Department of Computer Science and Engineering

# EECS 3214: Computer Network Protocols and Applications

# Final Examination

**Instructor: N. Vlajic**

**Instructions:**

- Examination time: 180 min.
- Print your name and CS student number in the space provided below.
- This examination is closed book and closed notes. Use of calculators is allowed.
- There are 9 questions. The points for each question are given in square brackets, next to the question title. The overall maximum score is 100.
- Answer each question in the space provided. If you need to continue an answer onto the back of a page, clearly indicate that and label the continuation with the question number.

**FIRST NAME:** ______________________________

**LAST NAME:** ______________________________

**STUDENT #:** ______________________________

| Question | Points |
|---|---|
| 1 | / 10 |
| 2 | / 7 |
| 3 | / 12 |
| 4 | / 9 |
| 5 | / 12 |
| 6 | / 16 |
| 7 | / 12 |
| 8 | / 22 |
| Total | / 100 |

## 1. Multiple Choice, True/False [10 points]

Multiple choice questions – each question is worth [1 point].

(1) QoS parameter that measures/reflects 'variation in packet delay' is called ______________ .

- (a) reliability
- (b) throughput
- **(c) jitter**
- (d) deviation

(2) __________ is generally more suitable for implementation/use of multiple routing metrics then __________ .

- **(a) OSPF, RIP**
- (b) RIP, OSPF
- (c) both protocols are equally suitable
- (d) none of the above

(3) AS that has multiple connections to the outside world but refuses to carry transit traffic (i.e., carries only local traffic) is called ____________________ .

- (a) Stub AS
- (b) Transit AS
- **(c) Multihomed AS**
- (d) none of the above

(4) When an ICMP Error Reporting message is issued, (in its body) this message will contain ____________________ of the original IP packet that has caused/provoked the 'error'.

- (a) IP header only
- (b) IP payload only
- (c) entire IP packet (header + payload)
- **(d) none of the above**

(5) UDP uses ____________________ to handle incoming user datagrams that go to different
processes on the same host.

- (a) flow control
- (b) multiplexing
- **(c) demultiplexing**
- (d) none of the above

(6) Hosts A and B are exchanging TCP packets, sending useful data both ways (from A to B, and B to A). The value of ‘window size’ field in TCP packets sent from A to B is used to ______________________ .

(a) control the rate of packet transmission from A

**(b) control the rate of packet transmission from B**

(c) control the size of packets sent from A

(d) control the size of packets sent from B

(7) Assume a TCP client has connected to a TCP server, then exchanged a few TCP packets with the server, and finally issued a TCP-FIN request. Upon receiving the TCP-FIN request from this client, the server will ______________________ .

**(a) immediately release the receive buffer associated with this client**

(b) immediately release the send buffer associated with this client

(c) immediately release both the send and receive buffer associated with this client

(d) the server will not release any resource at this point

(8) What is usually returned when a request is made to connect to a TCP port at which no server is listening?

(a) A TCP segment with the ACK and SYN bits set to 1.

(b) A TCP segment with the ACK and FIN bits set to 1.

**(c) A TCP segment with the ACK and RST bits set to 1.**

(d) A TCP segment with the ACK and PSH bits set to 1.

(9) In most operating systems, the duration of TCP Keep-Alive Timer is set to ___________ .

(a) 1 minute

(b) 10 minutes

(c) 1 hour

**(d) 2 hours**

(10) In a network, after the load reaches and exceeds the network capacity, throughput _________________ .

(a) Increases sharply.

(b) Increases proportionally with the load.

**(c) Declines sharply.**

(d) Declines proportionally with the load.

## 2. NAT [7 points]

Consider a node A behind a NAT, sending packets to a node B. We will follow a simple packet exchange of A sending to B, having the packet altered by NAT, and then the response coming back, again altered by NAT, and then arriving at A.

