

CSE 123 Computer Networks

Homework 1 Solutions

(Total Points: 60)

Instructions:

Take a printout of this template, write/sketch answers in the spaces provided below each question, and submit the scanned copies of your homework on Gradescope. You can even write/sketch your answers digitally.

Q1 Layering and Encapsulation (10 pts)

We represent the suite of protocols that make up a network system with a protocol graph. The nodes of the graph correspond to protocols or applications. A node uses services offered by the node directly connected below it in the protocol graph and offers services to the node directly connected above it in the protocol graph.

Figure 1 shows a protocol graph for a network system consisting of various applications and protocols.

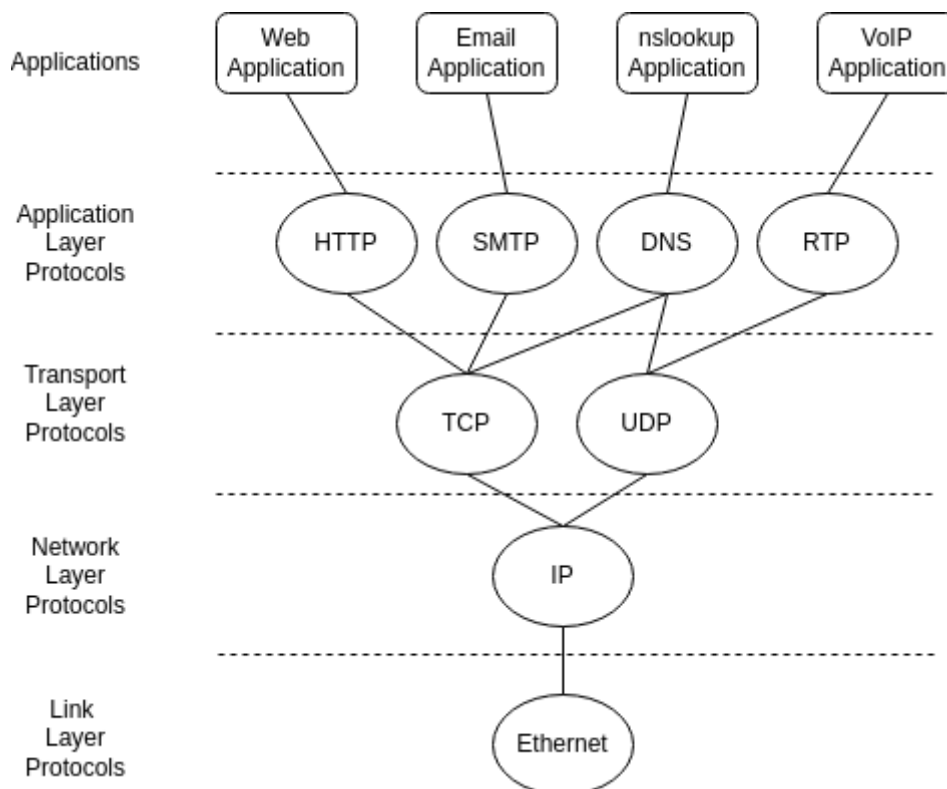
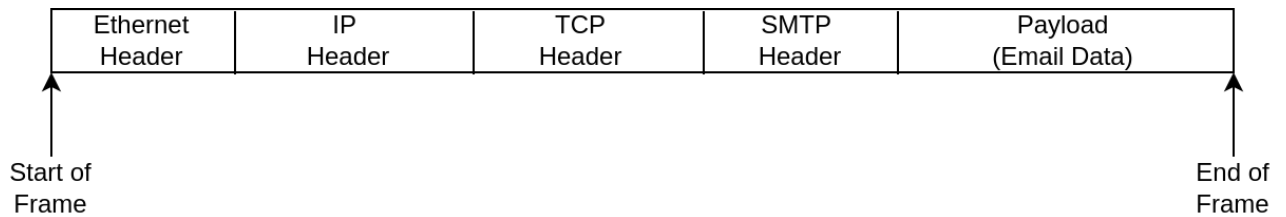


Figure 1: Protocol Graph of a Sample Network System

Referring to the above protocol graph, answer the below questions.

