COMP3331/9331 Sample Mid-session Exam Solutions

Question 1.

(a) OK: Connection-setup overhead, short-duration interaction. NOT OK: Header overhead.

(b) OK: Variable delays from reliability/congestion control in TCP, loss tolerant applications, NOT OK: connection setup overhead.

(c) NS record.

(d) The 3 packets sent to the 4^{th} hop appear to be routed along different paths. This is evident from the fact that the 4^{th} hop router is different for each of these 3 packets.

Question 2.

Circuit Switching:
Total delay = setup time + propagation delay + transmission delay
= 0.2 + 4 (0.001) + 3200/9600 = 0.5373 seconds

Datagram Packet Switching:
Total number of packets = 3200/800 = 4
The first packet will take 4 * (824/9600) + 4 (0.001) seconds to reach the destination.
The second packet will arrive (824/9600) seconds after the first one.
In this way, the last packet will arrive 3 * (824/9600) after the first packet.
Hence the total delay = 4 * (824/9600) + 4 (0.001) + 3 * (824/9600) = 0.60483 seconds.

Question 3.

Part 1

The sequence of operation is as follows: A sends the HTTP GET request directly to the Web Cache, The Web Cache sends a type A DNS query for the server S to its local name server, N. N sends out the DNS query to the DNS root server. A sequence of DNS query response messages is executed (depending on whether recursive or iterative queries or a combination is employed) and local name server, N finally receives the IP address for S. N passes on this IP address to C. C then initiates a TCP connection with S by sending a SYN message. Once the TCP connection is established (using the 3-way TCP handshake), the GET request for the index.html page is sent to S by C. S replies back to C with the requested page. C then passes on this page to A.

Consequently, the table overleaf can be filled with the order of events as shown.

ID	Source	Destination	Source Port	Destination Port	Protocol	Contents	Order
1	C	DNS root server		DNS (53)	UDP	DNS type A query for S	X
2	A	C		Web Cache	TCP	GET http://S/index.html	1
3	N	DNS root server		DNS (53)	UDP	DNS type A query for S	2
4	C	S		HTTP (80)	TCP	SYN	4
5	C	S		HTTP (80)	TCP	GET index.html	5
6	S	A	HTTP (80)		TCP	index.html	X
7	C	A	Web cache		TCP	index.html	7
8	N	C	DNS (53)		UDP	IP address for S	3
9	S	C	HTTP (80)		TCP	index.html	6
10	N	A	DNS (53)		UDP	IP address for S	X
11	A	S		HTTP (80)	TCP	GET index.html	X

Part 2

Note that the Web cache will cache the index.html page in the previous sequence of operation (in Part 1). When the user on machine A types in the same URL in the browser, the corresponding HTTP GET request will be directed to the Web cache C (as before). In this case, C can reply back directly with the requested page, since it has cached it. As a result the following events will be eliminated (events indicated by their corresponding IDs from the table above): 3,4 5, 8 and 9.

Question 4.

Go-back N: 15, 15, 15, 16

Selective repeat: 15, 18, 17, 16

Question 5.

(a) To begin with, the sender is at the ”wait for ACK0” state and the receiver is at the ”wait for 0 from below” state. Since the packet is lost, the receiver doesn’t receive any packet and takes no action. When timeout occurs at the sender, it resends the packet and starts the timer. When the receiver receives this packet, it finds that the packet is not corrupted and has the correct sequence number, so it extract the data and deliver it to the upper layer. The receiver also makes an ACK packet with sequence

number 0 to be delivered to the sender. The receiver then moves to the state ”Wait for 1 from below.” When the sender receives the ACK(0) packet, it advances to the ”Wait for 1 from above” state.

(b) The sender will be in the ”wait for ACK 1” state and the receiver will be in the ”wait for 1 from below” state after the sender has send the second packet the first time. When the receiver receives a corrupted packet, it sends an ACK packet with sequence number 0 to the sender. When the sender receives this ACK(0) packet, it find that it is not corrupted and has sequence number 0, it does nothing. The sender remains in the ”wait for ACK 1” state until timeout has occurred. When the sender times out, and the sender resends the packet with sequence number 1. When the receiver receives the packet, it checks that it is not corrupted and finds that it has the correct sequence number, so it extracts the data and delivers it to the upper layer. The receiver also makes an ACK packet with sequence number 1 to be delivered to the sender. The receiver then advances to the state ”Wait for 0 from below.” When the sender receives the ACK(1) packet, it advances to the ”Wait for 0 from above state” and stops the timer.

(c) Sequence number will not be needed in this case because no packets will be duplicated. To see this, there are only two possible scenarios here:

Scenario 1: the packet is delivered and ACK is received correctly.

Scenario 2: The data packet is lost and when the sender times out, the data packet is re-sent.

Note that under these scenarios, the receiver will never receive a duplicate packet, therefore, sequence number is not necessary.