You know the following:

Packet leaving A:
- Destination address: 4.3.2.1
- Source address: 192.168.2.7
- Destination port: 80
- Source port: 1067

Packet leaving B:
- Destination address: 2.2.2.2
- Source address: 4.3.2.1
- Destination port: 2500
- Source port: 80

**(a) [4 points]** For the packet leaving the NAT towards A, please fill in the blanks:

| | |
|---|---|
| Destination address: | ______________________ |
| Source address: | ______________________ |
| Destination port: | ______________________ |
| Source port: | ______________________ |

**Solution:**

Destination address: **192.168.2.7**
Source address: **4.3.2.1**
Destination port: **1067**
Source port: **80**

**(b) [3 points]** Fill in the following blanks:

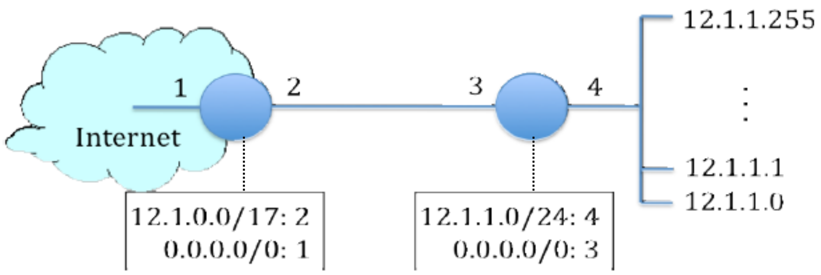
| | |
|---|---|
| IP address of the NAT: | ______________________ |
| IP address of node A: | ______________________ |
| IP address of node B: | ______________________ |

**Solution:**

IP address of the NAT: **2.2.2.2**
IP address of node A: **192.168.2.7**
IP address of node B: **4.3.2.1**

## 3. Subnetting [12 points]

A small university campus is assigned a large address block 12.1.0.0/17, but is only using a portion of these addresses (in 12.1.1.0/24) to number its computers. The campus uses a single Internet Service Provider (ISP) to reach the rest of the Internet. This picture shows the
forwarding tables on the ISP's router (on the left) and the campus edge router (on the right):



For example, the ISP forwards all packets with destination addresses in 12.1.0.0/17 out link #2 toward the campus edge router. Both routers include a default forwarding entry 0.0.0.0/0 that can match any (other) destination IP address not specified by the previous entry.

**(a) [2 points]** How many IP addresses does the campus "own" in its 12.1.0.0/17 block?

Solution:

**The 17-bit mask leaves 32-17, or 15 bits to identify the addresses within the block. As such, the block contains $2^{15}$ or 32,768 addresses.**

**(b) [2 points]** What are the smallest and largest IP addresses that the campus "owns", whether or not the campus is currently using the address?

Solution:

**12.1.0.0-12.1.127.255**

**(c) [2 points]** Suppose the ISP router receives a packet with destination IP address 12.1.1.1? What path does this packet follow? (I.e., specify the list of links this packet is sent through.)

Solution:

**The packet flows over the path 1→ 2 → 3 → 4.**

**(d) [3 points]** Suppose the ISP router receives a packet with destination IP address 12.1.6.1? What path does this packet follow? (I.e., specify the list of links this packet is sent through.)

Solution:

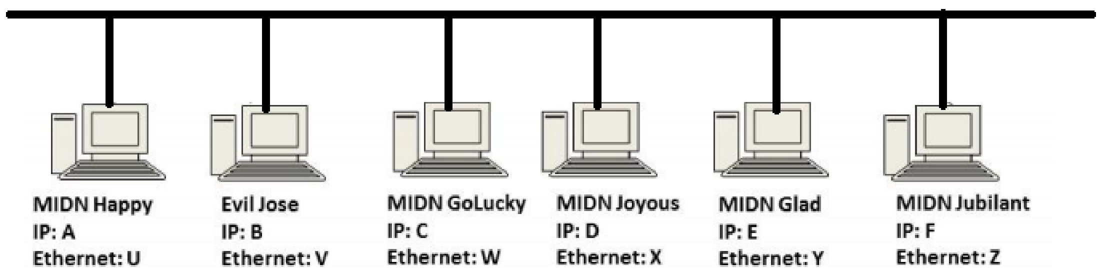
```
The packet flows over the path 1→ 2 → 3 → 2 → 3 …
```

**(e) [3 points]** What ultimately determines the number of links a packet with destination IP address 12.1.6.1 is sent through (i.e., the number of times this packet is actually forwarded)?

```
The looping packet is discarded once its IP TTL (Time-To-Live)
expires.
```

## 4. Auxiliary Network Protocols [9 points]

**4.1) [6 points]** Suppose there is a LAN network as shown in the figure below. Further assume that Evil Jose machine is in hands of a malicious hacker. The hacker would like to interfere with the internal operation of the network, so whenever any of the other hosts (i.e., their respective IP module) have an IP packet destined for MIDN Happy, the IP packet actually gets sent/delivered to the IP module of Evil Jose. (Recall, the actual delivering of a packet to an appropriate host is resolved by/at MAC layer.)



a) [2 points] Which protocol do you think the hacker should use/manipulate to have his goal accomplished?

