

# Midterm Exam

## Computer Networks Fall 2022

---

### Part I (10 points)

#### A. True/False Questions (1 point each)

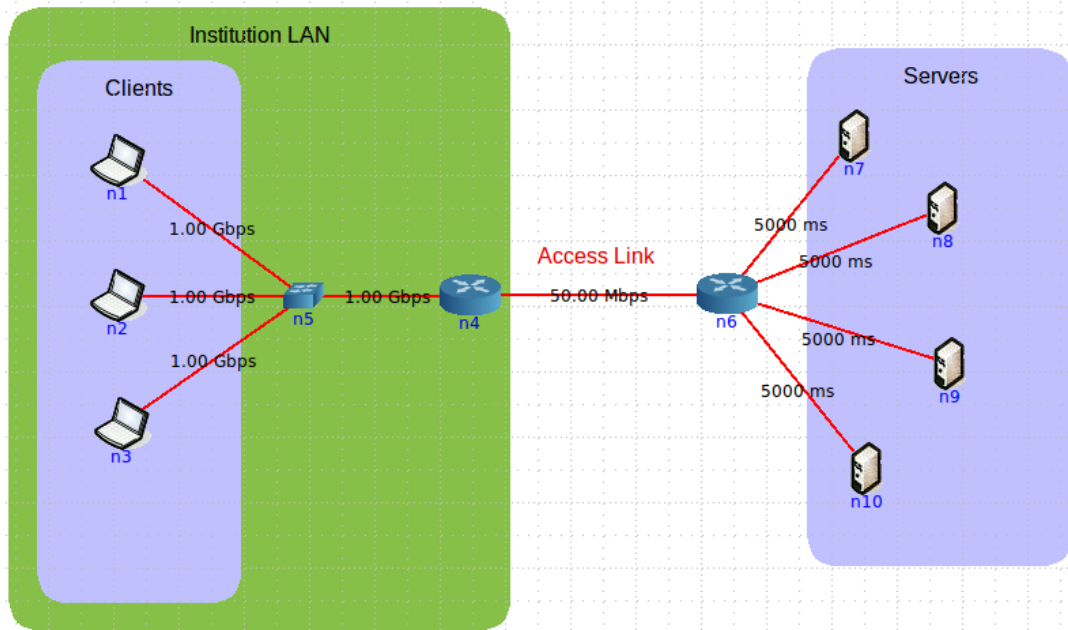
1	F
2	F
3	F
4	T
5	F
6	F
7	F
8	F
9	T
10	F

1.    T       F       If the file index.php in this request does not exist, the HTTP server will return code 400: GET /kurose\_ross/interactive/index.php HTTP/1.1.
  - a. False
    - i. We can't assume every http server strictly follows http semantics.
    - ii. If we do, the standard response for page not found is 404.
2.    T       **F**       UDP drops messages received out of order.
  - a. False
    - i. Doesn't care
3.    T       F       Internet is a network of networks that is managed centrally.
  - a. False
4.    T       F       Packet-switching is less reliable than circuit-switching but circuit-switching provides less efficient use of communication resources.
  - a. True

5.     T     F     FTP applications are loss-tolerant.  
      a. False
6.     T     F     A UDP socket is identified by source address and destination  
      address only.  
      a. False  
      i. Need ports
7.     T     F     A stop-and-wait transport protocol is neither reliable nor  
      efficient.  
      a. False  
      i. Reliability is guaranteed in stop-and-wait transport protocol
8.     T     F     If one side of a TCP connection advertises a RWND (receiver  
      window size) of 64K, that means it allocated 64K bytes of space for its Recv-Q.  
      a. False  
      i. RWND is not Recv-Q
9.     T     F     Timeouts occur in TCP when a duplicate ack is received.  
      a. True
10.    T     F     There are 13 physical DNS root name “servers” in the world  
      a. False

## Part II (5 points)

1. (1.5 points) Suppose an institutional network on a 1Gbps LAN is connected to the Internet using a 50Mbps link. Suppose clients at the institutional network make about 560000 requests per hour with an average response size of 100Kb. Suppose that the size of the HTTP request is negligible and that the Internet delay is 2500 milliseconds.



- What is the traffic intensity on the LAN (per second)?
  - $560000(\text{req/h}) \times 100(\text{Kb/req}) / 3600(\text{s/h}) = 15.55 \text{ Mbps}$
  - This value is smaller than Access Link so should be valid
  - We are ignoring the retransmit, header size/Etc. in this case.
  - $\text{Intensity} = 15.55 / 1000 = 0.016$
- What is the traffic intensity on the access link (per second)?
  - The same as LAN, 15.55 Mbps
  - $\text{Intensity} = 15.55 / 50 = 0.311$
- Would it be useful to place a web cache in this case (justify your answer)? If yes, then calculate traffic intensity on the access link if the hit ratio is 90%.
  - Yes, I certainly will.
  - Although the request rate is not capped by Access Link, the Internet delay is very high. Hitting the web cache will significantly reduce the delay from seconds to milliseconds and improve the overall user experience (improves the performance of some single threaded network applications as well).

