General Instructions:

* All work must be completed using MS Word, Excel, or Visio
  + Do NOT attach pictures of anything handwritten or hand-drawn to this document—it will not be graded if you do
  + You may embed Visio drawings or insert them as screenshots within the document
  + You may use Word’s Draw feature for encoding problems
  + You may use drawings, tables, and figures as a part of your answers
  + You may use bulleted phrases or complete paragraphs in your answers
  + Make sure you provide an answer to each question and that the answers you provide actually answer the questions
* Show ALL work when calculating values
  + For all problems involving file size, assume the numbers are based on 1K = 1,024 bytes
  + For all problems with network speeds or distance, assume the numbers are based on 1K = 1,000 bytes
  + Show all bandwidth in bits per second
* Submit this document to Canvas when you have completed the exam.

Use the following diagram and information for the next two questions.

Diagram

Description automatically generated with low confidence

Link speeds:

* Link 1 – 100 Mbps
* Link 2 – 50 Gbps
* Link 3 – 30 Gbps
* Link 4 – 1 Gbps

Forwarding delays between packets:

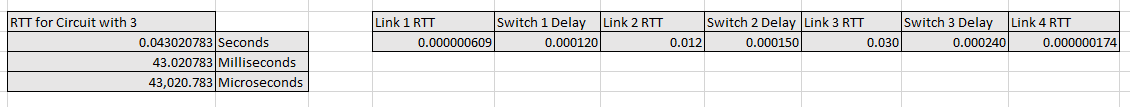
* Switch A – 60 µs
* Switch B – 75 µs
* Switch C – 120 µs

A 2 RTT handshake is required before data can be transmitted.

Your calculations should be accurate to seven decimal points.

For each question, put only the answer in this document—no calculations. Attach the spreadsheet where you did your calculations, and I will refer to it as needed.

# What is the RTT for the circuit.



0.043020783 seconds

# What is the total time, in seconds, required to transfer a 40 GB file over the circuit?

A picture containing text, screenshot, line, font

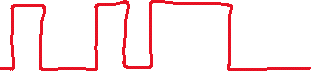
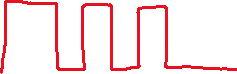
Description automatically generated

11,299.487631 seconds (or 3 hours, 8 minutes, and 19 seconds)

# Show the 4B/5B encoding and the resulting NRZI signal for the following bit sequence. Assume the NRZI signal starts low.

1101 1000 0101 1110





# Decode the following HDLC frame.

011111101100011111011000001110111110111110000011101010111011101111110



HDLC uses the distinguished bit sequence “01111110” to denote the start and end of frame, allowing clock synchronization before the message is received. Because we don’t want to encounter 6 consecutive 1’s unless we are looking at the distinguished bit sequence, HDLC pads sequences with 6 or more consecutives 1’s with a 0. To decode this, we first remove the 8 bit distinguishing sequence in the beginning and end. Then, we go through the message and look for 5 consecutive 1’s. If the next bit after 5 consecutive 1’s is a 0 followed by a 1, we remove the padded 0. I highlighted all of the padded bits in red. By removing all the bits in red, we get the following sequence:

110001111111000001110111111111100000111010101110111

# If the last three bytes of data were ETX, A, and DLE, what values would be in the five bytes before the CRC portion of a PPP frame?

Because we need to escape certain characters in data, we will need to pad the data with those escape characters. That would turn the last three bytes of data to:

DLE ETX A DLE DLE

Which is also, coincidentally, the last 5 bytes before the CRC portion. In PPP, the data comes directly before the CRC bytes.

# The following message arrived that is protected by the following CRC polynomial.

* Does the message have an error or not? Why?
* In the below table, show your work that allowed you to make the determination of whether an error exists in the message or not. Add rows and columns to the below table as necessary.

M(x): 11010011100011

CRC-3-GSM: x3 + x + 1

CRC that was in the message: 011

|  |  |
| --- | --- |
|  | X0 |
| T(x) |  |
| C(x) |  |
| Remainder |  |

A picture containing text, number, screenshot, plot

Description automatically generated

This message does not have an error in it; the CRC is correct.

# Draw a timeline that shows the transmission and acknowledgements for eight frames. Assume a sliding window protocol is in use where SWS = RWS = 3. Use a timeout interval of 2 RTTs. Show the transmission of Frames 1 thru 7 and the ACKs for those transmissions. Assume Frame 2 is lost before it gets to the receiver.

