Problem Set #1 Solutions

Note: 1KB = 1024 bytes, 1MB = 1024 KB, 1GB = 1024 MB

The actual final solution can vary a bit because some students prefer to use the 1000 vs 1024 notation. They should get credit as long as they use the right formula and do not make significant calculation mistakes. For example, substituting 50KB for 5KB or if there is no logical path to the answer. These solutions have used 1 Mbps = 10^6 bits/ sec, 1 Gbps = 10^9 bit/sec notations for calculations. So again, for notations in Problem 2, the answers could vary slightly.

Both of following conversions are considered right:

1. 1 Mbps = 10^6 bits/ sec, 1 Gbps = 10^9 bit/sec,
2. 1 Mbps = 1024 x 1024 x 8 bits/ sec, 1 Gbps = 1024 x 1024 x 1024 x 8 bit/sec

Problem 1 –

Students should show the related formulas or some work as to how they arrived at their answers

-1pt if answer is wrong but formula is correct

if no formula/any work then -1pt for 2pt question, -2pt for 3pt question

Suppose a 2-Gbps point-to-point link is being set up between the Earth and a new lunar colony. The distance from the moon to the Earth is approximately 385,000 km, and data travels over the link at the speed of light—3×10^8 m/s.

(a) Calculate the minimum RTT for the link. (2 pt)

The minimum

RTT = (2 x distance)/ (speed of light)

= (2 x 385,000,000 m)/ (3 x 10^8 m/sec) = 2.57 sec.

(b) Using the RTT as the delay, calculate the delay × bandwidth product for the link. (2 pt)

The delay x bandwidth product is

= delay × bandwidth

= 2.57 sec x 2Gbps

= 5140 x 10 ^{6} bits = 5140 x 10 ^{6} / (1024 x 1024 x 8) = 612.735 MB

(c) What is the significance of the delay × bandwidth product computed in (b)? (3 pts)

Open ended question, to be given full credit if some of these keywords match student's answers. Use your judgement.

BDP - This represents the amount of data the sender can send before it would be possible to receive a response. In data communications, bandwidth-delay product refers to the product of a data link's capacity (in bits per second) and its round-trip delay time (in seconds). The result, an amount of data measured in bits (or bytes), is equivalent to the maximum amount of data on the network circuit at any given time, i.e., data that has been transmitted but not yet acknowledged.

A high bandwidth-delay product is an important problem case in the design of protocols such as Transmission Control Protocol (TCP) in respect of TCP tuning, because the protocol can only achieve optimum throughput if a sender sends a sufficiently large quantity of data before being required to stop and wait until a confirming message is received from the receiver, acknowledging successful receipt of that data. If the quantity of data sent is insufficient compared with the bandwidth-delay product, then the link is not being kept busy and the protocol is operating below peak efficiency for the link. Protocols that hope to succeed in this respect need carefully designed self-monitoring, self-tuning algorithms.

(d) A camera on the lunar base takes pictures of the Earth and saves them in digital format to disk. Suppose Mission Control on Earth wishes to download the most current image, which is 30 MB. What is the minimum amount of time that will lapse between when the request for the data goes out and the transfer is finished? (3 pts)

Students should show the related formulas or some work as to how they arrived at their answers

-1pt if answer is wrong but formula is correct

if no formula/any related work then -1pt for 2pt question, -2pt for 3pt question

We require at least one RTT before the picture could begin arriving at the ground (TCP would take two RTTs). Assuming bandwidth delay only, it would then take

= file size/Bandwidth

= 30MB/2 x 10^{9}bps

= (30 x 1024 x 1024 x 8)/ (2 x 10^{9}) bps

= 0.125 sec to finish sending, for a total time of 2.57 + 0.125 = 2.695 sec until the last picture bit arrives on earth.

Problem 2 – (4 pt)

What is the throughput when retrieving a 4MB file across a 1Gbps network with a round trip time of 150msec, again ignoring ACKs?

