Problem 1

Consider the Host A in Figure 2 is trying to send the following IP packet:

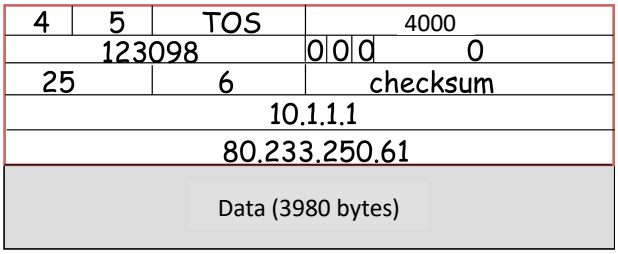

Figure 1: An IP packet.

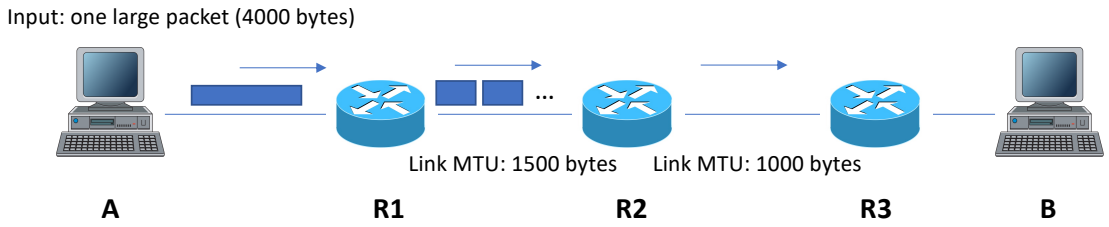

Figure 2: Network Topology.

Assuming that the maximum transmission unit that can be transferred over the link between R1 and R2 is 1500 bytes, the maximum transmission unit that can be transferred over the link between R2 and R3 is 1000 bytes. For each of the fragment transmitted over link R1-R2 and R2-R3: show the total length, identification, flags, fragment offset, TTL, and IP payload size.

From R1 to R2:

Total Length (Bytes)	Identification	Flags	Fragment offset	TTL	Data size (Bytes)
1500	123098	001	0	24	1480
1500	123098	001	185	24	1480
1040	123098	000	370	24	1020

From R2 to R3:

Total Length (Bytes)	Identification	Flags	Fragment offset	TTL	Data size (Bytes)
996	123098	001	0	23	976
524	123098	001	122	23	504
996	123098	001	0	23	976
524	123098	001	122	23	504
996	123098	001	0	23	976
64	123098	000	122	23	44

Problem 2

Assume there are 3 streams over a QUIC connection, each stream continuously sends frames (starting from sequence number 1). Assume each stream has equal priority, and a QUIC packet can contain 6 frames in total. NOTE: for frame 1 of stream 1 we can use S1-1 to represent it's number.

1. Assume there is no packet loss, what are the frame numbers contained in the first two QUIC packets.
2. If the fourth QUIC packet is lost, which frames need to be put back into the transmission queue for future packet?
3. If the fourth QUIC packet was lost, what will be contained in the ACK frame of the sixth QUIC packet (the value of Largest Acknowledged, ACK Block Count, packet number range of each ACK Block, Gaps).

1. The first packet will have frames S1-1, S1-2, S2-1, S2-2, S3-1, and S3-2. The second packet will have frames S1-3, S1-4, S2-3, S2-4, S3-3, and S3-4.
2. S1-7, S1-8, S2-7, S2-8, S3-7, and S3-8.
3. Largest Ack: 6

   ACK Block Count: 2

   Packet Number Ranges: 1: 5-6, 2: 1-3

   Gaps: 1

Problem 3

In CSMA/CD, after the fifth collision, what is the probability that a node chooses K = 4? The result K = 4 corresponds to a delay of how many seconds on a 10 Mbps Ethernet?

$\frac{1}{32}$.
The delay is equal to $\frac{4*512bits}{10^7 \frac{bits}{second}} = 2.048 * 10^{-4} seconds$

Problem 4

Please answer the following questions regarding checksum.

1. Why is the IP header checksum recalculated at every router?
2. What is covered by IP checksum and TCP checksum?

1. The IP header checksum has to be recalculated at every router since the router may fragment the IP packet, and due to this fragmentation we have to make sure that we can use this checksum to verify the fragmented packet.
2. The IP checksum covers the content of the IP packet header, which results in safer network transmission, but is not completely foolproof due to the weaknesses of general sum checksums. The TCP checksum covers the actual TCP content of the packet. This is necessary as a packet may be transmitted correctly (meaning IP checksum requirements are fulfilled), but it may not have been reassembled correctly or may have been corrupted, meaning that the TCP checksum will fail, indicating this error.

Problem 5

Please answer the following questions about the QUIC protocol.

1. How did HTTP/2 solve the head-of-line blocking problem? What was the limitation to the approach?
2. Why is it hard for new transport protocols to be deployed? Why is QUIC easier to deploy compared to previous transport layer protocols such as SCTP?
3. Can you identify any potential issues with the QUIC protocol design?

1. HTTP/2 addressed the head-of-line blocking problem by creating multiple streams to allow concurrent data transmission. However the limitation lies in the fact that it still runs atop a TCP connection, meaning it still had to face the same issues with TCP segment recovery when a segment was lost.
2. It's hard to deploy new transport protocol since a new protocol would need to account for reliability and security amongst heavily crowded networks, not to mention it would have to go through a large bureaucratic process of vetting and approval and then be supported by all current systems, something that could take quite a while indeed. QUIC is easier to deploy because it's implemented in the user space and is then deployed atop UDP, a preexisting protocol with almost universal support. This allows it to out-perform protocols implemented atop TCP, whilst achieving multiplexing and similar reliability through application and transport layer logic. This then shifts the stress of reliability checking onto the less strained application-transport interface.
3. The QUIC protocol, as we mentioned earlier, is implemented in the user space. This allows for a great deal of flexibility and quick updates to the protocol. However, the way in which the QUIC protocol has been designed does not guarantee chronological delivery of packets. In the case of co-dependent requests being sent to a server, we would then be forced to wait to hear the ACK of the first request before sending our second dependent request, thereby making the parallelism afforded by QUIC streams redundant.