A picture containing line, diagram, parallel

Description automatically generated

Use the following AS diagram for the next three questions:

Diagram

Description automatically generated

# Complete the BGP routing table for the AS. Add rows as necessary.

BGP Table for AS

|  |  |
| --- | --- |
| Prefix | BGP Next Hop |
| 98.98/12 | R2 |
| 122.4.32/20 | R1 |
| 72.11.22/24 | R5 |
| 198.96.222/24 | R7 |

# Complete the IGP Routing Table for each router. Add rows as necessary.

R1 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R2 | R6 |
| R3 | R6 |
| R4 | R6 |
| R5 | R6 |
| R6 | R6 |
| R7 | R6 |

R2 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R1 | R6 |
| R3 | R6 |
| R4 | R6 |
| R5 | R6 |
| R6 | R6 |
| R7 | R6 |

R3 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R1 | R4 |
| R2 | R4 |
| R4 | R4 |
| R5 | R5 |
| R6 | R4 |
| R7 | R4 |

R4 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R1 | R6 |
| R2 | R6 |
| R3 | R3 |
| R5 | R3 |
| R6 | R6 |
| R7 | R7 |

R5 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R1 | R3 |
| R2 | R3 |
| R3 | R3 |
| R4 | R3 |
| R6 | R3 |
| R7 | R3 |

R6 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R1 | R1 |
| R2 | R2 |
| R3 | R4 |
| R4 | R4 |
| R5 | R4 |
| R7 | R4 |

R7 IGP Table

|  |  |
| --- | --- |
| Router | IGP Path |
| R1 | R4 |
| R2 | R4 |
| R3 | R4 |
| R4 | R4 |
| R5 | R4 |
| R6 | R4 |

# Complete the Combined Routing Table for each router. Add rows as necessary.

R1 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 98.96/22 | R6 |
| 72.11.22/24 | R6 |
| 198.96.222/24 | R6 |

R2 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 122.4.32/20 | R6 |
| 72.11.22/24 | R6 |
| 198.96.222/24 | R6 |

R3 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 122.4.32/20 | R4 |
| 98.96/12 | R4 |
| 72.11.22/24 | R5 |
| 198.96.222/24 | R4 |

R4 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 122.4.32/20 | R6 |
| 98.96/12 | R6 |
| 72.11.22/24 | R3 |
| 198.96.222/24 | R7 |

R5 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 122.4.32/20 | R3 |
| 98.96/12 | R3 |
| 198.96.222/24 | R3 |

R6 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 122.4.32/20 | R1 |
| 98.96/12 | R2 |
| 72.11.22/24 | R4 |
| 198.96.222/24 | R4 |

R7 Combined Table

|  |  |
| --- | --- |
| Prefix | IGP Path |
| 122.4.32/20 | R4 |
| 98.96/12 | R4 |
| 72.11.22/24 | R4 |

# Using the network diagram below, complete the virtual circuit tables (next page) for all switches after the following connections have been established.

* 1. Add rows to each virtual circuit table as needed
  2. Assume that the sequence of connections is cumulative; that is, the first connection is still up when the second connection is established and so on.
  3. Assume that the VCI assignment always picks the lowest unused VCI on each link, starting with zero.

**Connections to be made:**

1. Host B connects to Host E
2. Host C connects to Host F
3. Host D connects to Host E
4. Host A connects to Host E
5. Host F connects to Host A
6. Host D connects to Host B

Diagram

Description automatically generated with low confidence



B1 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 3 | B | 0 | A | 0 |
| 4 | N/A | N/A | A | 1 |
| 5 | A | 2 | N/A | N/A |
| 6 | B | 1 | A | 3 |

B2 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 3 | N/A | N/A | B | 0 |
| 6 | N/A | N/A | B | 1 |

B3 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | A | 0 | C | 0 |
| 3 | B | 0 | C | 1 |
| 4 | B | 1 | C | 2 |
| 6 | B | 2 | A | 1 |

B4 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 2 | N/A | N/A | A | 0 |

B5 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 2 | A | 0 | N/A | N/A |
| 3 | B | 0 | C | 0 |
| 4 | B | 1 | C | 1 |
| 5 | N/A | N/A | B | 2 |
| 6 | B | 3 | C | 2 |

B6 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | A | 0 | N/A | N/A |
| 3 | A | 1 | N/A | N/A |
| 4 | A | 2 | N/A | N/A |