(a) Assuming that the file transfer has to be initiated by a request. (2 pt)
(b) Otherwise. (2 pt)

Students should show the related formulas or some work as to how they arrived at their answers
-1pt if answer is wrong but formula is correct

if no formula/any work then -1pt for 2pt question, -2pt for 3pt question

Problem 2 – Soln

(a) Assuming that the file transfer has to be initiated by a request:

Transmit time = 4MB/1Gbps
= 4 x 1024 x 1024 x 8 / 1 x 10^{9}
= 33.55 msec

Throughput (actual data transferred)
= file size / (RTT + transmit time)
// 1 RTT delay in the case of request initiated by sender
= 4MB / (150msec + transmit time)

= 4MB / (150msec + 33.55msec)

= 4 x 8 x 1024 x 1024 bit / 0.183 msec

= 183.35 x 10 ^{6} bps

(b) Otherwise:

// 0.5 RTT delay in the case of request not initiated by sender

Throughput

= 4MB / (75msec + transmit time)

= 4MB / (75msec + (4MB /1Gbps))

= 4 x 8 x 1024 x 1024 bit / (75msec + 33.55msec)

= 4 x 8 x 1024 x 1024 bit /0.1085 msec

= 0.309 x 10^9 bps

Problem 3 – (10 pts)

Calculate the total time required to transfer a 2000 KB file in the following cases, assuming an RTT of 100 msec, a packet size of 1 KB data, and an initial 2×RTT of "handshaking" before data is sent:

Students should show the related formulas or some work as to how they arrived at their answers

-1pt if answer is wrong but formula is correct

if no formula/any work then -1pt for 2pt question, -2pt for 3pt question

(a) The bandwidth is 1 Mbps, and data packets can be sent continuously. (2 pt)

a) The total transfer time is the sum of
   - the time required for the handshaking: 2 x RT T
   - the propagation delay: 1/2 x RT T
   - the transmission time: 2000 x packet size / bandwidth
   - total transfer time = 0.2s + 0.05s +(2000 x 1024 x 8) bit /1 x10^6 bit/s) ≈ 16.63 sec

(b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next. (2 pt)

The total transfer time is the sum of

- the time required for the handshaking: 2 x RTT
- the transfer time for 1999 packets: 1999 RTT
- the transfer time for the last packet: 1 / 2 RTT

total transfer time

= 0.2 sec + (1999.5 x 0.1 sec) + 2000 x (8 x 1024 bit)/ (1.5 x 10^6 bit/s)

≈ 210.67 sec

(c) The bandwidth is “infinite,” meaning that we take transmit time to be zero, and up to 40 packets can be sent per RTT. (2 pt)

The total transfer time is the sum of

- the time required for the handshaking: 2 x RTT
- the propagation delay and the time we have to wait due to the protocol: 49 x RTT
- the transmission time for the 2000 packets: 2000 x (packet size / bandwidth) = 0 (BW is infinite)

total transfer time = 0.2s + 4.95s + 0s = 5.15sec

(d) The bandwidth is infinite, and during the first RTT we can send one packet (2^{1-1}), during the second RTT we can send two packets (2^{2-1}), during the third we can send four (2^{3-1}), and so on. (4 pt)

$2^{n+1} – 1$ = 2000 packets

If students have directly calculated the entire summation without the formula,
$2^1 + 2^2 + 2^3 + \ldots$, It is also correct.
-1pt if answer is wrong but formula/working is correct
if no formula/any work then -3pt

Thus, we need 9.96 = approx. 10 RTTs to transmit the 2000 packets.
The total transfer time is the sum of

- the time required for the handshaking: 2xRT T
- the propagation delay and the time we have to wait due to the protocol: 10 x RT T
- the transmission time for the 2000 packets: 2000 x (packet size/ bandwidth) = 0

total transfer time = 0.2 sec + 10 x 0.1 sec + 0.5 sec = 1.25 sec

Problem 4 – 2 pt

Determine the width of a bit on a 20 Gbps link. Assume a copper wire, where the speed of propagation is $2.3 * 10^8$ m/s.

Students should show the related formulas or some work as to how they arrived at their answers
-1pt if answer is wrong but formula is correct

Width of bit = speed / bandwidth

2.3×10^8 m/s / (20×10^9 bit/s)

= 0.0115 m/bit

A single bit has a width of 1.15 cm on the wire.