Solution: ARP protocol

b) [4 points] How many packets (of the protocol specified in a)), and of which type and content, should Evil Jose generate to have this goal accomplished. (You can assume that the hacker is able to alternate the operation of Evil Jose's OS.)

Solution:

This type of attack is called 'ARP poisoning'. In the worst case scenario, Evil Jose should issue 4 ARP requests – requesting the MAC address of all other machines (but itself and MIDN Happy).

In these **ARP requests**:

MAC Sender: Evil Jose's MAC
IP Sender: should be **spoofed** (MIDN Happy's IP)
MAC Target: ???
IP Destination: target machine (MIDN GoLucky, MIDN Joyous, MIDN Glad, MIDN Jubilant)

When each of the four machines receive such a request, they will send a response; but before that, they will first (wrongfully) update their ARP tables, mapping MIDN Happy's IP to Evil Jose's MAC. Through this action, the hacker will successfully 'poison' their ARP tables, and ultimately accomplish his goal.

In many real-world networks, only one such ARP request is sufficient as ARP requests are sent to the LAN's broadcast address, hence all machines read them and update their ARP tables accordingly.

Any answer that talked about Evil Jose waiting for, and then responding to ARP requests for MIDN Happy, were NOT correct – as there would be no guarantees that MIND Happy would not also respond to these requests …

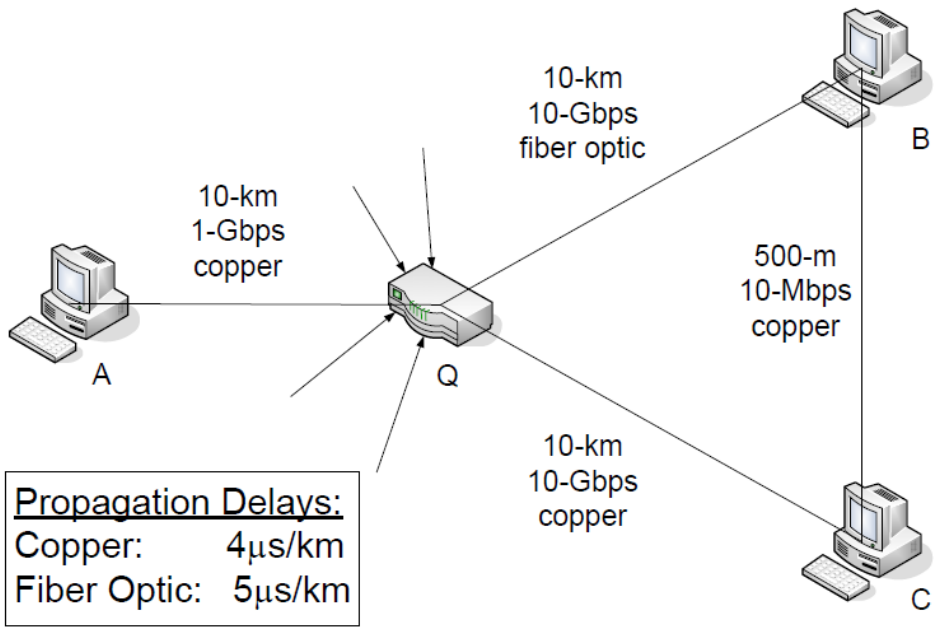
**4.2) [3 points]** We have seen in class that the implementation of ICMP Source Quench functionality, both on end-hosts and routers, is optional. Explain why it is so (i.e., why the implementation is not mandatory)?

Solution:

Source Quench ICMP messages are issued in cases when router/host drops packets due to congestion (i.e., buffer has reached its maximum capacity).
The implementation is optional (not mandatory), as we would not like to force the router/host to issue ICMP 'warning' messages at times when the router/host is already excessively busy dealing with the incoming load/traffic (i.e., cannot keep up with the load) …

## 5. Queueing [12 points]

Consider the scenario shown in the below figure. Host A is sending tiny packets to hosts B and C (neglect transmission time, just consider latency). Q is a store-and-forward switch with an average arrival rate of 10-Gbps and a buffer that contains 16 Mbytes of packets on average. (All units are in base-10.)



