COMP 6461 Theory Assignment 2

Mrinal Rai– 40193024

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.

a. 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?

It is possible to allow intermediate servers as part of the mail server communication. The main advantages in doing so is that the mails would be stored at multiple servers. So, in case of failure of one server, mails ca still be retrieved from other servers. Also, different receivers can be connected to different mail servers in order to balance the load. But, in doing so security and confidentiality of the mails can be compromised, as mails are stored at more than on unreliable server. So, the chances of security breach increase with the increase in the number of the mail servers between the sender and the receiver.

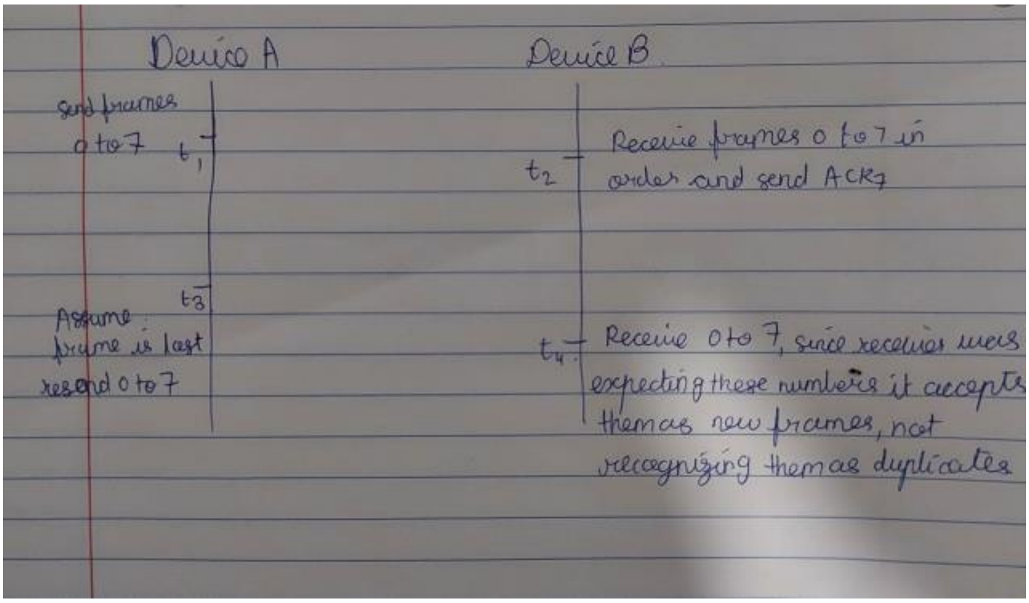
b. Considering webmail, are there any cases when SMTP between the two mail servers can be replaced by HTTP? Explain why, or why not.

No, the SMTP between the mail servers cannot be replaced with HTTP as the mail servers are designed to follow the mail format which is sued by the SMTP whereas HTTP does not use the format. HTTP is a pull protocol and SMTP is push protocol. In case we replace SMTP with HTTP then each mail server should be pulling data from each other which is not the functionality of the mail server.

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.

If the file transfer between multiple hosts is done using the Client-server architecture, then the server needs to upload n (number of hosts) copies of the file sequentially. Whereas if this is don using P2P architecture, server needs to upload the file only once and then any host can download that copy and therefore takes responsibility of sharing the file to one host. This is continued till all the hosts receive the file. Hence the complexity if the time taken to share the file case of Client-server architecture more than P2P architecture.

3. Show, through a detailed scenario/example, why GBN protocol would fail in case the sender's window size exceeds 2k-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.



send frames 0 to 7

Receive frames 0 to 7 in order and send ACK7

Assume frame is lost resend 0 to 7

Receive 0 to 7, since receiver was expecting these numbers it accepts them as new frames, not recognizing them as duplicates

In the above scenario the device B receives old data as new data and hence window size greater than 2k -1 does not work. In case where sender window never reaches its maximum size the GBN would produce the same performance as unrestricted protocol. This is the case when ACK's come back fast enough so the sender just advances its window and sends again.