Problem 5 – 4 pt

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.

Students should show the related formulas or some work as to how they arrived at their answers

-1pt if answer is wrong but formula is correct.

if no formula/any work then -3pt

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)

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

Take the derivative with respect to p,

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

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

P = ½

Problem 6 – 6 pt

Suppose there is a 10 Mbps microwave link between a geostationary satellite and its base station on Earth. Every minute the satellite takes a digital photo and sends it to the base station. Assume a propagation speed of 2.4×10^8 meters/sec. geostationary satellite is 36,000 kilometers away from earth surface

-1pt if answer is wrong but formula is correct.

a. What is the propagation delay of the link? (2 pt)

Propagation delay
= distance / speed
= 36,000 km / 2.4 x 10^8
= 150 msec

b. What is the bandwidth-delay product, R x (propagation delay)? (2 pt)

= 150 msec x 10 x 10^6

= 1,500,000 bits

c. Let *x* denote the size of the photo. What is the minimum value of *x* for the microwave link to be continuously transmitting? (2 pt)

Every minute a picture is sent so time = 60 secs

Propagation delay

= time / bandwidth

= 60 secs x 10 x 10^9

= 600,000,000 bits

Problem 7 – 9 pts

Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1,024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions:

Open ended question, to be given full credit if some of these keywords match student's answers. Use your judgement.

(a) What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer. (3 pts)

Here we have a window size of N = 4. Suppose the receiver has received packet k − 1, and has ACKed this packet and all other preceding packets. If all of these ACK's have been received by the sender, then the sender's window is $[k, k + N - 1]$. Suppose next that none of the ACKs have been received by the sender. In this second case, the sender's window contains k

-1 and the N packets up to and including $k-\mathrm{N}$. The sender's window is thus $[k-N, k-1]$. By these arguments, the senders window is of size 3 and begins somewhere in the range $[k-N, k]$.

(b) What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time t? Justify your answer. (3 pts)

If the receiver is waiting for packet k, then it has received (and ACKed) packet k-1 and the N-1 packets before that. If none of those N ACKs have been yet received by the sender, then ACK messages with values of [k-N,k-1] may still be propagating back. Because the sender has sent packets [k-N, k-1], it must be the case that the sender has already received an ACK for k-N-1. Once the receiver has sent an ACK for k-N-1 it will never send an ACK that is less than k-N-1. Thus, the range of in-flight ACK values can be in the range [k-N-1, k-1].

(c) With the Go-Back-N protocol, is it possible for the sender to receive an ACK for a packet that falls outside of its current window? Justify your answer with an example. (3 pts)

Yes, it is possible. Consider the following example scenario. Suppose the sender has a window size of 3 and sends packets 1, 2 and 3 at time t_0. At t_1, the receiver ACKs packets 1, 2 and 3. At t_2 , the sender times out and resends 1, 2 and 3. At t_3, the receiver receives the duplicates and re-acknowledges 1, 2 and 3. At t_4 , the sender receives the ACKs that the receiver sent at and advances its window to packets 4, 5 and 6. Finally, at t_5, the sender receives the ACKs that the receiver sent at t_2 . These ACKs are outside its window.

For all timing diagrams, -6 pts if none of the solutions match even remotely.

Problem 8 – 8 pt

Draw a time line diagram for the sliding window algorithm with SWS = RWS = 3 frames, for the following situations. Use a timeout interval of about 2 × RTT. And assume 2 frames must be send ½ RTT apart which means if everything is normal Sender will receive ACK and then immediately send the next frame.

