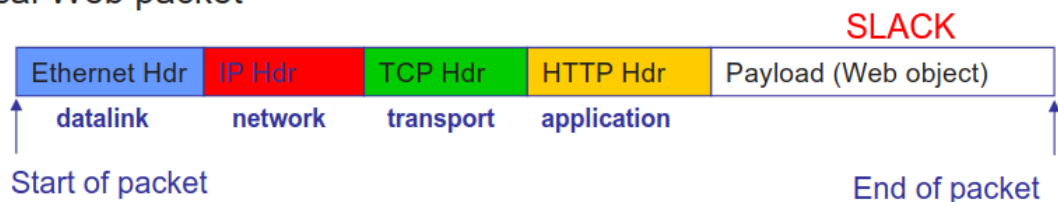


CSE 123 Fall 2026

Homework 1 Solutions

1 a) Suppose that you want to send a (short) Slack message using your computer that is connected to the Internet over an Ethernet network. Assume that Slack sends messages using HTTP requests. Draw a diagram of an Ethernet frame carrying your message as it appears on the wire. In your diagram, show the location of 1) various protocol headers, 2) application-layer payload (message), 3) start of the frame, 4) end of the frame. It is OK if you don't know the particular protocol in use at each layer, but you should indicate to which layer each header corresponds.

- Typical Web packet



Reference: <https://cseweb.ucsd.edu/classes/fa26/cse123-001/lectures/123-fa26-l2.pdf> p.8

b) For the Ethernet frame you drew in 1a), which is the first header that the receiving host will process? Which is the first header that your computer added?

Reference: <https://7e.systemsapproach.org/introduction.html#intro-3-3-encapsulation>

First processed: Ethernet Header

First added: HTTP Header

The process of encapsulation is then repeated at each level of the protocol graph; for example, IP encapsulates UDP's message by attaching a header of its own; ETH does the same with the IP packet. If we now assume that ETH sends the message to its peer over some network, then when the message arrives at the destination host, it is processed in the opposite order: ETH first interprets the ETH header at the front of the message (i.e., takes whatever action is appropriate given the contents of the header) and passes the body of the message (but not the ETH header) up to IP, which takes whatever action is indicated by the IP header that its peer attached and passes the body of the message (but not the IP header) up to UDP, which does the same. The message passed up from UDP to the application on host 2 is exactly the same message as the application passed down to UDP on host 1; the application does not see any of the headers that have been attached to it to implement the lower-level communication services. This whole process is illustrated in Figure 13. Note that in a multi-hop

2. Consider a code on four-bit strings that contains (only) the following four codewords:

0001, 1000, 0110, 1011

a) What is the Hamming distance of this code?

Reference:

<https://cseweb.ucsd.edu/classes/fa26/cse123-001/lectures/123-fa26-l3.pdf> p.10

Distances:

0001 to 1000 : 2

0001 to 0110 : 3

0001 to 1011 : 2

1000 to 0110 : 3

1000 to 1011 : 2

0110 to 1011 : 3

Minimum distance = 2

b) What is the rate of this code if we use it to encode two-bit strings? Is it efficient? If it is not efficient, please explain.

Reference:

<https://cseweb.ucsd.edu/classes/fa26/cse123-001/lectures/123-fa26-l3.pdf> p.13

Need 4 bits to encode 2, rate = $2/4 = 0.5 = 50\%$

It is not efficient, as a 4-bit string could encode 3-bit messages using 8 codewords, achieving a higher rate while still detecting single-bit errors.

c) How many bit flips can be detected using this code? How many bit flips can be corrected?

Reference:

<https://cseweb.ucsd.edu/classes/fa26/cse123-001/lectures/123-fa26-l3.pdf> p.11

Given our Hamming distance of 2

$2d + 1 = 2 \rightarrow d = 0.5$

Can detect up to $2d = 1$ bit flip

Can correct up to $d = \lfloor 0.5 \rfloor = 0$ bit flips

3. Suppose you are transmitting an 16-bit message using a protocol that protects the payload with a CRC defined by the polynomial $C(x) = x^8 + x^5 + x^4 + 1$.

Reference:

<https://7e.systemsapproach.org/technology.html#tech-1-3-error-detection> &
<https://cseweb.ucsd.edu/classes/fa26/cse123-001/lectures/123-fa26-l3.pdf> p. 18

- a) How many bits are actually required to transmit the message and the frame check sequence? How many bits correspond to the frame check sequence?

x^8 implies 8 bits for CRC
message has 16 bits
 $16 + 8 = 24$ total bits transmitted

Brief explanation:

The degree of the CRC generator polynomial is the highest exponent = 8.

When you divide by a degree-8 polynomial, the remainder has degree at most 7.

A polynomial of degree at most 7 can be represented with 8 bits.

- b) Suppose that exactly one bit error occurred during transmission. Will the receiver be able to detect it? Why or why not?

Yes, because $C(x)$ has non-zero coefficients x^8 and x^0 (1).

- All single-bit errors, as long as the x^k and x^0 terms have nonzero coefficients

- c) On the contrary, suppose the receiver received the following bits without error: 010101011010101011111111. What was the payload?

