seconds. On the other hand, RIP requires a period of 180 seconds (worst case) to detect a node failure. The grace period of 180 seconds in RIP is due to the fact that RIP is running over UDP which cannot be relied upon to deliver messages consistently. Again, OSPF requires more message exchanges (HELLO packet every 10 seconds) than RIP (update message every 30 seconds).

Link coming up – OSPF converges faster than RIP. When a link is coming up, the OSPF routers attached to this link start sending HELLO packets. Next, these router pairs exchange link-state database description packets and send link-state request packets for those LSA headers that are not in their respective link-state databases. After the databases are updated and synchronized, these OSPF routers send the updated database description packet to their own neighbors and execute the routing algorithms to find out the shortest path. In contrast, a RIP router must first perform a distance vector routing calculation. Next, the router sends the update distance vector its own neighbor. The time it takes for RIP to distribute the new link information is slower than OSPF due to the processing overhead imposed on each RIP router.

8.48. Consider the OSPF protocol.

Solutions follow questions:

(a) Explain how OSPF operates in an autonomous system that has not defined areas.

When OSPF operates in an AS that has no defined area, the broadcastpackets (link-state update and HELLO) must flow all over the AS. If the network (AS) is too large, this approach consumes too many network resources and it does not scale well. Also, each OSPF router link-state database and routing table size increase dramatically when the number of nodes within the AS increases.

(b) Explain how the notion of area reduces the amount of routing traffic exchanged.

When area is used in OSPF, the number of routers within an area decreases as compared to the previous case. The broadcast packets only need to flow within an area. Therefore, the network traffic is reduced and utilization of the network resources is increased. Also, each router has a smaller link-state database and routing table.

(c) Is the notion of area related to subnetting? Explain. What happens if all addresses in an area have the same prefix?

The idea behind subnetting is to add another hierarchical level within a particular class of IP address. The idea behind an area is similar in that it involves the use of hierarchy to simplify routing, but there is no direct relationship between area and subnetting. In an area, multiple classes of IP addresses can exist. Within each class of IP addresses, different ways of subnetting can be done. Therefore, subnetting is just another level of hierarchical level to allow a network administrator to better manage a particular class of IP address.

If all addresses in an area have the same prefix, the area border router (ABR) will advertise or exchange a simple summary to indicate all the addresses with this prefix belong to this particular area. The ABR contains the whole network topology and performs appropriate routing within its area upon the transmission between different areas.

8.49. Assume that there are *N* routers in the network and that every router has m neighbors.

Solutions follow questions:

(a) Estimate the amount of memory required to store the information used by the distance-vector routing.

Each node needs the distance to each neighbor and the distance from each neighbor to all destinations which is $m(N-1)$ entries. Assuming E bytes for each entry the amount of memory is $m(N-1)E$.

(b) Estimate the amount of memory required to store the information by the link-state algorithm.

Each node needs the information for all links across the network. Each node is connected to m links and there are N nodes in the network. Therefore the total number of links in the network is $(1/2)Nm$ entries. Assuming E bytes per entry the amount of memory is $(1/2)mNE$.

8.50. Suppose a network uses distance-vector routing. What happens if the router sends a distance vector with all 0s?

Solution:

A distance vector with all zeros means that the node has distance 0 to all other nodes. This will prompt all neighbors to route all their packets through the given router. Eventually all packets in the network will be routed to this router, resulting in what can be characterized as a "black hole."

8.51. Suppose a network uses link-state routing. Explain what happens if:

Solutions follow questions:

(a) The router fails to claim a link that is attached to it.

The link will be eventually omitted from the routing tables and as a result will not be utilized. This will result in some routing decisions that are not the shortest (optimal) paths. Loss of connectivity is also possible. Moreover, a routing loop may occur due to confusion among routers as to which links they are attached to.

(b) The router claims to have a link that does not exist.

If a router claims to have a link that does not exist, it reports false topology information to other routers within the network. If this non-existing link belongs to a shortest path, all the packets that are sent via this path will be lost. This will affect the network performance severely.

8.52. Consider a broadcast network that has *n* OSPF routers.

Solutions follow questions:

(a) Estimate the number of database exchanges required to synchronize routing databases.

In the worst case, for an n-router OSPF network, n^2 database exchanges are required to synchronize all of the routers databases.

(b) What is the number of database exchanges after a designated router is introduced into the network?

If a designated router is introduced, the number of database exchanges will reduce to *n*. This is because each router must only communicate with the designated router to obtain link-state information.