Frames 3 and 6 are lost on their first transmissions. Draw the algorithm with time line diagram till Frame 6 is sent. (8pt)

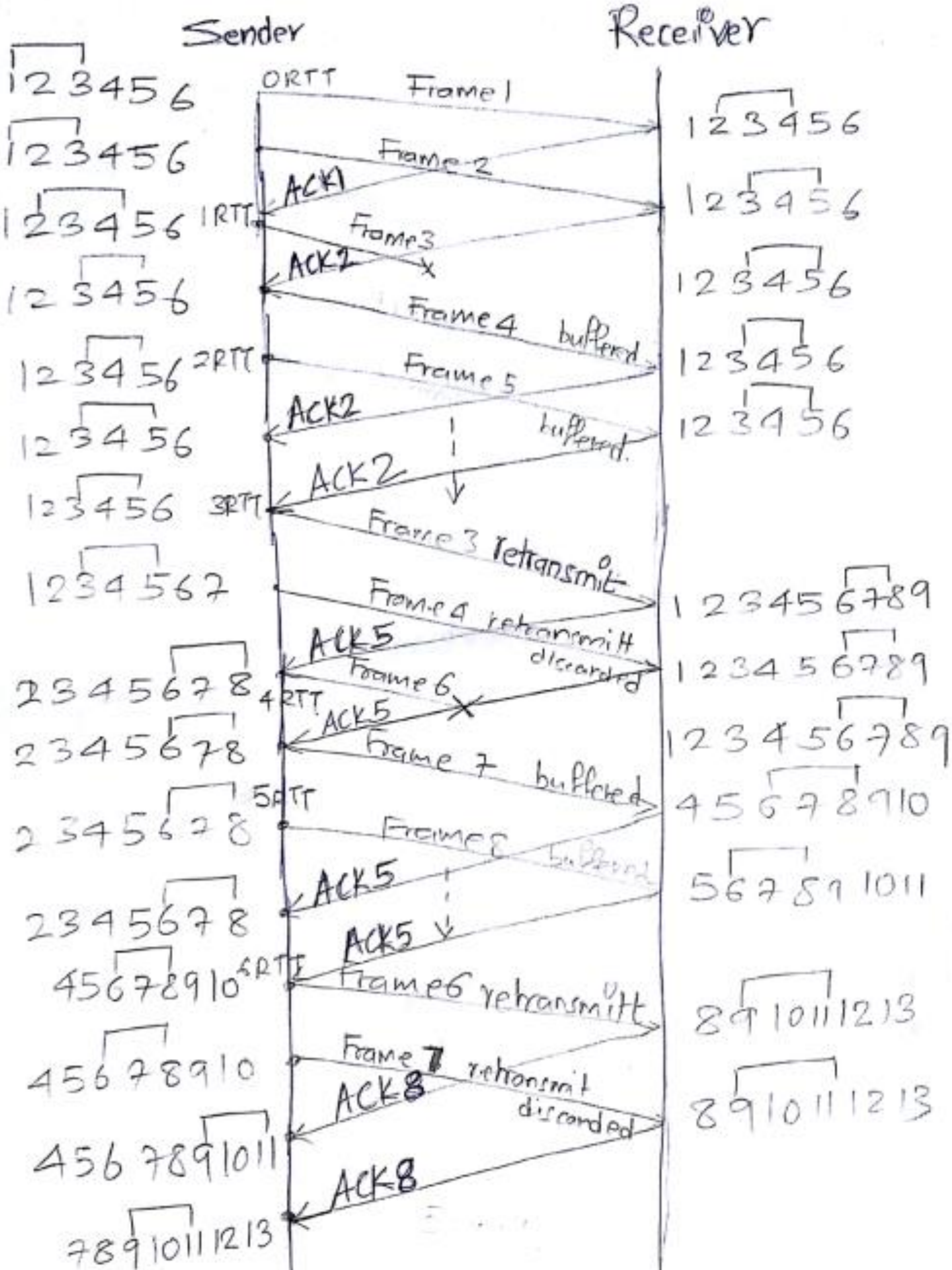
Solution:

Check if all the ACKS at the given time intervals in the solution have been denoted by the student

- 1pt if only some ACKS have been denoted.
- 5pt if no ACKS, no sender and receiver windows, no frame numbers have been denoted
- Frames which are retransmitted can be discarded or buffered if receiver's window size. Both such solutions are right. Frames which are retransmitted if out of receiver's window size **should** be discarded.