Q1.1 Suppose that you want to send an email using your computer that is connected to the Internet over an Ethernet network. Draw a diagram of an Ethernet frame carrying your email data as it appears on the wire. In your diagram, show the location of 1) various protocol headers, 2) application-layer payload (email data), 3) start of the frame, 4) end of the frame. (3 pts)

Sol: 2 pt for the correct location of headers and payload; 1 pt for the correct location of start and end of frame

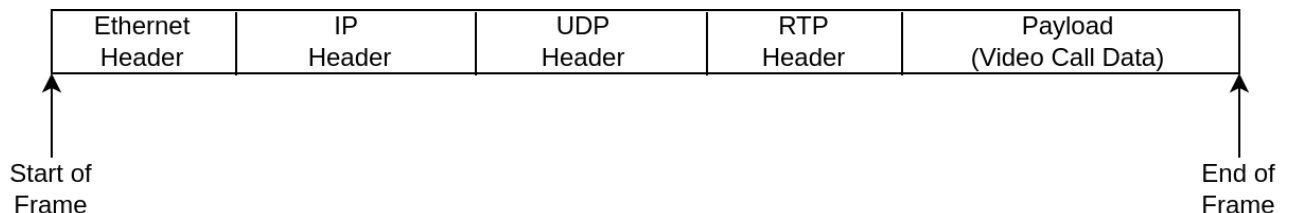


Q1.2 For the Ethernet frame you drew in Q1.1, which is the first header that your computer added to your email data? Which is the last header that your computer added to your email data? **(2 pts)**

Sol: First Header: SMTP Header **(1 pt)**; Last Header: Ethernet Header **(1 pt)**

Q1.3 Suppose that you want to make a video call to your friend using a VoIP application installed on your computer. Your computer and your friend's computer are each connected to the Internet via an Ethernet network. Draw a diagram of an Ethernet frame carrying your video call data as it appears on the wire. In your diagram, show the location of 1) various protocol headers, 2) application-layer payload (video call data), 3) start of the frame, 4) end of the frame. **(3 pts)**

Sol: 2 pt for the correct location of headers and payload; 1 pt for the correct location of start and end of frame



Q1.4 When the Ethernet frame you drew in Q1.3 reaches your friend's computer, which is the first header that your friend's computer strips? Which is the last header that your friend's computer strips? **(2 pts)**

Sol: First Header: Ethernet Header **(1 pt)**; Last Header: RTP Header **(1 pt)**

Q2 Framing (15 pt)

Given our understanding of the HDLC protocol, Figure 2 shows a bitstream received at a HDLC receiver.

Sol:

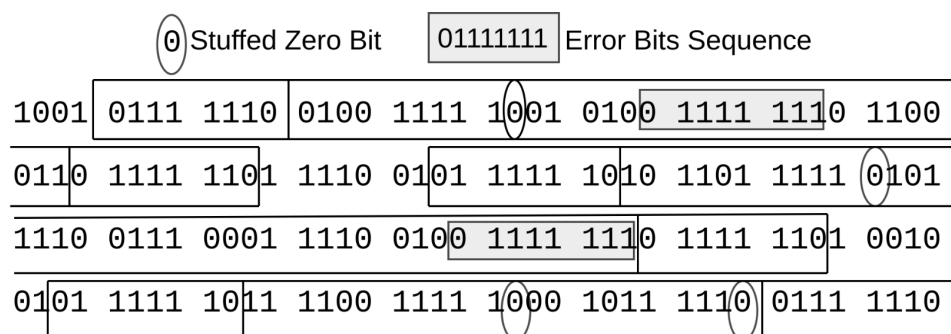


Figure 2: Bitstream received at a HDLC receiver

Note: You will get full credits if you have highlighted the **111 1111** bit pattern to indicate the error.

Q2.1 Sketch a rectangle enclosing each unique frame in Figure 2, and draw vertical boundaries to separate the start-of-frame marker, payload, and end-of-frame marker. **One frame has been sketched to give you an example. (3 pt)**

Grading Scheme: 1 pt for each rectangle correctly sketched; 0.5 pt for each pair of vertical boundaries correctly sketched.

