Student ID:1407870 Name: Muhan Guan

Q1.

A D E G

Slot1: A, D, E, G -collision

Slot2: A, D-collision

Slot3: A

Slot4: D

Slot5: E, G-collision

Slot6: E

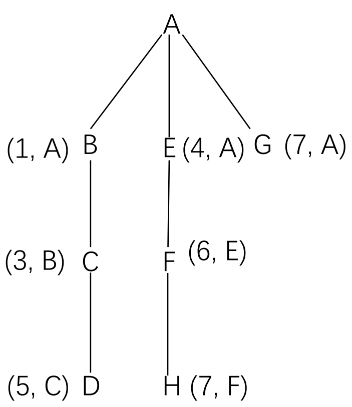
Slot7: G

Q2.

(1)

step	A	B	C	D	E	F	G	H	Nodes in the tree
1	0	∞	∞	∞	∞	∞	∞	∞	{A}
2	-	1	∞	7	4	∞	7	∞	{A, B}
3	-	-	3	7	4	∞	7	∞	{A, B, C}
4	-	-	-	5	4	∞	7	∞	{A, B, C, E}
5	-	-	-	5	-	6	7	8	{A, B, C, E, D}
6	-	-	-	-	-	6	7	8	{A, B, C, E, D, F}
7	-	-	-	-	-	-	7	7	{A, B, C, E, D, F, G}
8	-	-	-	-	-	-	-	-	{A, B, C, E, D, F, G, H}

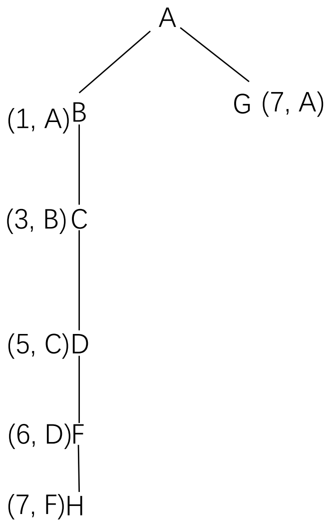
weights of Shortest path from A to the other nodes



(2) Given the condition of router E is offline, so we need to remove all flows that related to E (including E-A, E-H, and E-F), and calculate new routing table again.

step	A	B	C	D	F	G	H	Nodes in the tree
1	0	∞	∞	∞	∞	∞	∞	{A}
2	-	1		7	∞	7	∞	{A, B}
3	-	-	3	7	∞	7	∞	{A, B, C}
4	-	-	-	5	∞	7	∞	{A, B, C, D}
5	-	-	-	-	6	7	∞	{A, B, C, D, F}
6	-	-	-	-	-	7	7	{A, B, C, D, F, G}
7	-	-	-	-	-	-	-	{A, B, C, D, F, G, H}

new paths and their weights:



Q3.

(1) Simplicity: Both short requests and short replies may be sent as a single UDP packet in the simplest case. Not only is the code simple, but fewer messages are required (one in each direction) than with a protocol requiring an initial setup like TCP.

Faster operation: the connectionless feature of UDP can shorten communication time used by RPC, ,because no setup is needed in advance and no release is needed afterward, especially when only several parameters would be transmitted by RPC.

(2) In both cases, the client will resend request to the server, under the implementation of RPC, a reply serves as an implicit acknowledgment for a request, so the request need not be separately acknowledged, so the client would set a timer to retransmit request again and again if no reply received after timeout.

(3) When the operation performed by calling RPC is not idempotent, such as modifying database content or creating resources, etc., and these operations cannot be repeated multiple times, it may be necessary to set up a TCP connection and send the request over it rather than using UDP.

Q4.

(1) Video streaming applications are sensitive to jitter, if transmission time varies randomly between 1~2seconds, the result would be terrible, as the arrival times of audio and video frames could greatly vary, audio-video synchronization issues may occur, for example, the audio may arrive earlier, leading to the audio playing ahead of time.

Besides, the quality of live stream is significantly affected by bandwidth, if the required bandwidth of network exceeds its available capacity, then the video quality will degrade in the form of stutters, outright interruptions or reduction on video resolution, which would have negative impact on users’ viewing experience.