- iii. Only the cache miss traffic (10%) will need to go through Access Link

1.  $15.55 \text{ Mbps} * 10\% = 1.55 \text{ Mbps}$
2.  $\text{Intensity} = 1.55/50 = 0.0311$

2. (1 point) Calculate the UDP checksum over the following binary:

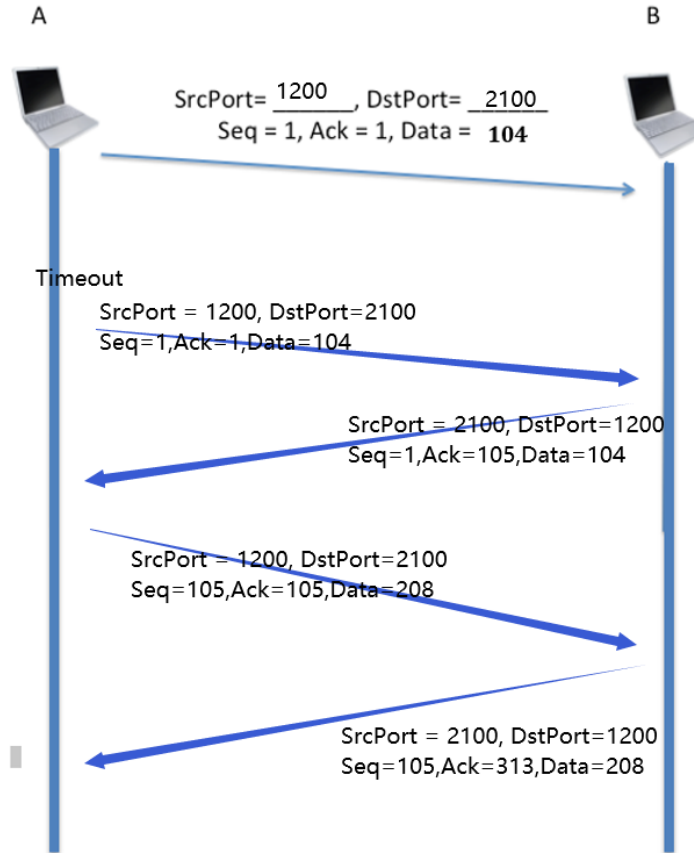
10111000101000110001000110101010100000010111111001010010100  
01000.

```
1011100010100011
+
0001000110101010
=
1100101001001101
+
1000000101111110
=
10100101111001011
wrap around
=
0100101111001100
+
0101001010001000
=
1001111001010100
```

3. (1 points) Assuming  $\alpha = 0.125$ , calculate the estimated round trip time over the three instantaneous round-trip times from ping below:
- 64 bytes from xxx.xxx.xxx.xxx: icmp\_seq=0 ttl=252 time=132.339 ms
    - o EstimatedRTT = 132.339ms
  - 64 bytes from xxx.xxx.xxx.xxx: icmp\_seq=1 ttl=252 time=22.121 ms
    - o EstimatedRTT =  $132.399 * 0.875 + 22.121 * 0.125 = 118.614 \text{ ms}$
  - 64 bytes from xxx.xxx.xxx.xxx: icmp\_seq=2 ttl=252 time=99.982 ms
    - o EstimatedRTT =  $118.614 * 0.875 + 99.982 * 0.125 = 116.285 \text{ ms}$
  - Final estimated RTT is 116.285 ms
4. (1.5 points) Assuming an application at client **A** initiated a connection from port 1200 to a server at **B** on port 2100 using a TCP socket. Whatever the application at client **A** sends to the server at **B**, the server echoes back in uppercase. Anything that the client gets from the server, it duplicates and sends it out to the server in lowercase. For example, if the client sends "hi", the server replies "HI". The client then duplicates that and sends back to the server as "hihi". The server

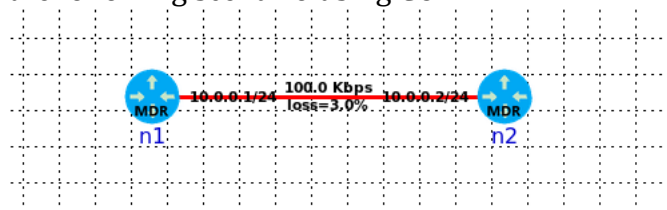
in return replies with “HIHI”, and so on. Sketch the exchange between the client and the server indicating the source port, destination port, sequence number, ACK sequence number, and data size of each exchange.

- The initial message sent by the client is 104bytes (as shown below) and that A and B are connected by an Ethernet cable.
- The first segment was lost
- Do this for two messages that A sends.



### Part III: CORE (5 points)

1) Create the following scenario using CORE



a.