Q2.2 In the payload of each frame in Figure 2, circle the stuffed bits. **(2 pt)**

Grading Scheme: 0.5 pt for each correct stuffed bit. -0.5 pt for each incorrect bit circled.

Q2.3 In the payload of each frame in Figure 2, if there is an error then shade the entire sequence of bits that indicate the presence of an error. **(2 pt)**

Grading Scheme: 1 pt for each sequence of bits correctly shaded.

Q2.4 For each valid frame (i.e., ones that have no errors) in Figure 2, write the payload (unstuffed version without start-of-frame and end-of-frame markers) in hexadecimal form. **(3 pt)**

Sol: Unstuffed version: 1111 0011 1110 0101 1111 ==> 0xF3E5F

Q2.5 Suppose an HDLC sender wants to send a message 0x6FB3. Write down the entire bitstream for the frame that the sender will send. Do not forget to add the bits for the start-of-frame and the end-of-frame marker. **(5 pt)**

Sol: 1 pt for the correct start of frame marker; 1 pt for the correct end of frame marker; 1 pt for the correct location of stuffed bit; 2 pt for overall correct bitstream sequence.

0111 1110 0110 1111 1001 1001 1011 1111 0

Q3 Codewords and Hamming Distance (8 pt)

Consider a code over nine-bit strings that ONLY contains the following four codewords: 000011110, 011000011, 101110101, 110101000

Q3.1 What is the hamming distance for this code? **(2 pt)**

Sol: 6 (For a set of multiple codewords, the Hamming distance of the code is the minimum hamming distance between any pair of its members)

Q3.2 How many bit flips can be detected? How many bit flips can be corrected? **(2 pt)**

Sol: 1 pt for each correct answer

Hamming distance of code is 6. It can be represented as $(2 \times 2.5) + 1$. Therefore, $d = 2.5$.

No. of Bit Flips that can be detected = $2 \times d = 2 \times 2.5 = 5$

No. of Bit Flips that can be corrected = $\text{FLOOR}(d)$
= $\text{FLOOR}(2.5) = 2$

Q3.3 What is the rate of this code if it is used to encode two-bit strings? **(2 pt)**

Sol: Rate of Code = (No. of Data Bits in Codeword / Total bits in Codeword)
= $2 / 9$

Q3.4 Is the above coding scheme efficient? Provide a reason for your claim. **(2 pt)**

Sol: No. **(1 pt)** A code is NOT efficient if you can flip the Hamming Distance number of bits and NOT end up at another codeword. Let's say we flipped 6 bits in the first codeword 000011110 such that the new pattern is 111111101. The resultant pattern is not a codeword. **(1 pt)**

Q4 Error Detection using Cyclic Remainder Check (17 pt)

Q4.1 Suppose a sender wants to send binary data 1110 011 using the CRC generator polynomial $x^5 + x^2 + 1$. What are the actual bits that get transmitted? Show your work. (9 pt)

Sol: 1 pt for correct divisor; 1 pt for each correct division output; 1 pt for correct final bitstream

$$\begin{array}{r}
 1111111 \\
 100101 \overline{) 111001100000} \\
 \underline{100101} \\
 111001 \\
 \underline{100101} \\
 111000 \\
 \underline{100101} \\
 111010 \\
 \underline{100101} \\
 111110 \\
 \underline{100101} \\
 110110 \\
 \underline{100101} \\
 100110 \\
 \underline{100101} \\
 11
 \end{array}$$

Final bitstream sent by the sender: 1110 0110 000 0 XOR 11 = 1110 0110 0011

Q4.2 Suppose a sender and receiver are using the CRC generator polynomial $x^3 + x^2 + 1$ and the receiver receives the bit string 1011 0000 11. Was the message received correctly? If so, what was it? If not, how do you know? Show your work. (7 pt)

Sol: 1 pt for correct divisor; 1 pt for each correct division output; 1 pt for correct message

$$\begin{array}{r}
 11111 \\
 1101 \overline{) 1011000011} \\
 \underline{1101} \\
 1100 \\
 \underline{1101} \\
 1000 \\
 \underline{1101} \\
 1011 \\
 \underline{1101} \\
 1101 \\
 \underline{1101} \\
 0000
 \end{array}$$

Yes, since the remainder is zero, the message is received correctly.