For example, if current receiver window size (denoted by those square brackets over numbers on receiver side) is 345, and frame 4 is retransmitted, this frame can be buffered or discarded, both solutions are correct till it is a frame within receiver's window size i.e between 345. For example in this solution frame 7 is retransmitted but it is out of window size as current receiver window is pointing to 8,9,10 so it would be discarded. Same applied to solution **for problem 9.**

Problem 8



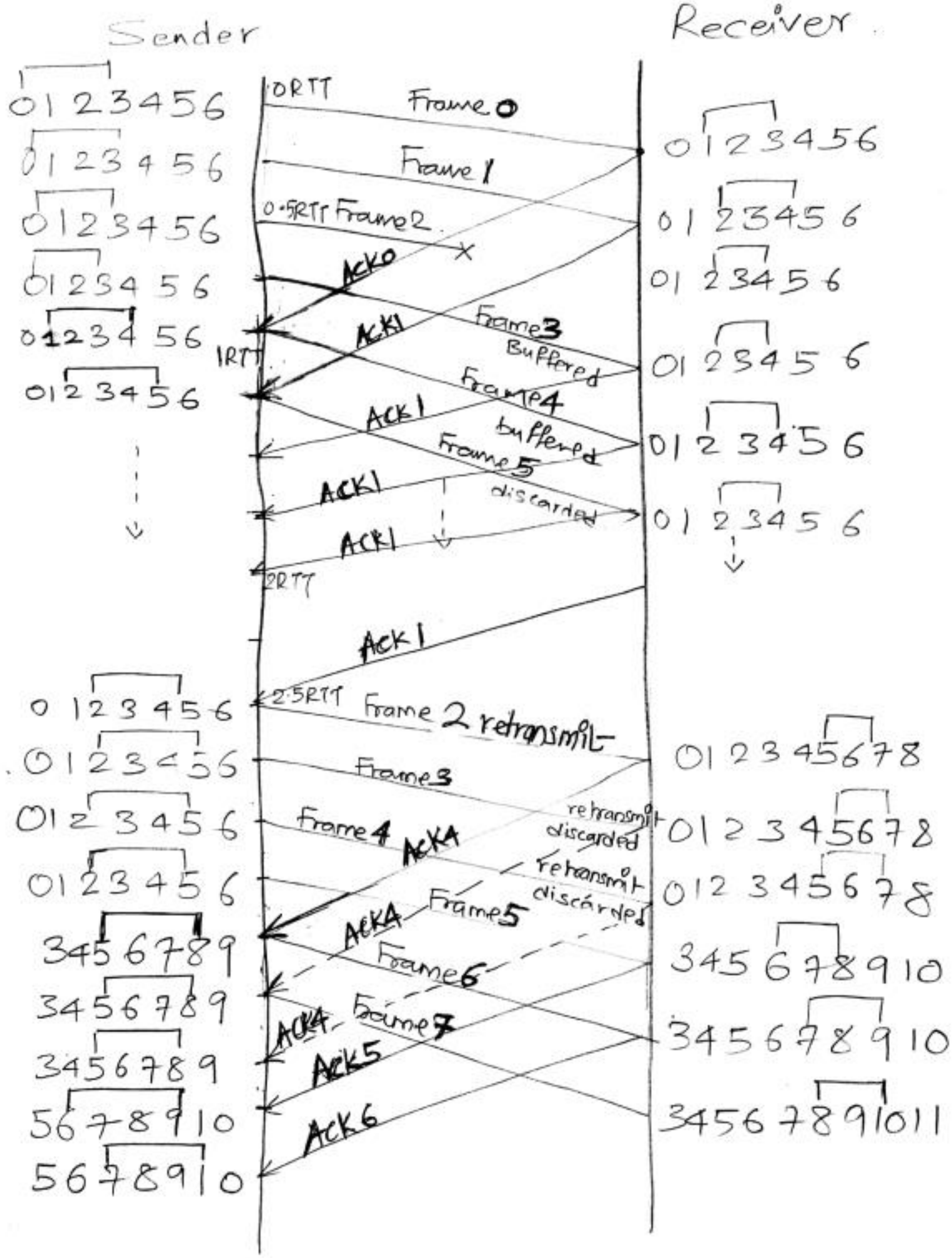
Students are expected to draw till Frame 6 is successfully transmitted.

