CMPEN362 — Midterm Exam, Fall 2025

Name: (all capital letters) solution
Student email: ______________________________
Section: (circle one) Section 1 Section 2

P1	/10
P2	/10
P3	/10
Total	/30

Problem 1

Check all the correct answers. [1 pt per question]

1. From a service view, the Internet is:
   ☐ a network of interconnected ISP networks
   ☐ communication links and packet switches
   ✓ an infrastructure that provides communication services to applications
   ✓ an infrastructure that provides programming interface to application developers

2. The two key network-core functions are:
   ✓ determine the source-destination route
   ☐ guarantee delivery along the route
   ✓ forward packets from a router's input port to an output port according to the route
   ☐ ensure minimum throughput along the route

3. The four sources of packet delay at a single hop include: queueing delay, propagation delay,
   ✓ transmission delay ☐ pipelining delay
   ✓ processing delay ☐ decryption and re-encryption delay

4. What can NOT cause packet loss:
   ☐ buffer overflow at packet switch ☐ failure to pass checksum
   ✓ undetected bit error ☐ malicious attack

5. The three tiers of DNS servers are: root DNS server,
   ☐ Tier-1 DNS servers ✓ authorative DNS servers
   ✓ TLD DNS servers ☐ local DNS servers

6. The communication endpoints that send/receive messages over the Internet are
   ☐ users ✓ processes ☐ hosts ☐ sockets

7. The header fields identifying a TCP socket are
   ✓ source IP address ✓ source port number
   ✓ destination IP address ✓ destination port number

8. The reasons for preferring TCP over UDP include:
   ☐ smaller header ✓ lossless data transfer
   ☐ smaller delay ✓ in-order delivery

9. A process running a P2P application is
   ☐ always a client process
   ✓ a client process when initiating communication
   ☐ always a server process
   ✓ a server process when waiting to be contacted

10. Suppose host A sends a TCP segment to host B with sequence number 212, acknowledgement number 70, and a payload of 20 bytes. The return segment from B to A will have acknowledgement number
    ☐ 212 ✓ 232 ☐ 70 ☐ 282

Problem 2

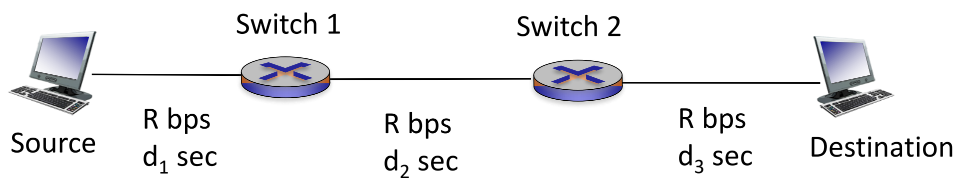

Figure 1: Problem 2.a illustration.

a) Consider a source-destination pair connected by 2 packet switches via 3 links as in Fig. 1. Suppose link i ($i = 1, 2, 3$) has a bandwidth of R bps and a propagation delay of d_i seconds. Ignore queuing and processing delays. Suppose that an M-bit message is segmented into P packets of equal length.

(i) How long does it take for the first packet to arrive at the destination? [1 pt]

$$\sum_{i=1}^{3}\left(\frac{M}{PR} + d_i\right) = \frac{3M}{PR} + \sum_{i=1}^{3} d_i$$

(ii) How long does it take for all the packets to arrive at the destination? [1 pt]

$$\frac{3M}{PR} + \sum_{i=1}^{3} d_i + (P-1)\frac{M}{PR} = \frac{(P+2)M}{PR} + \sum_{i=1}^{3} d_i$$

(iii) How long does it take for all the packets to arrive at the destination if each packet has an H-bit header? [1 pt]

$$\frac{3(M/P+H)}{R} + \sum_{i=1}^{3} d_i + \frac{(P-1)(M/P+H)}{R} = \frac{(P+2)(M/P+H)}{R} + \sum_{i=1}^{3} d_i$$

(iv) What is the minimum message size M such that the last packet does not depart the source before the first packet arrives at the first switch (including headers)? [2 pt]

Last packet departs the source at $P \cdot \frac{M/P+H}{R}$

First packet arrives at Switch 1 at $\frac{M/P+H}{R} + d_1$

Minimum M is given by: $P \cdot \frac{M/P+H}{R} \geq \frac{M/P+H}{R} + d_1$ [1 pt]

$\implies M \geq \frac{d_1 PR}{P-1} - PH$ [1 pt]