Length of frame check sequence bits = Degree of CRC generator polynomial = 3

Therefore, the last three bits in the message are frame check sequence bits and the rest bits are the original message. The original message is **1011 000**.

Q4.3 What is the maximum length burst error that can be reliably detected with CRC generator polynomial $x^8 + x^2 + x^1 + 1$? **(1 pt)**

Sol: Maximum length burst error that can be reliably detected using CRC = Degree of CRC generator polynomial = 8

Q5 Sliding Window Protocol (10 pts)

Consider a sender and receiver that implement a sliding window protocol using selective retransmission, as described in Section 2.5 of the textbook. Assume that:

1. Frames take an integral sequence number in the range [0,6].
2. Sender Window Size = Receiver Window Size = 3.
3. The receiver sends a cumulative ACK (ACK for the latest **in-order** frame received) each time it receives a frame from the sender.
4. One-way delay from sender to receiver and vice versa is 1 ms.
5. The sender will send no faster than 1 frame per 1 ms.
6. The sender considers a frame lost if it has not received an ACK within 5 ms of the original transmission.
7. If an ACK is received at the same instant that a sender is able to send, it should process the ACK before sending.

Figure 3 shows a series of exchanges between the sender and receiver. Figure 4 shows the state of sender and receiver after the completion of the last exchange shown in Figure 3.

Q5.1 Your task is to sketch the remaining exchanges in Figure 3 until ACK 4 reaches the sender assuming no further drops. (6 pt)

Sol:

0.75 pt each for Frame 3 and Frame 4 sent at $t = 3$ ms and $t = 4$ ms respectively.

0.75 pt each for ACK1 at $t = 4$ ms and $t = 5$ ms.

1 pt each for Frame 2 and Frame 3 re-transmitted at $t=7$ ms and $t=8$ ms respectively.

1 pt for ACK4 at $t=8$ ms

- 2 pt if $LFS - LAR \leq SWS$ condition violated at any point.

Tricky Observation: In selective re-transmission, only the lost frames are re-transmitted. But, in the below case, the timer for Frame 3 expires before it receives a cumulative ACK acknowledging Frame 3. Hence, Frame 3 gets re-transmitted.