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.

When TCP starts sending frames, it increases its window size additively. On the other hand, UDP keeps sending packets without packets without break. When the network gets congested, TCP would detect it and reduce the window size multiplicatively, but UDP would keep sending hence congesting the network and silencing TCP client. UDP can be changes to send messages at a constant pace with some timeout function where UDP does not send any packets in between allowing others to use the network. Packet elimination can be used at routers to block traffic from source which is sending packets continuously and at high pace.

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?

Consider the data to be sent is 0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0

Sender side:

Chunk 1: 0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0

Chunk 2: 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0

Sum: 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 1 0 0

Checksum: 0 1 0 0 1 0 1 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1

Receiver side: 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0

Chunk 1: 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0

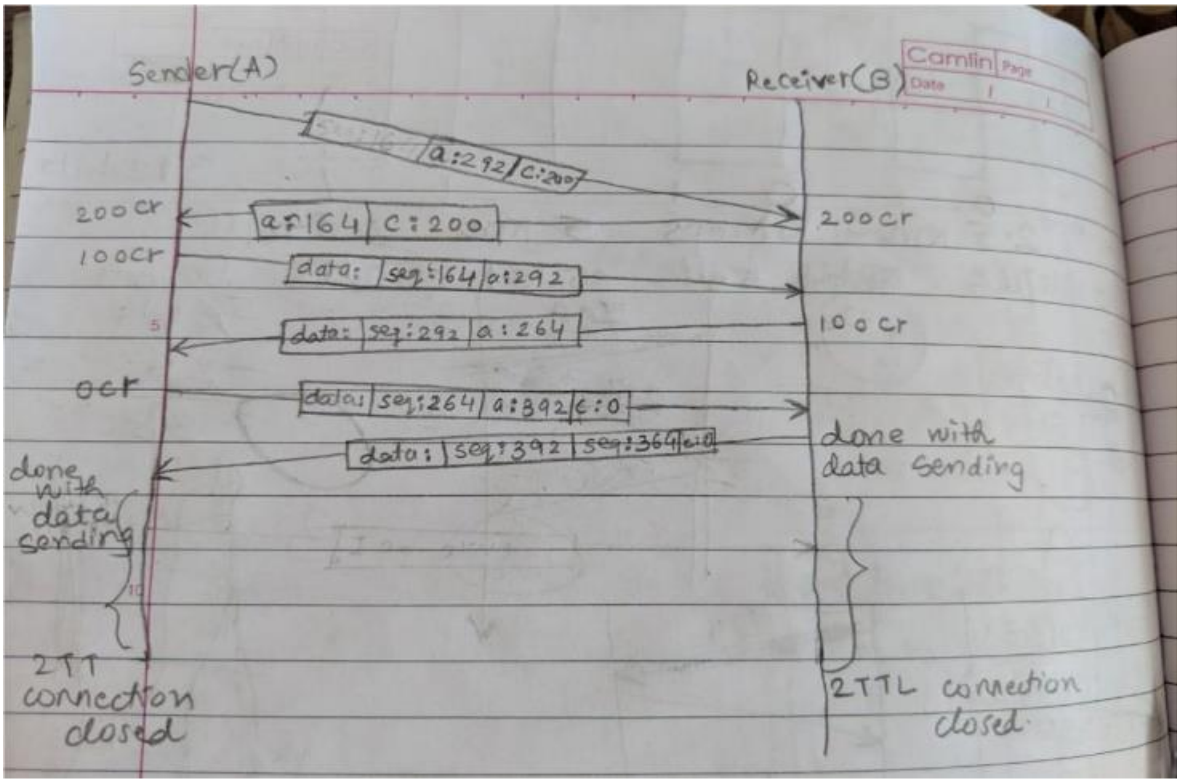
Chunk 2: 1 0 1 0 1 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0

Sum: 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 1 0 0

Checksum: 0 1 0 0 1 0 1 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1

The checksum matches with the one sent by the sender even though 2-bit errors are present. Hence, it is evident that checksum can be inconclusive of error detection. In case errors are detected by checksum it fully guarantees that error must have actually occurred because changes in checksum would occur only in two cases if the data is altered or the checksum itself gets corrupted.