Problem 9 – 8 pt

Draw a timeline diagram (up to frame 7) that for the sliding window algorithm with SWS=4 frames and RWS=3 frames, when the third frame (frame 2) is lost. The receiver uses cumulative ACKs. Use a timeout interval of about 2 x RTT. Assuming that the transmit time (insertion delay) of a frame is equal to 0.25 RTT and the frames can be processed instantaneously if they arrive in order. On each data frame and ACK frame, you need to indicate the sequence number (start from 0). In addition, you need to indicate what action is taken by the receiver of when it is received, when ack is sent for a frame, if a frame is discarded or if it is buffered till waiting for cumulative ACK.

Problem 9

Assume – Timeout 2RTT, 0.25RTT delay, SWS=4, RWS=3



Problem 10 – 8 pt

Consider the sliding window algorithm with SWS = RWS = 5 and MaxSeqNum = 9. The Nth packet DATA [N] contains N mod 9 in its sequence number field.

Show an example in which the algorithm becomes confused, no packets may arrive out of order. **DRAW IT OUT use the same suggested format in Problem 8 -** that is, a scenario for example which the receiver expects DATA[9] and accepts DATA[0]—which has the same transmitted sequence number. Notice that this implies that MaxSeqNum > 9 is necessary and sufficient.

As long as the student shows how packets are mistaken and the process follows the protocol, give them full credit.

Following solutions are for reference -

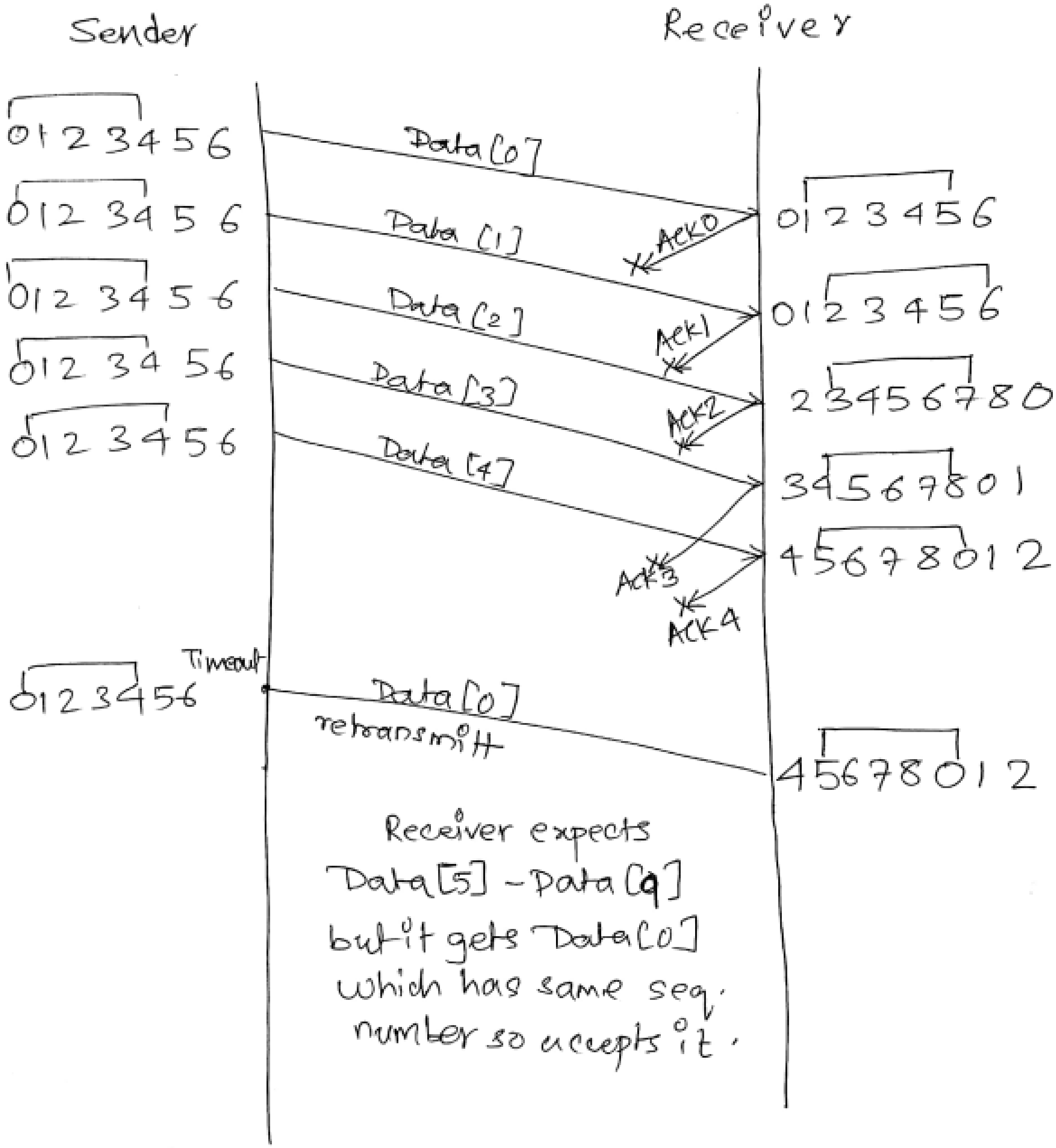
1. **Solution 1 – separately drawn in the next page for reference purpose**
2. **Solution 2 – if students have drawn a figure that depicts following scenario that too is correct -**