Since there was no error, the message is divided into payload + CRC bits

0101010110101010 + 11111111

Payload = 0101010110101010 (i.e., the first 16 bits)

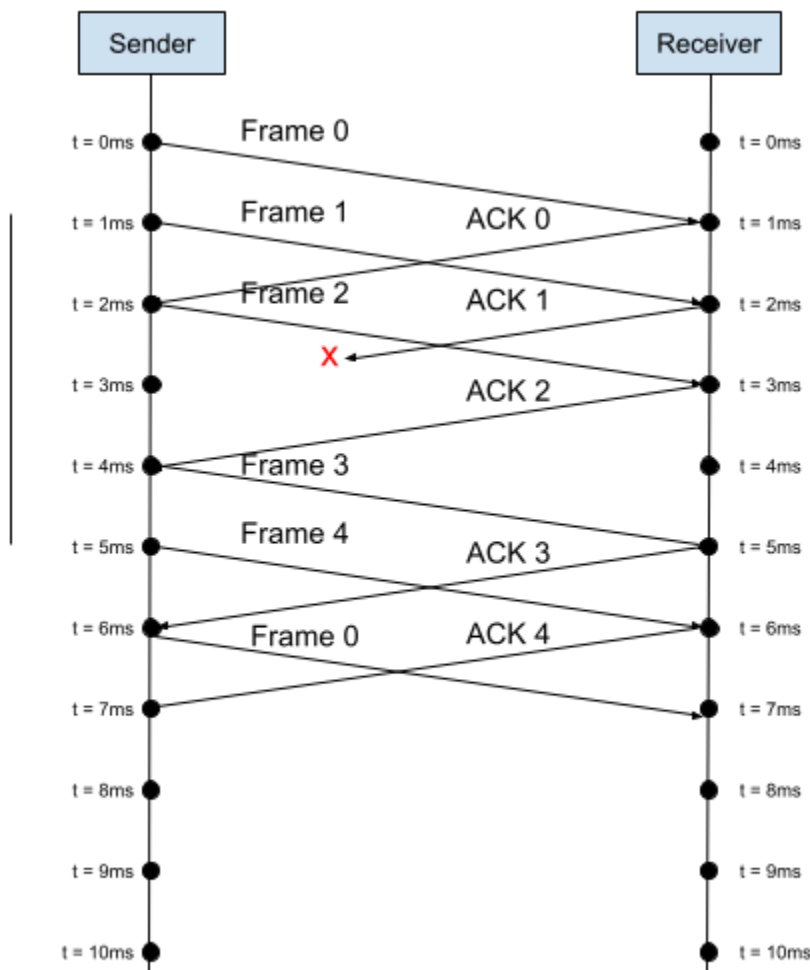
4. Consider a sender and receiver that implement a sliding window protocol using selective retransmission, as described in Section 12.4 of the textbook. Assume that:

- The frame sequence number space ranges from 0 to 4
- The send window and receive window size are 2.
- The receiver sends a cumulative ACK each time it receives a frame from the sender
- The one-way delay between sender and receiver is *slightly less* than 1ms
- The sender sends frames no faster than once per millisecond
- The sender uses a timeout of 4 ms.

The Figure below shows a series of frame exchanges between the sender and receiver

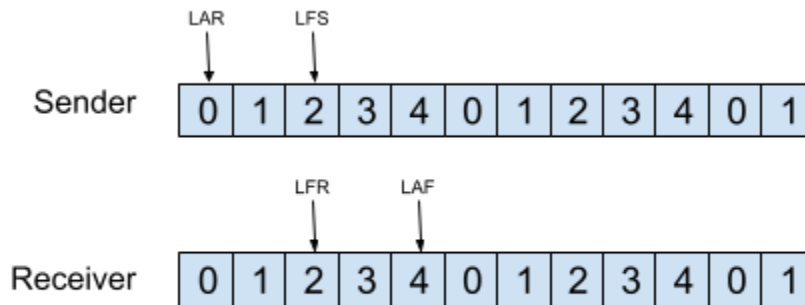
Reference:

<https://cseweb.ucsd.edu/classes/fa26/cse123-001/lectures/123-fa26-l4.pdf> p. 6-14 & <https://7e.systemsapproach.org/reliable.html#tcp-4-sliding-window>



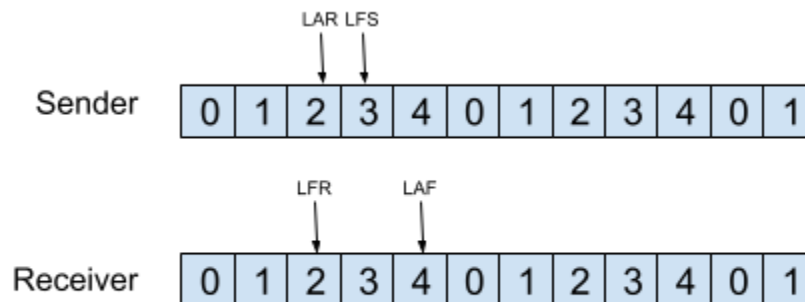
a) As you can see, the transmission of ACK 1 was lost. You should extend the diagram above until the sender receives the ACK for Frame 4 assuming no further frame (or ACK) losses.

The Figure below depicts the state of the sender and receiver at $t=3\text{ms}$.



Where LAR stands for Last Acknowledgement Received, LFS is Last Frame Sent, LFR is Last (in-order) Frame Received, and LAF is Largest Acceptable Frame.

b) Using your extended trace from the prior question, draw a similar state diagram for the Sender and Receiver at $t=4.5\text{ms}$. Include arrows for LAR, LFS, LFR, and LAF.



c) And, finally, draw a similar diagram at the end of the trace from part (a) (i.e., the moment after the Sender receives the ACK for Packet 4 but before it sends anything further).