(c) Why is the backup designated router introduced? What is the resulting number of database exchanges?

If the primary designated router fails, the information exchange will function properly when the backup designated router is introduced. This mechanism is designed to protect against failure and to provide fast recovery. The number of database exchanges is 2*n* since every router must update both the primary and backup designated router with link-state information.

8.53. Suppose *n* OSPF routers are connected to a non-broadcast multi-access network, for example, ATM.

Solutions follow questions:

(a) How many virtual circuits are required to provide the required full connectivity?

For a full connectivity, ATM requires an $n(n - 1)$ virtual circuit connections.

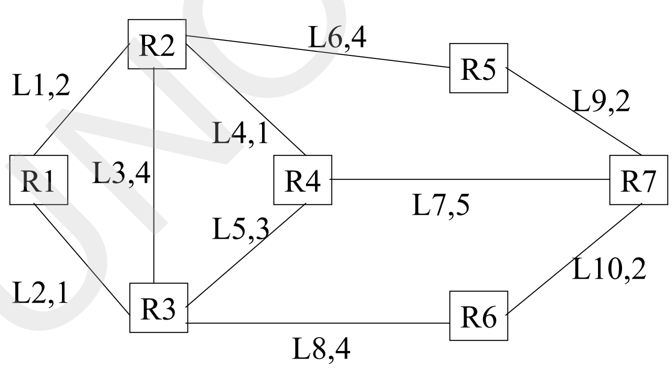
(b) Does OSPF function correctly if a virtual circuit fails?

Yes, OSPF will function correctly if one of the VC fails. This is because the failed VC becomes a failed link in the IP layer and the packet can re-route via a different link (VC) to reach the destination.

(c) Is the number of required virtual circuits reduced if point-to-multipoint virtual circuits are available?

If point-to-multipoint VCs are used, the number of VCs required will be *n*.

8.54. The figure below shows seven routers connected with links that have the indicated costs. Use the Hello protocol to show how the routers develop the same topology database for the network.



Solution:

In the first stage, every router sends a HELLO packet to all the links that it is attached to.

- R1 sends HELLO packets to R2 and R3 via L1 and L2 respectively.
- R2 sends HELLO packets to R1, R3, R4 and R5 via L1, L3, L4 and L6 respectively.
- R3 sends HELLO packets to R1, R2, R4 and R6 via L2, L3, L5 and L8 respectively.
- R4 sends HELLO packets to R2, R3 and R7 via L4, L5 and L7 respectively.

- R5 sends HELLO packets to R2 and R7 via L6 and L9 respectively.
- R6 sends HELLO packets to R3 and R7 via L8 and L10 respectively.
- R7 sends HELLO packets to R4, R5 and R6 via L7, L9 and L10 respectively.

In the next stage, when a router receives a HELLO packet, it replies with a HELLO packet containing the router ID of each neighbor it has currently seen. For example, when R1 receives HELLO from R2, it will send a HELLO packet to R2. The HELLO packet will inform R2 that the neighbors R1 currently sees are R2 and R3, given that R3's HELLO packet has been received by R1 before R2's HELLO packet.

Once all the routers know their neighbor routers, routers exchange Database Description packets to check if their link-state databases are in agreement. If the databases are not in agreement, the routers exchange link-state request and update packets to synchronize their databases.

8.55. Consider the exchange of Hello messages in OSPF.

Solutions follow questions:

(a) Estimate the number of Hello messages exchanged per unit time.

A HELLO packet is sent periodically every 10 seconds. Assume the number of links within a network is L. The total number of HELLO packet exchanged per 10-second unit time is $2L$.

(b) Estimate the size of the Hello messages.

The size of the HELLO packet is 20 bytes for the header plus an entry for each neighbor the router sees. Therefore the size depends on the number of neighbors a router is connected to. Assume the average degree for a router is m. Each router ID is 4 bytes. The size of a HELLO packet = 20 + $4m$ bytes.

(c) Estimate the bandwidth consumed by Hello messages.

The bandwidth consumed by the HELLO packets is $2L$ [(20 x $4m$) x 8] / 10 bps.

8.56. Consider the notion of adjacency in OSPF.

Solution:

(a) Explain why it is essential that all adjacent routers be synchronized.

Adjacent routers must be synchronized to ensure that routers use the same topology when running the routing algorithm, and ultimately to avoid loops and unnecessary packet dropping in the network.

(b) Explain why it is sufficient that all adjacent routers be synchronized, that is, it is not necessary that all pairs of routers be synchronized.