MaxSeqNumber = 9 means that sequence numbers range from 0 to 8. Sender transmits DATA [0 … 4] first with sequence numbers 0 … 4. Receiver receives all frames and sends a cumulative ACK, which is lost. In this case, the sender retransmits DATA [0 … 4]. Now, receiver is waiting for frames with sequence numbers 5, 6, 7, 8, 0, so the receiver thinks it has received frame 0 (DATA [9]) while it has received DATA [0]. Hence, the receiver makes an error.

Problem 10

Max. seq no. = 9 SWS = RWS = 5

seq no.s → Data[0] – Data[8]



Problem 11 – 6 pt

Suppose you are designing a sliding window protocol for a 1-Mbps point-to-point link to a stationary satellite revolving around earth at an altitude of 3 x 10^4 km. Assuming each frame carries 1KB of data.

Students should show the related formulas or some work as to how they arrived at their answers
-1pt if answer is wrong but formula is correct.

a) What is the optimal window size? (2 pt)

Propagation delay (time)

= distance/speed

= 3 x 10^4 x 1000 meters / speed

= 30,000,000 meters

30,000,000 / (3 x 10^8) = 0.1 sec

1,000,000 x 0.1 = 100,000 bits

100,000/ (1024 x 8) = 12.2 or 13 packets each way, or 26 packets round trip.

Why? - This is how many packets that will completely fill the link up and back.

b) What will the window size be if the data rate is tripled? (2 pt)

3,000,000 x 0.1 = 300,000 bits

300,000/(1024x8) = 36.62 or 37 packets each way, or 72 packets round trip.