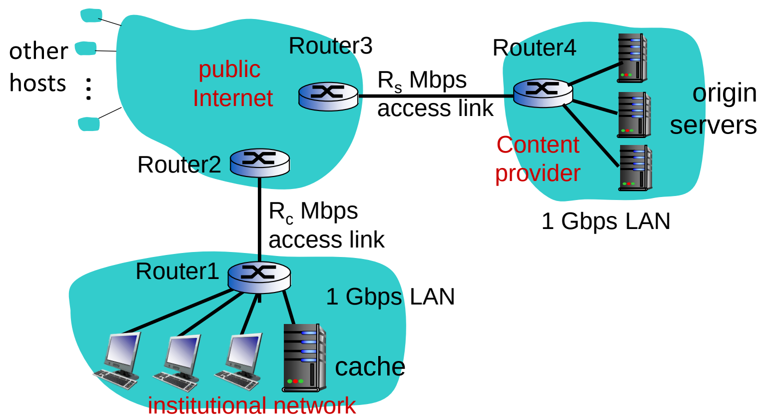

Figure 2: Problem 2.b illustration.

b) Consider serving web objects from origin servers to hosts as in Fig. 2. Suppose that each object is 1M bits, hosts in the institutional network generate 10 requests/sec, and other hosts generate 100 requests/sec. The average delay to send a request from Router2 to Router3 and obtain response is 1 second. Ignore propagation delays for access and LAN links. Let $R_c = 11$ Mbps and $R_s = 120$ Mbps. Answer to precision 0.001.

(i) What is the delay for a host in the institutional network to obtain one object from origin server? [2 pt]

Total delay = server LAN delay + server access delay + Internet delay + client access delay + client LAN delay [0.5 pt]
server/client LAN delay = 1M bits/1 Gbps = 1 ms
server access delay = $\frac{1}{120-100-10} = 0.1$ sec [0.5 pt]
client access delay = $\frac{1}{11-10} = 1$ sec [0.5 pt]
$\Longrightarrow$ total delay $= 0.001 + 0.1 + 1 + 1 + 0.001 = 2.102$ sec [0.5 pt]

(ii) Now deploy a web cache in the institutional network with hit rate 0.5. What is the average object downloading delay for hosts in the institutional network? [3 pt]

Hit delay = LAN delay = 1 ms [0.5 pt]
Miss delay = server LAN delay + new server access delay + Internet delay + new client access delay + client LAN delay [0.5 pt]
new server access delay = $\frac{1}{120-100-5} \approx 0.067$ sec [0.5 pt]
new client access delay = $\frac{1}{11-5} \approx 0.167$ sec [0.5 pt]
miss delay ≈ 1.236 sec (also correct: 1.235 sec) [0.5 pt]
$\Longrightarrow$ average delay $= 0.5\cdot$ hit_delay $+ 0.5\cdot$ miss_delay ≈ 0.619 sec (also correct: 0.618 sec) [0.5 pt]

Problem 3

Host S wants to send segments *alternatingly* to Hosts A and B. Each of A and B is connected to S via a *dedicated* channel that can lose/corrupt (but not reorder) segments. Design a stop-and-wait transport protocol to make sure that the receivers receive segments in the order of: A, B, A, B...

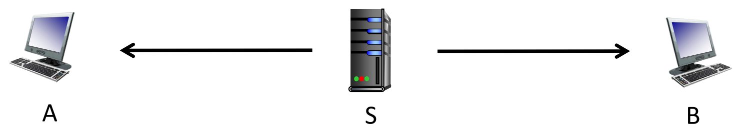

Figure 3: Problem 3 illustration.

You can use the following procedure calls:
– `rdt_send(data)`: called by upper layer to send data in 'data';
– `rdt_rcv(rcvpkt)`: called by lower layer after receiving packet 'rcvpkt';
– `from_host(rcvpkt, hostid)`: true if packet 'rcvpkt' is from host 'hostid';
– `has_seq(rcvpkt, seqnum)`: true if packet 'rcvpkt' has sequence number 'seqnum';
– `corrupt(rcvpkt)`: true if packet 'rcvpkt' is corrupted;
– `udt_send(sndpkt, hostid)`: call lower layer to send packet 'sndpkt' to host 'hostid';
– `extract(rcvpkt, data)`: extract payload of packet 'rcvpkt' into data structure 'data';
– `deliver(data)`: call upper layer to deliver data stored in 'data';
– `make_pkt(seqnum, data)`, `make_pkt(seqnum, ACK)`: return a data or acknowledgement packet with sequence number 'seqnum';
– `start_timer`: start timer;
– `stop_timer`: stop timer;
– `timeout`: called when timer runs out.
In addition, use "!" for negation, "&&" for logical AND, and "||" for logical OR.

