

**SEMESTER 1**

**Autumn 2021 Examination**

# CS320

**Computer Networks**

Dr M. Huggard, Dr J. Timoney, Dr. B. Hennelly

Time allowed: 2 hours

**Instructions**

|  |  |  |
| --- | --- | --- |
|  | **Yes** | **No** |
| Log Books Allowed |  | X |
| Formula Tables Allowed |  | X |
| Other Allowed (*enter details)* |  | X |

Time allowed: 2 hours

You must answer Question 1 and any three other questions.

Your mark will be based on Question 1 and your best three answers from the remaining questions

**All questions** carry equal marks

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** |  | **SHORT QUESTIONS**   1. Suppose you would like to urgently deliver 40 terabytes data from Dublin to London (Tera is 10 to the 12th power). You have available a 100 Mbps dedicated link for data transfer. Would you prefer to transmit the data via this link or instead use FedEx overnight? 40 terabytes = 40 \* 1012 \* 8 bits. So, if using the dedicated link, it will take 40 \* 1012 \* 8 / (100 \*106 ) =3200000 seconds = 37 days. But with FedEx overnight delivery, you can guarantee the data arrives in one day, and it should cost less than €100. 2. Suppose Bob joins a BitTorrent torrent, but he does not want to upload any data to any other peers (so called free-riding). Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is Bob’s claim possible? Why or why not? Yes. His first claim is possible, as long as there are enough peers staying in the swarm for a long enough time. Bob can always receive data through optimistic unchoking by other peers. 3. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local DNS server, so a DNS lookup is necessary to obtain the IP address. Suppose that *n* DNS servers (non-including your local DNS server) are visited before your host receives the IP address from DNS. Assuming the round-trip time from any DNS server to any other is RTT, what is the difference in time between a recursive and iterative DNS query. Iterative – local server must contact n DNS servers requiring n\*RTT. A recursive query is also n\*RTT 4. Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario? Answer: Application layer protocols: DNS and HTTP, Transport layer protocols: UDP for DNS; TCP for HTTP 5. Let *T* denote the time interval that a TCP connection takes to increase its congestion window size from *W/2* to *W*, where *W* is the maximum congestion window size. What is the average throughput of the TCP connection? Answer 0.75W/T 6. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is *m*, then the sequence number for the subsequent segment will necessarily be *m* + 1. Is this true or False – explain your answer. Answer, it could be true if the payload is one byte – the sequence numbers increases as a function of datasize. 7. What is meant by the “match plus action” operation of a router or switch? In the case of destination-based forwarding packet switch, what is matched and what is the action taken? In the case of an SDN, name three fields that can be matched, and three actions that can be taken. “Match plus action” means that a router or a switch tries to find a match between some of the header values of a packet with some entry in a flow table, and then based on that match, the router decides to which interface(s) the packet will be forwarded and even some more operations on the packet. In the case of destination-based forwarding packet switch, a router only tries to find a match between a flow table entry with the destination IP address of an arriving packet, and the action is to decide to which interface(s) the packet will be forwarded. In the case of an SDN, there are many fields can be matched, for example, IP source address, TCP source port, and source MAC address; there are also many actions can be taken, for example, forwarding, dropping, and modifying a field value. 8. It has been said that when IPv6 tunnels through IPv4 routers, IPv6 treats the IPv4 tunnels as link-layer protocols. Do you agree with this statement? Why or why not? Yes because the Ipv6 datagrams are encapsulated inside the IPv4 ones 9. Suppose two nodes start to transmit at the same time a packet of length *L* over a broadcast channel of rate *R*. Denote the propagation delay between the two nodes as *d*prop. Will there be a collision if *d*prop *< L*/*R*? Why or why not? There will be a collision in the sense that while a node is transmitting it will start to receive a packet from the other node. 10. In CSMA/CD, after the fifth collision, what is the probability that a node chooses *K* = 4? The result *K* = 4 corresponds to a delay of how many seconds on a 10 Mbps Ethernet? *After the 5th collision, the adapter chooses from {0, 1, 2,…, 31}. The probability that it chooses 4 is 1/32. It waits 204.8 microseconds.* | **[25 marks]**  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks]  [2.5 marks] |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.** |  | **QUESTION 2 – HTTP CACHE vs P2P** Consider the scenario in the figure below in which a server in LAN B is connected to a router by a 1 Gbps link, with a 1 ms propagation delay. That router in turn is connected to a second routers over a 1.54 Mbps link with a 200ms propagation delay. This second router is an access router for LAN A, on which there is a single local web server and *N* hosts, all connected by 1 Gbps links with negligible propagation delay. All *N* hosts want to download a file of size *F*, only one copy of which exists on the origin server on LAN B.       1. Assume the web-cache is empty at the start, what is the total length of time (in terms of *N* and *F*) taken to download the file by all *N* hosts. You can neglect the all DNS messages in your calculation as well as all TCP segments used to establish the connection.   Answer: The file is first downloaded by the cache which requires the following time:  Transmission delay link 1 + Propagation delay link 1  + Transmission delay link 2 + Propagation delay link 2  + Transmission delay link 3 + Propagation delay link 3  = **F/1x109 + 1x10-3**  **+ F/1.54x106 + 200x10-3**  **+F/1x109 + 0**  ~~This would also require a single HTTP request which take~~ **~~+200ms+1ms~~**  Following this, each of the other N-1 hosts will download a copy from the web cache. The bottleneck link is the 1Gbps link to the caches which will be evenly distributed to the N-1 hosts providing a capacity of 1Gbps/(N-1) to each download. Therefore, the time for each host to download in parallel is given by  **+ (N-1)F/1x109 + 0**  TOTAL TME = is the sum of all items in bold above   1. Now find the minimum length of time (in terms of *N* and *F*) taken to download the file by all *N* hosts, where this time the web cache is turned off, and the *N* hosts on LAN A and the origin server on LAN B use a peer-to-peer architecture to distribute the file. You can assume that each of the *N* hosts can use 50% of their 1Gbps links for download and 50% for upload. *You can neglect any messages sent to initiate and control the P2P distribution.*   Answer  DP2P= max{F/us,,F/di,,NF/(us + Sum{ui})}  DP2P= max{F/1Gbps,,F/0.5Gbps,,NF/(1Gbs + N0.5Gbs)}  DP2P= max{F/0.5Gbps,NF/(1Gbs + N0.5Gbs)}   1. Taking N = 100, and F = 1 Gigabytes, which of the two scenarios takes the shortest time.   Answer: F=1\*8\*1e9;N=100;TMEa = 0.402 + F\*(N/1.54e6 + 2/1e9)TMEb = 0.402 + F\*(1/1.54e6 + (2+N)/1e9)TMEa = 5.1950e+05TMEb = 6.0112e+03 fastest | **[25 marks]**  [10 marks]  [10 marks]  [5 marks] |