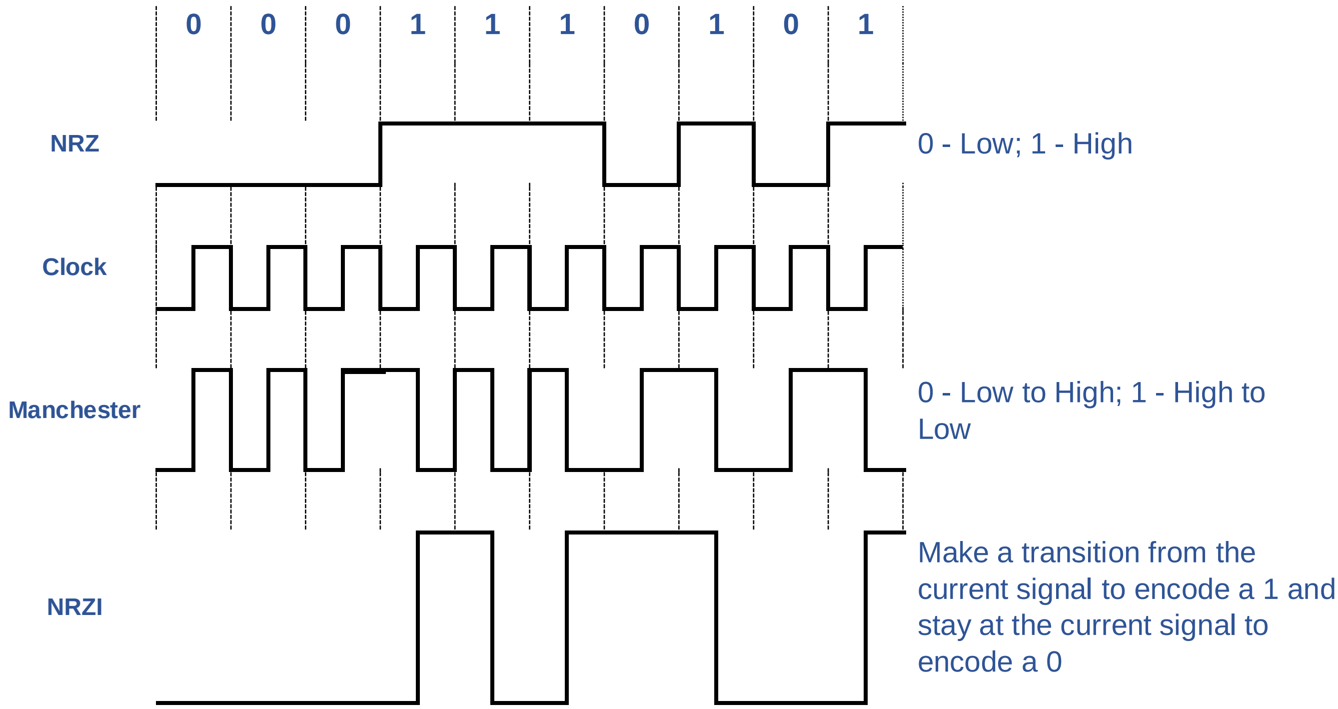
Problem 12 – 5 pt

Sketch the NRZ, Manchester, and NRZI encoding for the bit stream 0001110101. Assume that the NRZI signal starts out low.

1. Sketch the NRZ, Manchester, and NRZI encoding for the bit stream 0001110101. Assume that the NRZI signal starts out low.

- 4 pts if solutions do not match at all.
- 2 pt if solution is partially correct.

Solution:



Problem 13 – 6 pts

Suppose you are designing a sliding window protocol for a 4-Mbps point-to-point link that is 9 x 10^4 km long. Assuming each frame carries 4 KB of data, what is the minimum number of bits you need for the frame sequence numbers if –

Students should show the related formulas or some related work as to how they arrived at their answers

-1pt if answer is wrong but formula is correct.

a) RWS=1 (2 pts)

b) SWS = RWS (2 pts)

c) If we use the Stop-and-Wait ARQ protocol, what would be the maximum efficiency (defined as frames in transit/maximum possible frames in transit) that we can achieve? (2 pts)

Solutions –

Length of link = 9 x 10^7 m

File size S = 4 KB

Bandwidth B = 4 Mbps

The SWS should have enough frames to hold the packets that are needed to completely fill the pipe. This capacity, can be found using the bandwidth delay product. In this approach, we estimate the bandwidth delay product by the number of frames it takes to fill a pipe completely.

T_{delay} (propagation delay) = L/c = 0.3 s

RTT (round trip time)= 0.6 s

B_{fps} = Bandwidth/(file size S) = 123 fps (approx)

No. of frames $N_f = B_{fps} \times RTT$ = 74 frames (approx)

Thus, SWS should hold at least 74 frames.

a. RWS = 1

If RWS = 1 then the receiver can handle only one frame at a time and hence to fully utilize the capacity of the link we'll need to have at least 74 unique frame sequence numbers.

Thus, we'll need 7 bits.

b. SWS = RWS

In this case, the sender can send twice as many frames since the receiver can now store frames as well. Thus, we can send as many as 148 frames to fully utilize the link.

Thus, we'll need 8 bits.

c. In Stop-and-Wait ARQ protocol we can have at most 1 frame in transit. However, we calculated that at the maximum we can have 74 frames in transit.
   Efficiency = 1/74 = 0.014 or 1.4%