**(a) [3 points]** What is the average queueing delay that packets will incur going through the switch?

Solution:

$$\text{Littles result: } 16 \cdot 8 \cdot 10^6 = delay \cdot 10 \cdot 10^9$$

$$Delay = .0128s = 12.8ms = 12800\mu s$$

**(b) [2 points]** What is the overall latency between A and B?

Solution:

$$\text{Prop of A-Q} = 10\text{km} \cdot 4\mu\text{s/km} = 40\mu\text{s}$$

$$\text{Q-B} = 10\text{km} \cdot 5\mu\text{s/km} = 50\mu\text{s}$$

$$\text{Latency} = 40\mu s + 50\mu s + 12800\mu s = 12890\mu s$$

**(c) [2 points]** What is the overall latency between A and C?

Solution:

Prop of A-Q $= 10\text{km} \cdot 4\mu s/\text{km} = 40\mu s$

Q-C $= 10\text{km} \cdot 4\mu s/\text{km} = 40\mu s$

Latency $= 40\mu s + 40\mu s + 12800\mu s = 12880\mu s$

**(d) [2 points]** What is the overall latency between B and C?

Solution:

Prop of B-C $= 0.5\text{km} \cdot 4\mu s/\text{km} = 2\mu s$

Latency $= 2\mu s$

**(e) [3 points]** Without changing the network, can you propose a solution to decrease the delay between A and B?

Solution:

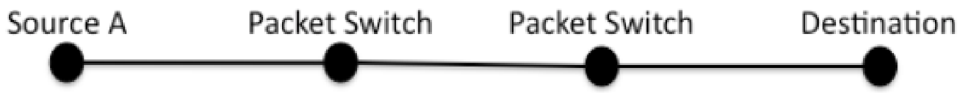
The A-B delay can be decreased if you follow the route from A-C and then go from C-B.

$12880\mu s + 2\mu s = 12882\mu s < 12890\mu s$

## 6. Packet Switching & Message Segmentation [16 points]

In modern packet-switched networks the source segments long, application-layer messages (e.g., an image or a music file) into smaller packets, and then sends these packets into the network, while the receiver reassembles the received segments back into the original message. We refer to this process as message segmentation.

Now, consider a message that is 8 Mbits long that has to be sent from source to destination as shown in the below figure. Suppose each link has a maximum capacity of 2 Mbps. Ignore propagation, queueing and processing delays.



**(a) [4 points]** Consider sending the message from source to destination without message segmentation. Keeping in mind that each switch uses store-and-forward packet switching, low long does it take to move the message from source host to destination host?

Solution:

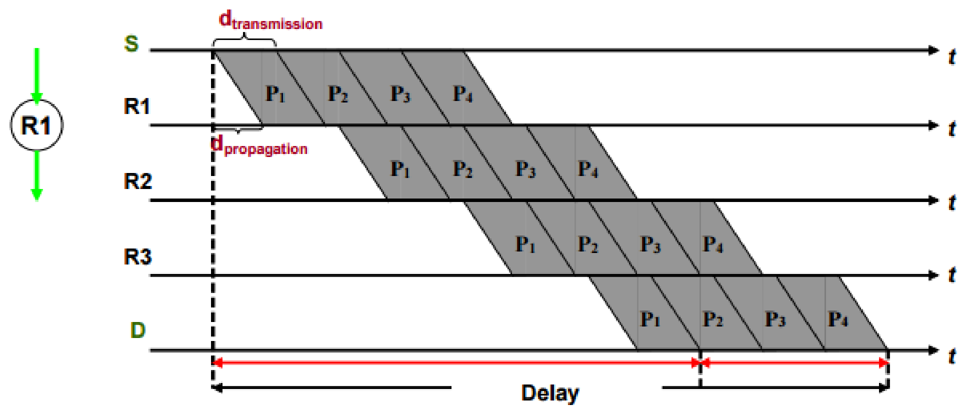
a) Time to send message from source host to first packet switch = $\frac{8\times10^6}{2\times10^6}\sec = 4\sec$. With store-and-forward switching, the total time to move message from source host to destination host = $4\sec\times 3\ hops = 12\sec$

Total delay = 12 sec

**(b) [8 points]** Now suppose that the message is segmented into 4,000 packets, which each packet being 2,000 bits long. How long does it take to move the file from source host to destination host when message segmentation is used?
Is this better or worse time/result than in a)?