|  |  |  |  |
| --- | --- | --- | --- |
| **3.** |  | **QUESTION 3 – RELIABLE DATA TRANSFER** **A** is an automated teller machine (ATM) that communicates with a bank server, **B** in order to debit a customer account as required. **A** is invoked to send a message to **B** whenever there is a request from above to debit an **account** by a particular **value** using the command **debit(account, value).** When **A** receives this call from above, it sends **B** a **request\_money packet** for the **account**. **B** replies with an acknowledgement that includes a positive or negative message depending on whether there us sufficient funds in the account, i.e. only if **B** calculates that the **balance** in the **account** is greater than the requested **value** (i.e. there are sufficient funds in the account) it will proceed to debit the account. If there are insufficient funds, **B** will not debit the account. **A** cannot process a second **debit(account, value)** request while the first one has not been resolved. Design a reliable data transfer protocol between **A** and **B**, operating over an unreliable channel that can *lose packets but not corrupt them*, that will guarantee that the customer can never withdraw more money than is in their account.   1. You are required to give the finite state machine diagram at **A**. Clearly label all instructions in the diagram. 2. Also describe the finite state machine diagram at **B**, clearly labeling all instructions in the diagram.   **debit(account, value)**  **packet = request\_money(account,value, 0)**  **udt\_send(packet)**  **Start\_timer**  **Receive ANY other packet from above or from B**  **Λ**  **Receive ANY data from B**  **Λ**  **Timeout**  **udt\_send(packet)**  **Start\_timer**    **Receive Ack(0)**  **Stop\_timer**  **If Ack(0)==positive**  **send\_above(‘pay customer’)**  **else send\_above(‘insufficient funds’)**  **The same two states are mirrored on this side of the dashed line – replace 0 with 1 everywhere**  **Receive packet debit(account, value, 0)**  **If balance > amount: balance-=value: Ack=Ack(0,positive)**  **Else: Ack=Ack(0,negative)**  **udt\_send(Ack)**  **Receive ANY other packet from A B**  **Λ**  **The same state is mirrored on this side of the dashed line – replace 0 with 1 everywhere** | **[25 marks]**  [15 marks]  [10 marks] |

