

Question 1 Identifying frame

1.1 Why is a fixed-length frame not a good idea? Please list two reasons.

Solution:

1. The large length might cause way too many spaces wasted on the small payload.

Solution:

2. The small lengths might cause way too many fragmentations for large payloads.

1.2 Why is a length-based frame (e.g length of frame explicitly written in the header) not a good idea? Please list two reasons.

1. The size part might get corrupted in transition.

2. Wastes space to include the length field.

(Alternative reasons: harder to program/design the receiver because it needs a variable-length buffer.)

1.3 Sentinel-based framing. Assuming the sentinel bit pattern is 01111110 (0X7E).

Each of the following bit patterns arrived at an HDLC receiver. For each pattern, determine whether it is a valid bit pattern (i.e., after the sender performs stuffing)? If it is valid, please give the original, unstuffed version of the payload in hexadecimal form. Append 0s at the end if there are not enough bits to form a hex representation. If it is not valid, please explain why--i.e., highlight any bits you believe to be in error. (The red bits represent the sentinels signaling the beginning and end of the frame.)

1.3.1

0111 1110 0101 1101 1101 0111 1110 0111 1110

Solution: It is not valid. 0111 1110 0101 1101 1101 0111 1110 0111 1110

We found the sentinel pattern again inside the frame. A correct stuffed bit pattern won't include the sentinel bit pattern.

1.3.2

0111 1110 0101 1111 0111 1100 1011 0111 1001 1111 10

Unstuffed: 0111 1110 0101 1111 1111 1010 1101 1110 0111 1110

Solution: It is valid, the unstuffed bit pattern would be 5 F F A D E

Question 2 Error Handling

2.1 Hamming Distance

Consider a code on six-bit strings that contains (only) the following four codewords:
000000, 000011, 001111, 111111

2.1.1 What is the hamming distance of this code?

The minimal number of different bit among each codeword is at least 2. Thus, the hamming distance is 2.

2.1.2 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.

This code uses 6 bits codewords to represent information that can be represented by 2 bits. Its rate is $\frac{2}{6} = 33\%$.

It is not efficient. An efficient scheme should have a uniform distance among neighbor codewords. 111111 has neighbor codewords as 000000, 001111. 111111 and 000000 have distance as 6. But 111111 and 001111 have hamming distance 2.

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

Since the Hamming distance in this code is 2, $2d + 1 = 2$, $2d = 1$.

We can detect up to $2d$ bit flips, which is 1 bit in this case.

We can correct up to d bit flips, which is 0 bit in this case ($d = \frac{1}{2} < 1$).

2.2 CRC

Suppose a sender and a receiver are using the CRC generator polynomial $x^4 + x^3 + 1$.

2.2.1 The receiver receives the bit string 1010 1010 0101. Was the message received correctly? If so, what was it? If not, how do you know? Show your work.

The generator would be 11001

```
1010 1010 0101
```

```
1100 1
```

```
-----  
0110 0010 0101
```

```
110 01
```

```
-----  
110 0101
```

```
110 01
```

```
-----  
1
```

The message was not received correctly. It does not come to 0 after using the generator to verify the message.

2.2.2 The receiver receives the bit string 1010 0111 1110. Was the message received correctly? If so, what was it? If not, how do you know? Show your work.

```
1010 0111 1110
```

```
1100 1
```

```
-----  
110 1111 1110
```

```
110 01
```

```
-----  
1011 1110
```

```
1100 1
```

```
-----  
111 0110
```

```
110 01
```

```
-----  
1 0010
```

```
1 1001
```

```
-----  
1011
```