B7 Virtual Circuit Table

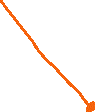
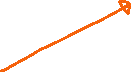
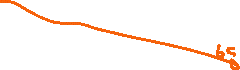
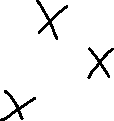
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | N/A | N/A | A | 0 |
| 6 | A | 1 | N/A | N/A |

# Given the following information, identify the links that form the spanning tree for the network diagram below. **List the links in numerical order.**

**Links:**

A picture containing text, sky, transport, ski tow

Description automatically generated



Links remaining: 1, 2, 5, 7, 8, 10

Link Bandwidths

|  |  |
| --- | --- |
| Link | Bandwidth |
| 1 | 10Gbit |
| 2 | 100Gbit |
| 3 | 1Gbit |
| 4 | 100Mbit |
| 5 | 1Gbit |
| 6 | 1Gbit |
| 7 | 10Gbit |
| 8 | 100Gbit |
| 9 | 10Gbit |
| 10 | 100Gbit |
| 11 | 10Gbit |
| 12 | 100Mbit |

Costs

|  |  |
| --- | --- |
| Bandwidth | Cost |
| 100Mbit | 50 |
| 1Gbit | 28 |
| 10Gbit | 15 |
| 100Gbit | 6 |

# In the tcp\_connection.pcapng file, what is the demultiplexing key used by all messages?

# Given the following information from Wireshark, what is the windows scaling factor?

Text

Description automatically generated

(192.168.200.135, 7876, 192.168.200.21, 2000)

# Given the following information from Wireshark, what would the next sequence number be?

Text

Description automatically generated

We take the current sequence number, in this case 1, and we add the TCP segment length, in this case 358. Therefore, the next sequence number is 359.

# When using a sliding window algorithm at the transport layer, what mechanism does receiver use to slow the sender?

The receiver uses window-based flow control to prevent the sender from sending more information than the receiver can hold. The receiver advertises their receive window size to the sender, and as the sender starts filling up the receiver’s buffer, the window size will decrease. The sender will then adjust accordingly and send less and less data. If the buffer ever gets full, the receiver will start dropping packets, causing the sender to stop getting ACKs back on the packets they send. When the sender eventually gets an ACK that opens the advertising window, the sender will then start transmitting using the slow start method to gradually speed up the transmission instead of flooding the network with all the missing packets.

# What is the size of the TCP header in the very first message of the tcp\_connection.pcapng file?

The first message has a TCP header size of 40 bytes. The default TCP header is 20 bytes, and this message has 20 bytes of options. We can see this in multiple places:

A screenshot of a computer

Description automatically generated

# In the tcp.pcappng file, message #26 has the PSH and ACK flags set. Explain the purpose of each of those flags.

In message 26, 192.168.200.21 is sending a message to 192.168.200.135. We know that 192.168.200.21 is the destination address for the HTTP communication captured in tcp.pcappng. The PSH flag is set to tell the sender (192.168.200.135) to flush its buffer of any unsent bytes, indicating that the receiver has been able to process all the previously received data and the sender can continue sending. The ACK flag being set, in conjunction with the PSH flag, acknowledges the receiver has processed and received all the data from the sender appropriately, and the sender can continue.

# Fill in the MPLS routing tables for network shown in the below diagram. Use the following messages to populate the tables. Add rows to the tables as necessary.

Diagram

Description automatically generated

1. Router R3 binds the label value 15 to the prefix 15.45/16
2. Router R3 advertises that the label value 15 is bound to the prefix 15.45/16
3. Router R6 binds the label value 82 to the prefix 88.90.0/22
4. Router R6 advertises that the label value 82 is bound to the prefix 88.90.0/22
5. Router R2 binds the label value 46 to the prefix 88.90.0/22
6. Router R2 advertises that the label value 46 is bound to the prefix 88.90.0/22
7. Router R4 binds the label value 33 to the prefix 148.2.3/24
8. Router R4 advertises that the label value 33 is bound to the prefix 148.2.3/24
9. Router R2 binds the label value 18 to the prefix 148.2.3/24
10. Router R2 advertises that the label value 18 is bound to the prefix 148.2.3/24
11. Router R5 binds the label value 90 to the prefix 18.3.129/22
12. Router R5 advertises that the label value 90 is bound to the prefix 18.3.129/22
13. Router R2 binds the label value 24 to the prefix 18.3.129/22
14. Router R2 advertises that the label value 24 is bound to the prefix 18.3.129/22