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 behaves as a receiver) starting with frame # 194, while the receiver (host B in general since it also behaves as sender) starts at frame # 329. 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.

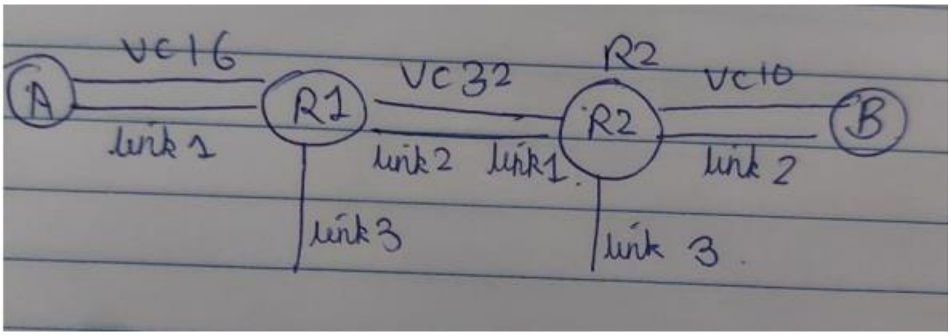


In this scenario the first two segments form A and B sends c=200 as they have full buffer capacity and in another scenario the sender sends data with c=100 and receiver also sends data with c=100 and in the end both sender and receiver send data with c=0 and thus keeps waiting infinitely for each other.

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?

Network assisted congestion control only works in the scenarios where the network is small enough to handle choke packets on the network. While in large networks the choke packets would congest the network even more. So NACC would only work in cases when the network is small. Internet is a big network and If TCP uses NACC over it, then routers would start sending choke packets on the network which would worsen the situation of a network like internet. Hence TCP uses end-end congestion control which detects the congestion itself and expects no help from the routers saving the network from further congestion.

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.



When a VC connection is setup, the receiver address is known to the sender. Hence, all the router tables would be set accordingly in the following manner:

Table for router 1:

Incoming Interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	16	2	32

Table router 2:

Incoming Interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	32	2	10

As shown in the above tables as the frame is handled using the VC numbers and link interface numbers, there is no need of B’s address in the frame.

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

CIDR is Classless Inter Domain Routing. With the use of usual classes if a network 1000 machines it has to ask for 4 class C address ranges which would in the end result in the company getting 4 different networks. So, when a host from 1 network has to send a packet to other host of the same company but in a different network, it will have to travel out of the network which would cost money to the company. This can resolve if the company asks for 4 consecutive address ranges and programs the router to read first 22 bits as the network address. This would help in merging the previously obtained 4 networks into 1. The disadvantage of this technique is that the company cannot take 5 consecutive class C for connecting 1200 machines as it would make the network address to be of 21 bits. Total number of class C resulting with 21-bit network address is 8. Taking them would mean that other 3 if used separately will also have the same network address which would violate the norm of having unique network address per network.

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.

NAT routers have a NAT table that keeps track of the requests (IP Address and port number) made from a private network. Before sending this request to the destination address, NAT router would change the source IP address to its own and attach a different port number than the original and adds this translation to the NAT table. When it receives another request from its own private network, it would attach the same IP address but a different port number to identify uniquely the responses from the destination. In this way, many machines over the internet can have same IP address. When a non-NAT host (A) wants to establish a connection to a host (B) behind the NAT, it can find a host C that is already connected to B and ask C to let B know that A wants to connect to B. In this way using reverse connection, a P2P non NAT host can establish a connection with another peer located under a NAT.