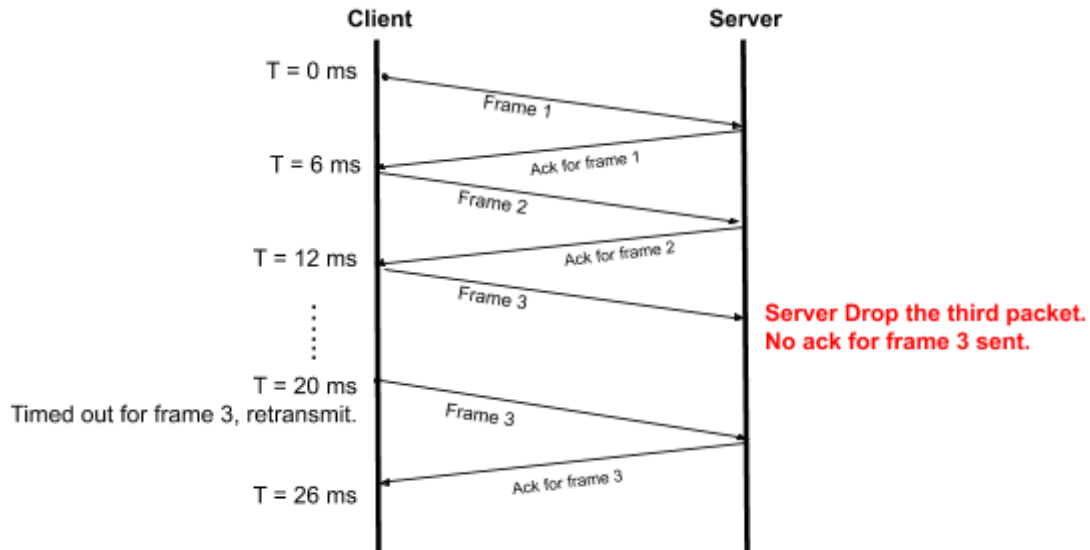
The message was not received correctly

2.2.3 What is the max burst error that can be detected with generator $x^4 + x^3 + 1$?

4 bits.