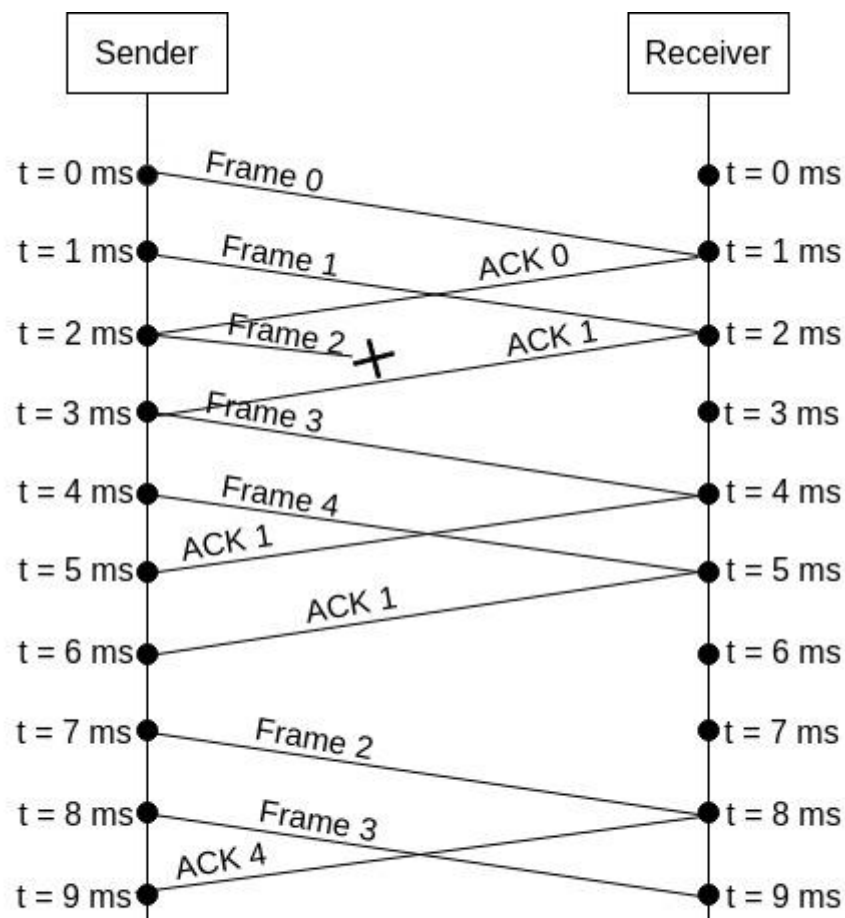


Figure 3: Series of exchanges between sender and receiver

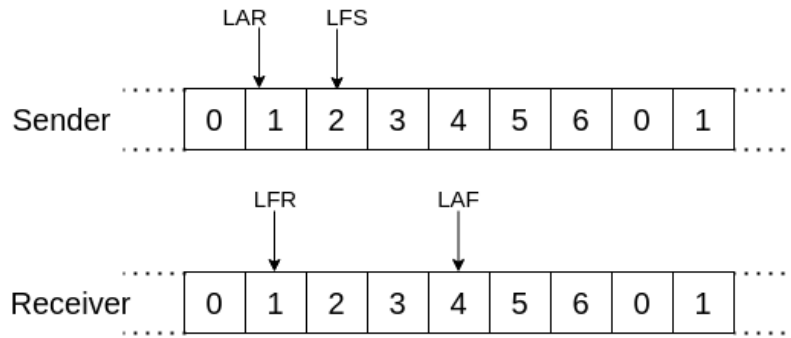


Figure 4: State of sender and receiver after the completion of the last exchange shown in Figure 3

Notation:

LAR: Last Acknowledgement Received by the sender

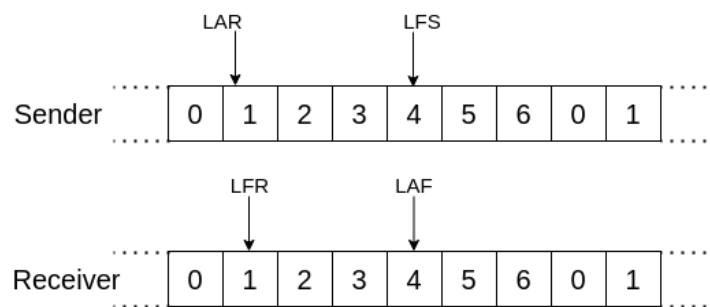
LFS: Last Frame Sent by the sender

LFR: Last **in-order** Frame Received by the receiver

LAF: Largest Acceptable Frame at the receiver

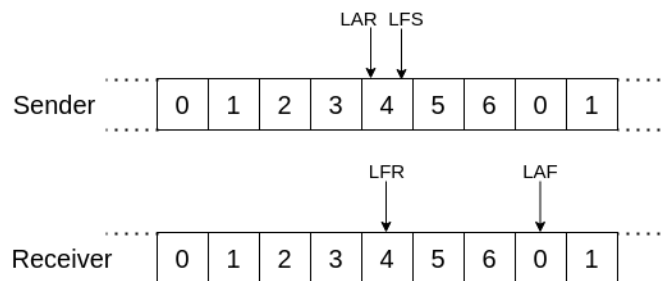
Q5.2 Consider the state of sender and receiver at time $t = 6.5$ ms. Sketch arrows pointing to the location of LAR, LFS, LFR, LAF at this point in time. (2 pt)

Sol: 0.5 pt for each arrow correctly sketched



Q5.3 Consider the state of sender and receiver when ACK 4 has arrived at the sender and the sender has not yet started its process of sending a new frame. Sketch arrows pointing to the location of LAR, LFS, LFR, LAF at this point in time. (2 pt)

Sol: 0.5 pt for each arrow correctly sketched



$LAF = LFR + \text{Sender Window Size} = 4 + 3 = 7$.

But, sequence numbers are in the range $[0, 6]$. Hence, there is a wrap-around.

Thus, LAF points to $(4+3) \text{ modulo } 7$ (here, 7 is the size of sequence number space) = 0.