For every adjacent router pair, the routers that are connected to this particular adjacent router pair must also be synchronized (according to the rule of adjacency). As a result, all the routers within the network will eventually synchronize.

8.57. Consider the robustness of OSPF.

Solutions follow questions:

(a) Explain how the LSA checksum provides robustness in the OSPF protocol.

The LSA checksum provides error detection for the entire content of the LSA except the link-state age. It gives a second level of error detection to ensure each individual LSA entry carries correct information. If an error is detected in an LSA entry, the router can discard the specific entry while continuing to use other entries in the LSA update packet.

(b) An OSPF router increments the LS Age each time it inserts the LSA into a link-state update packet. Explain how this protects against an LSA that is caught in a loop.

The router increments the LS age each time it inserts the LSA into a link-state update packet. Therefore, whenever a router receives an LSA that is caught in a loop, the router can verify the validity of the LSA with its current LS age. If the LSA is too old, the router will ignore the LSA packet.

(c) OSPF defines a minimum LS update interval of 5 seconds. Explain why.

OSPF uses flooding to distribute the LSA packets. If the LS update interval is too short, the router may not be able to distinguish the order of the LSA packets it receives. A too-short update interval may consume a lot of processing power just to handle the link-state database and routing algorithm. Therefore, choosing the interval of 5 seconds will allow router to exchange LSA twice with every HELLO packet interval. This would allow the network to respond faster in case of any link-state change while keeping the processing overhead low.

8.58. Assume that for OSPF updates occur every 30 minutes, an update packet can carry three LSAs, and each LSA is 36 bytes long. Estimate the bandwidth used in advertising one LSA.

Solution:

An OSPF update occurs every 30 minutes = 1800 seconds. An update can carry three LSAs. Each LSA is 36 bytes long. The OSPF common header is 24 bytes. The size of the LSA field in link-state update consumes 4 bytes. Therefore, if OSPF updates occur every 1800 seconds containing 3 LSAs, the bandwidth for advertising one LSA is equal to $([(24 + 4 + 3 \times 36) \times 8] / 1800) \times (1/3) = 0.201$ bits/second.

8.59. Identify elements where OSPF and BGP are similar and elements where they differ. Explain the reasons for similarity and difference.

Solution:

Similarities - OSPF and BGP are used to exchange routing information including active routes, inactive routes and error conditions within the network in general. OSPF handles the information exchange within an AS while the BGP handles the information exchange between different ASs. Both OSPF and BGP allow a router to construct the network topology based on the link-state and path vector information respectively. OSPF and BGP use HELLO and KEEPALIVE messages respectively to determine the presence of peers.

Differences - OSPF runs over IP while BGP runs over a TCP connection. OSPF is a link-state protocol and BGP uses a path-vector protocol. BGP can enforce policy by affecting the selection of different paths to a destination and by controlling the redistribution of routing information. BGP only requires an incremental update of the database. However, OSPF requires a periodic update (refresh every 30 minutes).

8.60. Discuss the OSPF alternate routing capability for the following cases:

Solutions follow questions:

(a) Traffic engineering, that is, the control of traffic flows in the network.

In OSPF, the link-state metric exchange between routers can represent the volume of flow of traffic for each individual link. Therefore, each router can use these link-state metrics (traffic flow) to determine the best route for a particular IP packet given its type-of-service (TOS) field. However, the processing required in each router increases dramatically. One possible solution is to make use of explicit routing by pre-establishing a path that is suitable for each class of service.

(b) QoS routing, that is, the identification of paths that meet certain QoS requirements.

OSPF can use a QoS parameter or metric for each link to determine the optimal path for QoS routing. All routers within the network can use a set of routing algorithms to send packets along a particular route that satisfies a certain QoS requirement, e.g. delay, bandwidth, or bit-error rate requirement.

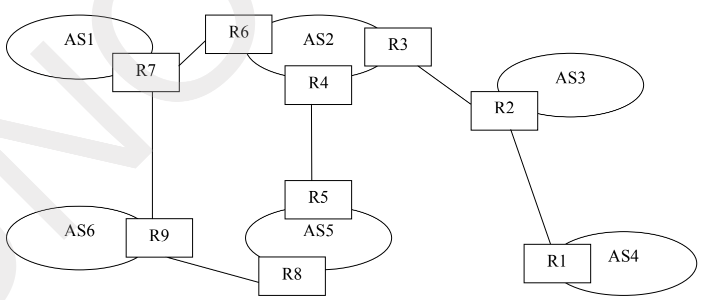
(c) Cost-sensitive routing, that is, the identification of paths that meet certain price constraints.