a) For the FSM at the sender S as shown in Fig. 4, give the content of states 5, 6, 7, and 8 following the convention of the given states [1 pt], and describe the meaning of each state. [1 pt]

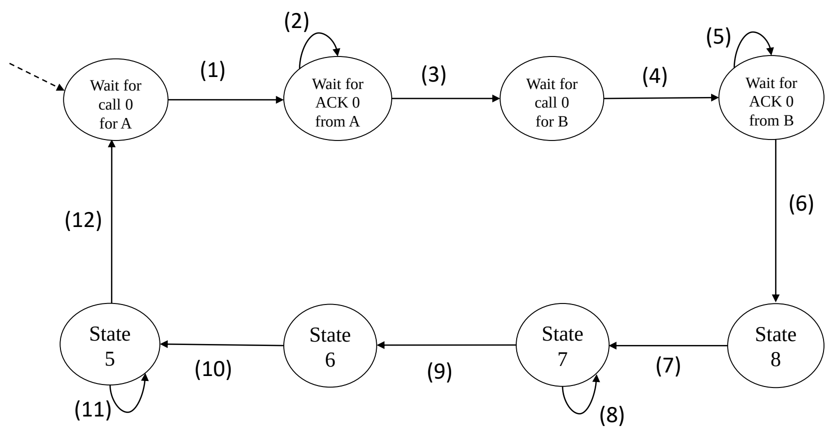

Figure 4: Problem 3: sender FSM.

state 5: "wait for ACK 1 from B", meaning waiting for acknowledgement with sequence number 1 from B
state 6: "wait for call 1 for B", meaning waiting for application data to be sent with sequence number 1 to B
state 7: "wait for ACK 1 from A", meaning waiting for acknowledgement with sequence number 1 from A
state 8: "wait for call 1 for A", meaning waiting for application data to be sent with sequence number 1 to A

Note: -0.5 pt if the order of states is reversed

b) Complete the event-action list for the following transition links. [4 pt]

(1)
```
rdt_send(data)
-------------------------
sndpkt = make_pkt(0, data)
udt_send(sndpkt, A)
start_timer
```

(2)
```
timeout
-------------------------
udt_send(sndpkt, A)
start_timer
```

(3)
```
rdt_rcv(rcvpkt) && !corrupt(rcvpkt) &&
from_host(rcvpkt, A) && has_seq(rcvpkt, 0)
-------------------------
stop_timer
```

c) Can we use the receiver FSM of one of the protocols learned in class for A? If so, give the protocol name [1 pt], its states [1.5 pt], and the meaning of each state [1.5 pt].

Yes, we can use rdt3.0 (or rdt2.2) receiver. [1 pt]

States and meaning:
"wait for 0 from below", meaning waiting for packet with sequence number 0 from the sender (that will be delivered through the network layer below) [1.5 pt]
"wait for 1 from below", meaning waiting for packet with sequence number 1 from the sender (that will be delivered through the network layer below) [1.5 pt]

Complete FSM (not required):

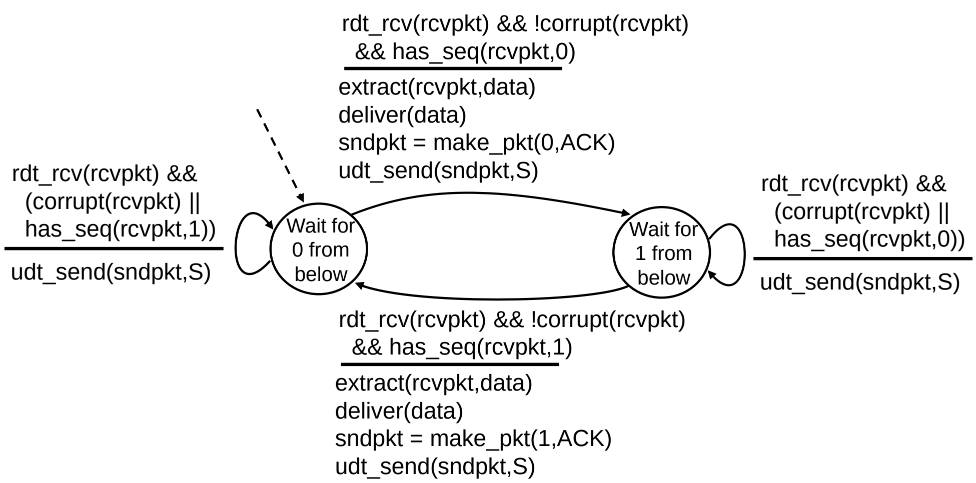

Figure 5: Problem 3: receiver FSM.