Solution:

Time to send 1st packet from source host to first packet switch = . $\frac{2\times10^3}{2\times10^6}\sec = 1\ m\sec$



**d = D1 + D2 + D3 + D4, where**
**D1 = Time to transmit entire 1st packet <u>over all hops</u>**
**D2 = Time to transmit entire 2nd packet**
**D3 = Time to transmit entire 3rd packet**
**D4 = Time to transmit entire 4th packet**

time to absorb the rest of the message

Time at which 1st packet is received at the destination host = . $1\,m\sec \times 3\,hops = 3\,m\sec$. After this, every 1msec one packet will be received; thus time at which last (4000th) packet is received = $3\,m\sec + 3999 * 1m\sec = 4.002\sec$

Total delay = 4.002 sec

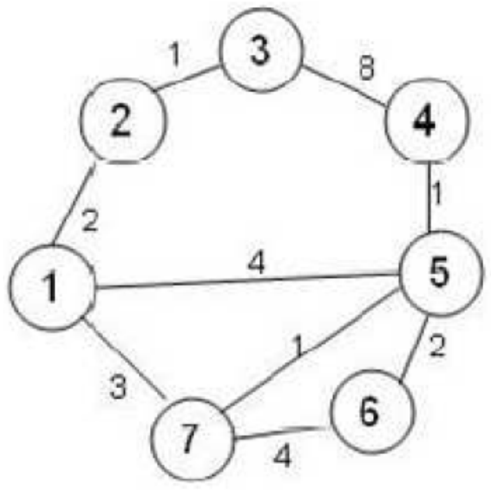
This result is significantly better than in a) – only 1/3 of the first delay!

**(c) [4 points]** What are potential drawbacks of message segmentations (as discussed in class)?

i. Packets have to be put in sequence at the destination.
ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

## 7. Routing [12 points]

**7.1) [8 points]** Run Dijkstra's algorithm on the following network to determine the routing tables for Node 3. Show all intermediate steps!



Solution:

It takes 7 steps as there are seven nodes all together. We show each step from node 3's perspective.

| Step | Set S | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | {} | inf | inf | 0 | inf | inf | inf | inf |
| 1 | {3} | inf | 1 | 0 | 8 | inf | inf | inf |
| 2 | {3,2} | 3 | 1 | 0 | 8 | inf | inf | inf |
| 3 | {3,2,1} | 3 | 1 | 0 | 8 | 7 | inf | 6 |
| 4 | {3,2,1,7} | 3 | 1 | 0 | 8 | 7 | 10 | 6 |
| 5 | {3,2,1,7,5} | 3 | 1 | 0 | 8 | 7 | 9 | 6 |
| 6 | {3,2,1,7,5,4} | 3 | 1 | 0 | 8 | 7 | 9 | 6 |
| 7 | {3,2,1,7,5,4,6} | 3 | 1 | 0 | 8 | 7 | 9 | 6 |

**7.2) [4 points]** Explain what is meant by the count-to-infinity problem in distance vector routing. How does BGP, which uses distance vector, avoid this problem?

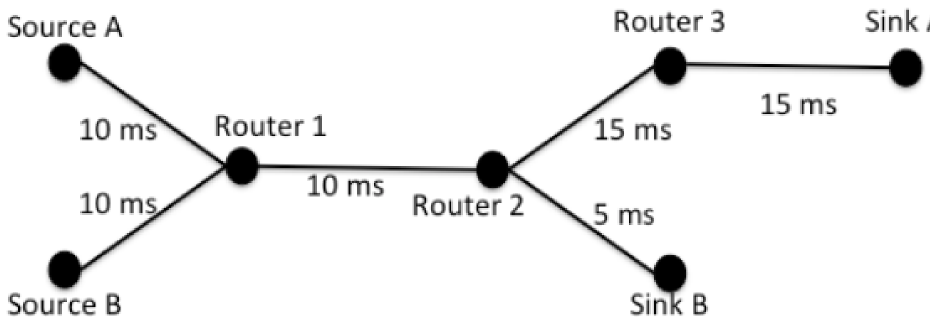
Solution:

```
Slow to hear about failed links since information propagated is only the
number of hops. A failed link may still be in the routing table due to
cycles in the possible routes (i.e. A->B->A->C).

BGP avoids this problem by maintaing an explicit path, not just a hop
count. This allows routes with cycles to be eliminated easily.
```