In cost-sensitive routing, the link-state metric can represent the cost of each link. The router determines the best route that meets a certain price constraint for a particular packet. An ingress router needs to have the information to determine the minimum cost paths to all destinations.

(d) Differential security routing, the identification of paths that provide different levels of security.

In differential security routing, the link-state metric involves identification of the security level of each link. All routers need to come up with a standard definition of a security metric for links. The router identifies the best route to satisfy each individual packet's security requirement.

8.61. Consider the autonomous systems and BGP routers in the following figure.



Solutions follow questions:

(a) Suppose that a certain network prefix belongs to AS4. Over which router pairs will the route to the given network be advertised?

If a route in AS4 requires advertising, R1 will run an eBGP protocol and establish a TCP connection to R2 to exchange the routing information. R2 will relay the message to R3 through a TCP connection based on eBGP. R3 forwards this information to R4 and R6 via iBGP. R6 advertises the information to R7 and R4 advertises to R5 via eBGP. R7advertises the information to R9 via eBGP. Finally, R5 sends the information to R8 via iBGP and R9 sends the information also to R8 via eBGP.

(b) Now suppose the link between R1 and R2 fails. Explain how a loop among AS1, AS2, AS6, and AS5 is avoided.

With the use of path-vector routing, a loop occurs whenever a BGP router receives an update message with an AS path attribute that contains its own AS number. The BGP router ignores the route and discards the information immediately.

(c) Suppose that R9 is configured to prefer AS1 as transit and R6 is configured to prefer AS1 as transit. Explain how BGP handles this situation.

The use of path-vector routing will again allow routers R9 and R6 to avoid the potential loop that can result from these preferences.

8.62. Why does BGP not exchange routing information periodically like RIP?

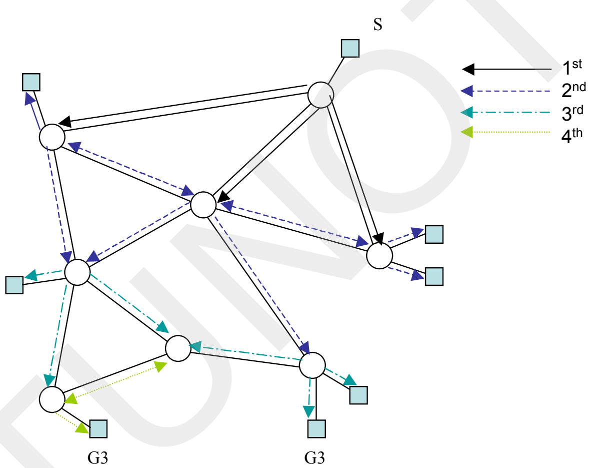
Solution:

BGP runs over TCP, which provides reliable service and simplifies BGP significantly by allowing the protocol to assume the availability of reliable information. In contrast, RIP runs over UDP, which is an unreliable protocol that may experience packet loss. RIP therefore requires periodic exchange of routing information to ensure routing information is correct and up-to-date.

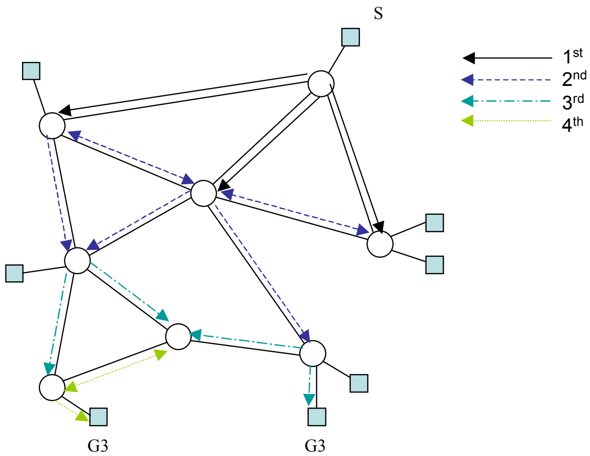
8.63. Consider the network shown in Figure 8.60. Suppose that a source connected to router 7 wishes to send information to multicast group G3.

Solutions follow questions:

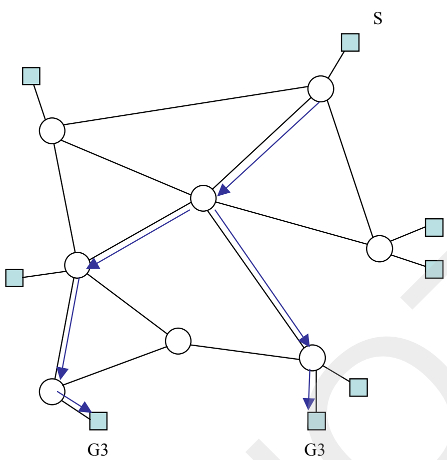
(a) Find the set of paths that are obtained from reverse-path broadcasting.



(b) Repeat for truncated reverse-path broadcasting.



(c) Repeat for reverse-path multicasting.



8.64. Discuss the operation of the reverse-path multicasting in the following two cases:

Solutions follow questions:

(a) The membership in the multicast group in the network is dense.

If the membership in the multicast group in the network is dense, then there will not be extensive use of prune messages and operation will be similar to reverse-path broadcasting. The utilization of bandwidth for multicasting packets will be relatively efficient.

(b) The membership in the multicast group in the network is sparse.

If the membership in the multicast group in the network is sparse, there will be extensive pruning of paths. The multicast tree will be "thin" relative to the overall network, and the utilization of bandwidth for multicasting packets will be inefficient.

8.65. Suppose an ISP has 1000 customers and that at any time during the busiest hour of the day, the probability that a particular user requires service is .20. The ISP uses DHCP. Is a class C address enough so that the probability is less than 1 percent that there is no IP address available when a customer places a request?

Solution:

A class C address space provides 254 IP addresses. The distribution of the number of active customers is given by a binomial distribution with parameters n = 1000 customers and p =.20. Let X be the number of active customers. The probability that $X = k$ customers are active at a given time is given by:

$$P[X = k] = \binom{1000}{k}(0.2)^k (0.8)^{1000-k}$$

The random variable X has mean equal to $m = np$ =1000*0.2 = 200 and variance given by $\sigma^2 = np(1-p)$ = 1000(.2)(.8) = 160.

For large values of n, the binomial distribution can be approximated by a Gaussian distribution with the same mean and variance. In particular we have that:

$$P[X > k] \approx P[X_{Gaus\sin} > k] = \frac{1}{\sqrt{2\pi\sigma^2}} \int_k^{\infty} e^{-(x-m)^2/2\sigma^2} dx = \frac{1}{\sqrt{2\pi}} \int_{\frac{k-m}{\sigma}}^{\infty} e^{-x^2/2} dx$$

The value of the above integral at the point $(k - m)/\sigma$ = (254-200)/40 = 1.35 is 0.088=8.8%. This is the probability that the ISP does not have enough IP addresses to serve customer connection requests.

8.66. Compare mobile IP with the procedures used by cellular telephone networks (Chapter 4) to handle roaming users.

Solutions follow questions:

(a) Which cellular network components provide the functions of the home and foreign agent?

The home location register in cellular network provides the equivalent functions of the home agent in mobile IP network. The visitor location register in cellular network provides the functions of the foreign agent in mobile IP network.

(b) Is the handling of mobility affected by whether the transfer service is connectionless or connection-oriented?

The basic handling of the mobile users is the same in terms of the operation of the home location and visitor location registers. The manner in which information is transferred is of course different given that one is connectionless and the other connection-oriented.

8.67. Consider a user that can be in several places (home networks) at different times. Suppose that the home networks of a user contain registration servers where users send updates of their location at a given time.

Solutions follows questions:

(a) Explain how a client process in a given end system can find out the location of a given user in order to establish a connection, for example, Internet telephone, at a given point in time.

A client process can locate a given user by sending a request to a designated registration server. This registration server contains the most up-to-date location of a given user. Once the client process finds the location, it can establish a connection based on the information. The way to locate the registration server can be predefined by the network administrator or based on the request IP address prefix. In the latter approach, the client can use the address prefix of the given user to locate the corresponding registration server.

(b) Suppose that proxy servers are available, whose function is to redirect location requests to another server that has more precise location information about the callee. For example, a university might have such a server, which redirects requests for prof@university.edu to departmental servers. Explain how a location request for engineer@home.com might be redirected to a.prof@ece.university.edu.

When a location request for engineer@home.com is sent to the proxy server, the server will forward the request to the home.com server. The home.com server can send a redirect message informing the proxy server that the engineer@home.com can be redirected to a.prof@ece.university.edu. Therefore, whenever the client sends messages to the engineer@home.com via the proxy server, the server will automatically redirect the message to a.prof@ece.university.edu.