|  |  |  |  |
| --- | --- | --- | --- |
| **4.** |  | **QUESTION 4 – TCP**  Consider that only a single *TCP (Reno)* connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver’s receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion avoidance phase, that is, ignore slow start.   1. What is the maximum window size (in segments) that this TCP connection can achieve? 2. What is the average window size (in segments) and average throughput (in bps) of this TCP connection? 3. How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?   Answer   1. Let W denote the max window size measured in segments. Then, W\*MSS/RTT = 10Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity. Thus, we have W\*1500\*8/0.15=10\*10^6, then W is about 125 segments. 2. As congestion window size varies from W/2 to W, then the average window size is 0.75W=94 (ceiling of 93.75) segments. Average throughput is 94\*1500\*8/0.15 =7.52Mbps. 3. When there is a packet loss, W becomes W/2, i.e., 125/2=62.   (125 - 62) \*0.15 = 9.45 seconds, as the number of RTTs (that this TCP connections needs in order to increase its window size from 62 to 125) is 63. Recall the window size increases by one in each RTT. | **[25 marks]**  [9 marks]  [8 marks]  [8 marks] |
| **5** |  | **QUESTION 5 – LINK STATE AND DISTANCE VECTOR ROUTING**  Consider the network shown in the figure below  Index of /~michael/kr1999/4-network   1. Show the operation of Dijkstra’s (Link State) algorithm for computing the least cost path from **A** to all destinations and based on this result define the least cost from **F** to **A**   ANSWER    From F to A go backwards: F E D C B A   1. Consider only the part of the network made up by the nodes **H**, **G**, and **B** and ignore all other nodes. Show the first two iterations of the distance-vector algorithm for this three-node network based on the Bellman-Ford Equation.   Answer: | **[25 marks]**  [15 marks]  [10 marks] |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **6.** |  | **LINK LAYER**  Consider three LANs interconnected by two routers, as shown in the figure below.     1. Provide MAC addresses and IP addresses for the interfaces at Host **A**, both routers, and Host **F**. Suppose Host **A** sends a datagram to Host **F**. Give the source and destination MAC addresses in the frame encapsulating this IP datagram as the frame is transmitted ***(i)***from **A** to the left router, ***(ii)***from the left router to the right router, ***(iii)***from the right router to **F**. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time. 2. Suppose now that the leftmost router in the network is replaced by a switch. Hosts **A, B, C**, and **D** and the right router are all star-connected into this switch. Give the source and destination MAC addresses in the frame encapsulating this IP datagram as the frame is transmitted ***(i)***from **A** to the switch, ***(ii)***from the switch to the right router, ***(iii)***from the right router to **F**. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time. | **[25marks]**  [12 marks]  [13 marks] |
|  |  | Answer (a) |  |
|  |  | i) from A to left router: Source MAC address: 00-00-00-00-00-00  Destination MAC address: 22-22-22-22-22-22  Source IP: 111.111.111.001  Destination IP: 133.333.333.003  ii) from the left router to the right router: Source MAC address: 33-33-33-33-33-33  Destination MAC address: 55-55-55-55-55-55  Source IP: 111.111.111.001  Destination IP: 133.333.333.003  iii) from the right router to F: Source MAC address: 88-88-88-88-88-88  Destination MAC address: 99-99-99-99-99-99  Source IP: 111.111.111.001  Destination IP: 133.333.333.003  **Answer (b)**  i) from A to switch: Source MAC address: 00-00-00-00-00-00  Destination MAC address: 55-55-55-55-55-55  Source IP: 111.111.111.001  Destination IP: 133.333.333.003  ii) from switch to right router: Source MAC address: 00-00-00-00-00-00  Destination MAC address: 55-55-55-55-55-55  Source IP: 111.111.111.001  Destination IP: 133.333.333.003  iii) from right router to F: Source MAC address: 88-88-88-88-88-88  Destination MAC address: 99-99-99-99-99-99  Source IP: 111.111.111.001  Destination IP: 133.333.333.003 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |