1) E-mail requires both sender and receiver mail servers to communicate directly. Further, all communications must be made between these two servers using SMTP.
   i)   Is it possible to allow intermediate servers as part of this communications? If so, what are the main advantages of doing so? If no, why is that infeasible, or what are the disadvantages of utilizing it if it was feasible?
   ii)   Considering webmail, are there any cases when SMTP between the two mail servers can be replaced by HTTP? Explain why, or why not.

2) Client-Server architecture can be used for file transfer; however it is assumed/said to be inferior in comparison to P2P for such operations. Considering a small number of interacting hosts that need to share files, is it true that client-server would perform badly? Explain clearly your answer. You must indicate why the number of interacting hosts/peers is significant in determining whether or not client-server is suitable for file transfer.

3) Show, through a detailed scenario/example, why GBN protocol would fail in case the sender's window size exceeds 2^k-1, where k is the number of frame bits for the sequence number.
   Considering that the window size is set to what it is expected to be. Is there any possibility that GBN could produce the same performance obtained by the Unrestricted Protocol? If no, explain clearly why this is infeasible. If so, explain clearly how, or under what conditions, GBN could produce the same performance of the Unrestricted Protocol.

4) Explain how UDP frames can end up TCP frames being delayed, theoretically, indefinitely! If so, what would you propose as changes to UDP to mitigate this problem? Your solution must mainly keep the advantages/purpose of UDP, while mitigating the problem at hand.

5) Show, through an example, how checksum could be inconclusive of error detection (i.e. does not guarantee that errors can be detected). In your example, assume transmitted data is broken into 24-bit chunks by the protocol utilizing checksum. In case errors are detected by checksum, does that fully (100%) guarantee that errors must have actually occurred?

6) Show through an example (sketch the scenario), how TCP uses the credit system to ensure that a sender would not overflow a receiver. In your scenario, assume that initial communication establishment resulted in sender (host A in general since it also behave as a receiver) starting with frame # 64, while the receiver (host B in general since it also behaves as sender) starts at frame # 192. Assume that the data size of all these exchanged frames is 100 bytes. Further, assume that at start, each of the hosts allows the other one a credit of 200. The rest of the scenario is up to you, however, you need to show at least one case when both of them coincidently set the credit to 0 at the same time.

7) Some may strongly argue that it is much better to use network-assisted congestion control instead of end-to-end congestion control. Do you support such argument? Clearly explain why or why not, and indicate the advantages of your choice over the other one. Why does TCP use end-to-end congestion control over the Internet?

8) With VC networks, the frame does not include the receiver address. Explain clearly how this would still work! Give a detailed scenario that explains how deliveries to the receiver can still be obtained. Show all needed tables at the routers in support to your scenario.

9) What is CIDR? What are the main advantages of using CIDR? Are there any major disadvantages to this technique?

10) With NAT routers, many machines over the Internet would have the same IP address! Explain how this would still work. Explain also how a P2P non-NAT host can establish a connection with another peer located under a NAT.