R1 Routing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Prefix | Interface | Remote Label |
|  | 15.45/16 | C | 15 |
|  | 88.90.0/22 | B | 46 |
|  | 148.2.3/24 | B | 18 |
|  | 18.3.129/22 | B | 24 |

R2 Routing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Prefix | Interface | Remote Label |
| 46 | 88.90.0/22 | A | 82 |
| 18 | 148.2.3/24 | D | 33 |
| 24 | 18.3.129/22 | C | 90 |

R3 Routing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Prefix | Interface | Remote Label |
| 15 | 15.45/16 | A |  |

R4 Routing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Prefix | Interface | Remote Label |
| 33 | 148.2.3/24 | B |  |

R5 Routing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Prefix | Interface | Remote Label |
| 90 | 18.3.129/22 | B |  |

R6 Routing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Label | Prefix | Interface | Remote Label |
| 82 | 88.90.0/22 | B |  |

# What is the 16-bit checksum for the following stream of bits?

0111 1111 0010 0000 1000 1011 0011 0011 1100 0000 1111 1101

First, we add the first 16 bits to the second 16 bits:

0 0111 1111 0010 0000

0 1000 1011 0011 0011

1 0000 1010 0101 0011

0 0000 0000 0000 0001

0 0000 1010 0101 0100

Now, we add the last 16 bits to the above result:

0 0000 1010 0101 0100

0 1100 0000 1111 1101

0 1100 1011 0101 0001

Now we must bitwise negate the sum, so we end with:

0 0011 0100 1010 1110

# Extra Credit (10 points) A router has the following four active flows, and each flow has packets to transmit. Given FQ is being used on the router, what is the order that the packets will be transmitted?

Flow 1

|  |  |
| --- | --- |
| Packet | *Pi* |
| A | 1500 |
| B | 1500 |
| C | 1500 |

Flow 2

|  |  |
| --- | --- |
| Packet | *Pi* |
| D | 40 |
| E | 40 |
| F | 40 |
| G | 1500 |
| H | 1500 |
| I | 1500 |

Flow 3

|  |  |
| --- | --- |
| Packet | *Pi* |
| J | 1500 |
| K | 1500 |

Flow 4

|  |  |
| --- | --- |
| Packet | *Pi* |
| L | 40 |
| M | 1500 |

Using the equations from chapter 6.2.2, we can calculate the theoretical finish time if the router was implementing a bit-by-bit round robin style of queueing. We can then select the smallest packet transmission times and service those first. Let’s just assume an arbitrary 1000 bps bandwidth. We can then calculate all the packet theoretical transmission times:

A: F = A + P/R = 0 + 1500/1000 = 1.5 ms

B: F = max(F\_A, A\_B) + P/R = max(1.5, 3) + 1500/1000 = 4.5 ms

C: F = max(F\_B, A\_C) + P/R = max(4.5, 6) + 1500/1000 = 7.5 ms

D: F = A + P/R = 0 + 40/1000 = 0.04 ms

E: F = max(F\_D, A\_E) + P/R = max(0.04, 1) + 40/1000 = 1.08 ms

F: F = max(F\_E, A\_F) + P/R = max(1.08, 2) + 40/1000 = 2.04 ms

G: F = max(F\_F, A\_G) + P/R = max(2.04, 4) + 1500/1000 = 5.54 ms

H: F = max(F\_G, A\_H) + P/R = max(5.54, 6) + 1500/1000 = 8.54 ms

I: F = max(F\_H, A\_I) + P/R = max(8.54, 8) + 1500/1000 = 10.04 ms

J: F = A + P/R = 2 + 1500/1000 = 3.5 ms

K: F = max(F\_J, A\_K) + P/R = max(3.5, 4) + 1500/1000 = 5.5 ms

L: F = A + P/R = 0 + 40/1000 = 0.04 ms

M: F = max(F\_L, A\_M) + P/R = max(0.04, 2) + 1500/1000 = 3.54 ms

After calculating the finish times for each packet, we can sort them in ascending order, breaking ties by which flow came first, and get the order of transmission:

D (F = 0.04 ms)

L (F = 0.04 ms)

E (F = 1.08 ms)

A (F = 1.5 ms)

F (F = 2.04 ms)

J (F = 3.5 ms)

M (F = 3.54 ms)

B (F = 4.5 ms)

K (F = 5.5 ms)

G (F = 5.54 ms)

C (F = 7.5 ms)

H (F = 8.54 ms)

I (F = 10.04 ms)

So, the correct order is: DLEAFJMBKGCHI