Question 3 Flow Control

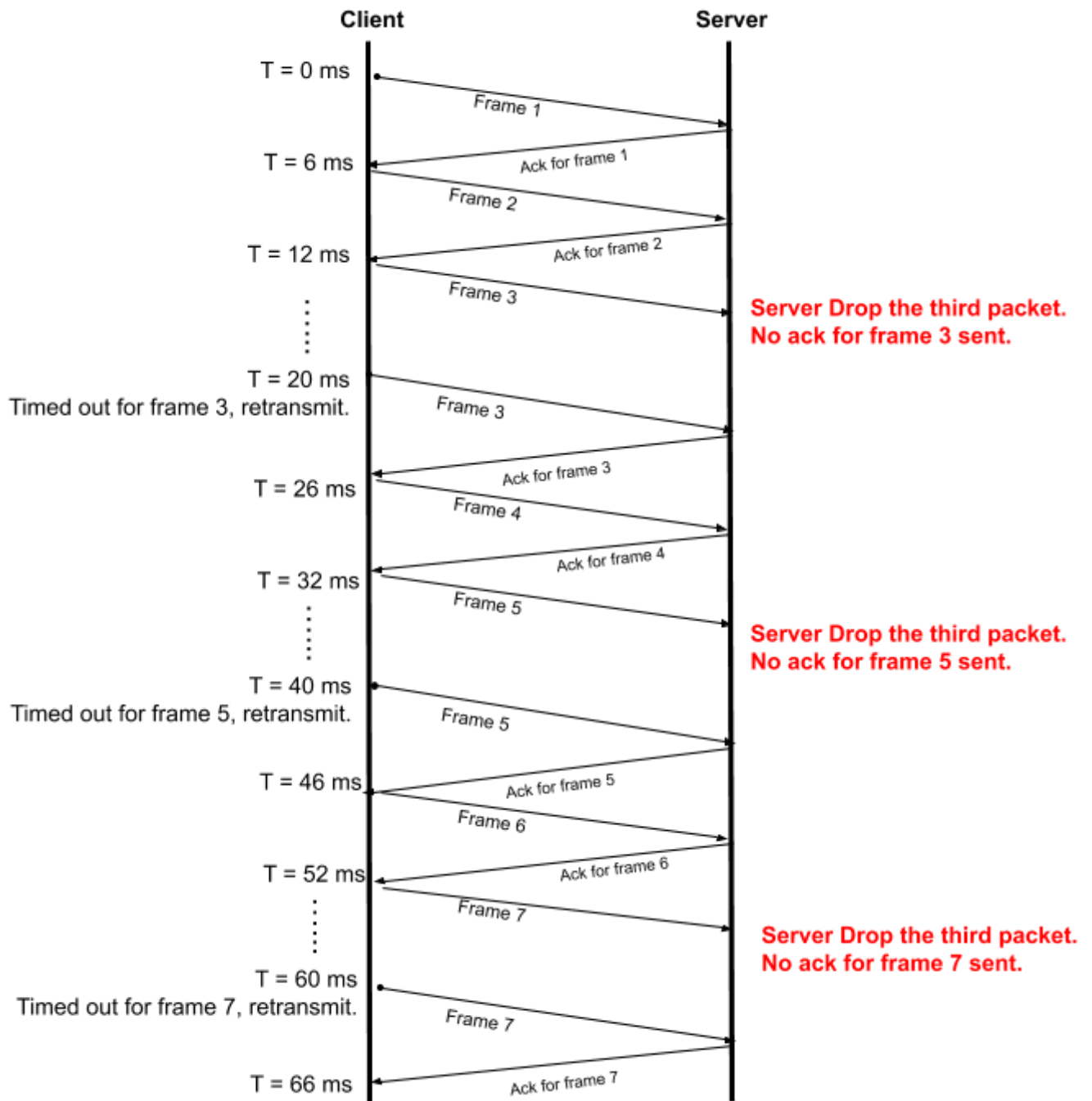
A client and a server are transmitting data frames using the stop-and-wait ARQ protocol. Assume Round Trip Time (RTT) is 6 ms; The client timeout is 8 ms; The server drops every third transmission received from the client (server will not send acknowledgment for that data frame); sequence number is correctly encoded in every data and acknowledgment frame; The client transmits the first frame at time 0. (Here is a message exchange diagram for you to understand what is happening)



3.1 Draw out the complete message exchange diagram until the client successfully communicates (i.e., receives explicit acknowledgments for) 7 frames.

Solution Diagram in the next page

(Intentionally left blank for drawing)



3.2. How many frames has the client already transmitted, including retransmissions, when it receives an acknowledgment of the 7th frame from the server?

1, 2, 3x, 3, 4, 5x, 5, 6, 7x, 7

10 frames. 3, 5, 7 would get dropped and transmit twice.

3.3. Compute the total time (in ms), including retransmissions, when the client receives an acknowledgment of the 7th frame from the server.

3 time out and 7 regular round trip => $3 * 8\text{ms} + 7 * 6\text{ms} = 24\text{ ms} + 42\text{ ms} = 66\text{ ms}$

Question 4 Transfer Layer Protocols

4.1 Present an example of usage of UDP and explain the benefit of it. Why TCP might be a bad choice in this example?

Streaming service can use UDP. TCP provides correctness but for streaming service, TCP will pause the delivery of the data while waiting for retransmissions of lost packets, leading to jitter/stutter in the video display. The overhead of TCP might decrease the continuity of video streaming. On the other hand, UDP will just try its best to deliver the packet to the client. While there might be some lost packets in the middle of video streaming, it can still let you watch the video without much pause or stutter.

4.2 Present an example of usage of TCP and explain the benefit of it. Why UDP might be a bad choice in this example?

When you deliver a web page, it contains a lot of information that required to be 100% correct. Especially when it comes to authentication and verification on the web page. In such a case, you would like to have an exact page as you expect it to be. By using UDP, you are likely to lose important pieces of information on the web page since UDP is not as reliable as TCP.