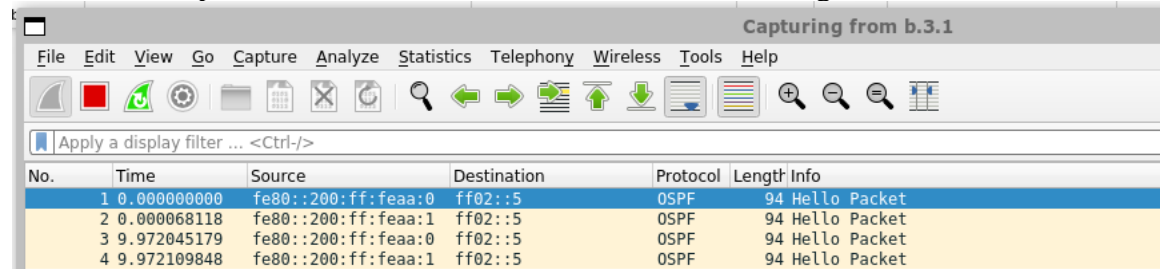
2) Use the python server script provided to start a server on node n2

a. Show the command you used

```
root@n2:/tmp/pycore,1/n2.conf# python3 /home/ubuntu/midterm/py_server_2.py
```

b.

3) Run Wireshark to Capture traffic on node n1 on the link connecting it to n2.



Wireshark interface showing a packet capture on interface b.3.1. The packet list contains four OSPF Hello packets, all with length 94 bytes. The packet details pane shows the structure of an OSPF Hello packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::200:ff:feaa:0	ff02::5	OSPF	94	Hello Packet
2	0.000068118	fe80::200:ff:feaa:1	ff02::5	OSPF	94	Hello Packet
3	9.972045179	fe80::200:ff:feaa:0	ff02::5	OSPF	94	Hello Packet
4	9.972109848	fe80::200:ff:feaa:1	ff02::5	OSPF	94	Hello Packet

a.

4) Use the python client script provided to start a client on node n1

a. Show the command you used

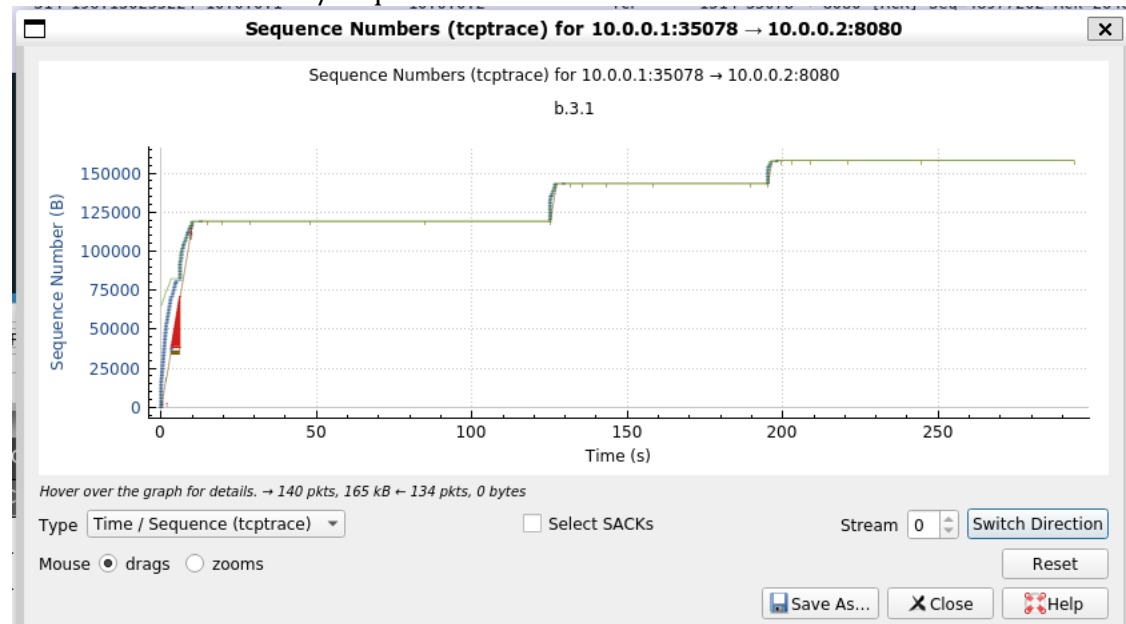
```
root@n1:/tmp/pycore,1/n1.conf# python3 /home/ubuntu/midterm/py_client_2.py
```

b.

5) Wait till client stops

6) Stop the Wireshark capture

7) Show a screenshot of the TCP/Sequence TCP Trace from Wireshark



a.

8) What part of the graph is ACK'd data?

a. The part under Yellow line is the ACK'd data

9) What part of the graph is in flight data?

a. The part between blue segment and yellow line is in flight data.

10) What part of the graph is Receiver Advertised Window?

a. The Green line shows the Receiver Advertised Window

11) Explain what is happening in the TCP connection by correlating ACK graph, in-flight graph, and Receiver Window Graph

a. The receiver window is full in most of the time, causing the transmission to stall. This behavior suggests that the receiver's processing speed is not fast enough.

- b. Validate your findings by investigating the code.
  - i. Yes, in line 20 there's a sleep 5 second code for receiving every 1024 byte of data.

12) Include a copy of the pcap in your submission