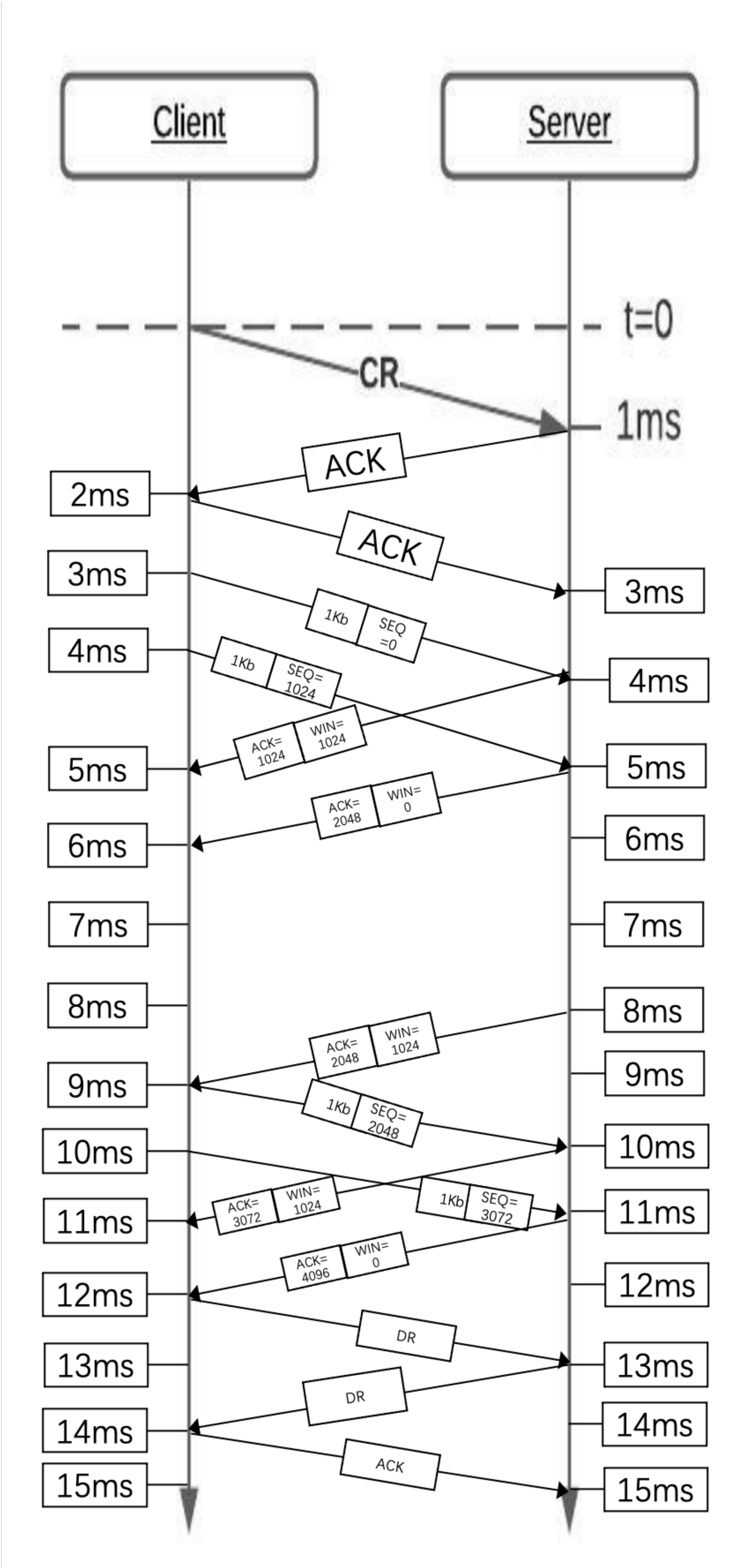
(2) One technique is to use buffering, buffering is a process where data packets are collected and stored (in a buffer) for a certain amount of time before they are processed or forwarded, such technique helps in reducing jitter because it allows time for out-of-order packets to arrive and be reordered, and for delayed packets to catch up. However, buffering introduces delay because data must be held for a period of time before

processing.

(3) No, because transit delays are unacceptable for Videoconferencing application. When transmitting a video conference, having a few pixels wrong is no problem, but having the image jerk along as the flow stops and starts to correct errors, or having to wait longer for a perfect video stream to arrive, is irritating, it would seriously impair the communication efficiency of real-time conference. Therefore, buffer is not an appropriate technique for videoconference as it could cause increased delay.

Q5.

1)


Client
Server
t=0
CR
1ms
ACK
2ms
ACK
3ms
3ms
1Kb
SEQ =0
4ms
1Kb
SEQ= 1024
4ms
ACK= 1024
WIN= 1024
5ms
5ms
ACK= 2048
WIN= 0
6ms
6ms
7ms
7ms
8ms
8ms
ACK= 2048
WIN= 1024
9ms
9ms
1Kb
SEQ= 2048
10ms
10ms
ACK= 3072
WIN= 1024
1Kb
SEQ= 3072
11ms
11ms
ACK= 4096
WIN= 0
12ms
12ms
DR
13ms
13ms
DR
14ms
14ms
ACK
15ms
15ms

2)
It would take 15ms in total.