1. Are the objectives of flow control and congestion control the same? Why or why not?

No the objectives are not the same flow control is about managing the buffer at the receiving end and ensuring that frames/packets are not sent faster than a rate at which is acceptable by the destination host. Flow control scope is point to point involving a sending and a receiving router.

Protect receiver’s buffer from overflow

Congestion control is about ensuring that the subnet (network) does not become overloaded and packets are able to reach their destination with a low response time. Congestion control is a global issue it involves all the host routers in the network.

1. What is the advantage in using hop-by-hop choke packet over typical choke packet method as a solution to network congestion?

A hop-by-hop choke is sent by the receiver and will affect each hop it passes through reducing its transmission speed immediately. This is because choke packets over long distances is not very effective as it will only slow the transmission of a particular node after it has reached its sender and propagates back to the receiver. Using a hop-by-hop choke packet will reduce the effects of network congestion a lot faster than a typical choke packet.

Choke packet will only affect source where as hop-by-hop affects each hop it passes through. Hop-by-hop choke packet requires each hop to reduce its transmission even before the choke packet arrives at the source hence it is quicker in handling congestion.

Reactive method

1. Describe two (2) major differences between the bit warning method and the RED method.

The bit warning method of congestion control occurs at the sending node (router) after it has received the warning bit alongside the ACK whilst the RED method occurs at the receiving router after it has monitored the queue length of an average packet and deemed that it is too long.

Warning bit drops packet only when there is no buffer space left, RED drops packet before the buffer size is full.

Warning bit method explicitly (directly warns) sends congestion notification to the source by setting a bit in the packet header, RED implicitly (indirectly indicate) notifies the source by dropping its packets.

Proactive vs Reactive method of congestion control.

Warning Bit only works with TCP not UDP. RED works with both TCP and UDP.

1. Why was it difficult to detect congestion in old days?

That is because hardware and software was much more unreliable hence if a packet were to be dropped it may have been an error or the packet being lost rather than congestion hence it was difficult to detect congestion.

Networks not reliable in old days thus packet loss can occur due to transmission error or a packet discard due to congestion therefore it is hard to isolate the real cause. Nowdays packet loss due to congestion network is a lot more reliable. Assume that packet loss caused by congestion

1. Consider the effect of using slow start on a line with a 10-msec round-trip time and no congestion. The receive window is 24KB and the maximum segment size is 2KB. How long does it take before the first full window can be sent?

First send 1 segment -> 2 segments -> 4 segments -> 8 segments -> 16 segments (N/A) hence send 24KB

Sender knows space available in receiver’s buffer due to Window Size in TCP header

Bursts are 2KB, 4KB, 8KB, 16KB, 24KB

Receiver window < 32KB

4 \* 10ms = 40ms before first full window can be sent

1. Why is TCP called a byte-stream protocol? How does UDP differ from TCP in this regard? Which layer is responsible to segment the data if UDP is used in Transport layer? (Is it Transport layer itself or Application layer)? What protocol would you use to multicast or broadcast a message?

It is called a byte-stream protocol as applications send data to the transport layer and TCP segments data before sending to Network layer. It also sends data as a stream of bytes not sequence of messages. Every byte has a sequence number. UDP however is a message oriented protocol hence it does not segment the data which is handled by the application layer.

You would use user datagram protocol to broadcast or multicast a message this is because there are lower overheads hence this allows for faster retransmissions.

TCP sends data as a stream of bytes. Messages received out of order utilise sequence number to order the data.

TCP does not support multicast/broadcast. Connection establishment will require too many resources (memory/CPU).

1. DNS uses UDP instead of TCP. What is the main difference between UDP and TCP? If a DNS packet is lost, will there be automatic recovery? Will that cause a problem? Why or why not?

In UDP a connection is not established however in TCP a connection has to be established for the transfer of data hence it is reliable delivery of packets. UDP does not wait for ACK from receiver and is causes the unreliable delivery of packets.

TCP is connection oriented hence it facilitates reliable transfer of data has error correction, flow control, and acknowledgments, sequence numbers etc.

No if a DNS packet is lost there will not be an automatic recovery the sender will have to send another query to the DNS server if they want the information. No this will not cause a problem because if you do not receive a response after some time just resend the query.

Process makes DNS request it starts a timer. If timer expires it makes the request again and no harm is caused.