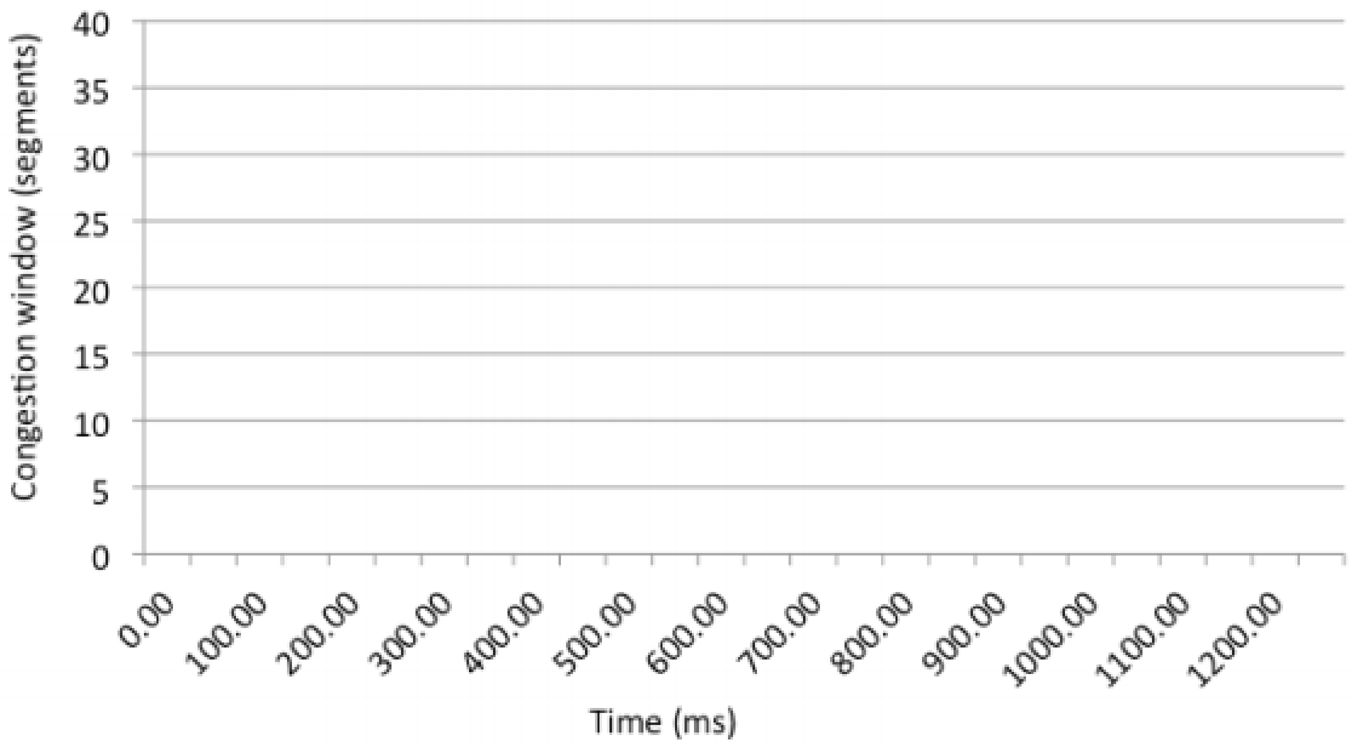
## 8. TCP Potpourri [22 points]

**8.1) [12 points]** For this problem you should familiarize yourself with the below figure first. Namely, assume that in the network shown in the below figure two parallel TCP transmissions are performed. TCP1 is a transmission between Source A and Sink A that uses TCP Tahoe. TCP2 is a transmission between Source B and Sink B that uses TCP Reno. Initial sstresh for both TCP transmissions is set to 32. In this specific scenario no additional delay through forwarding is introduced. In other words, RTT is only composed of the sums of the delay indicated on each link, times two!

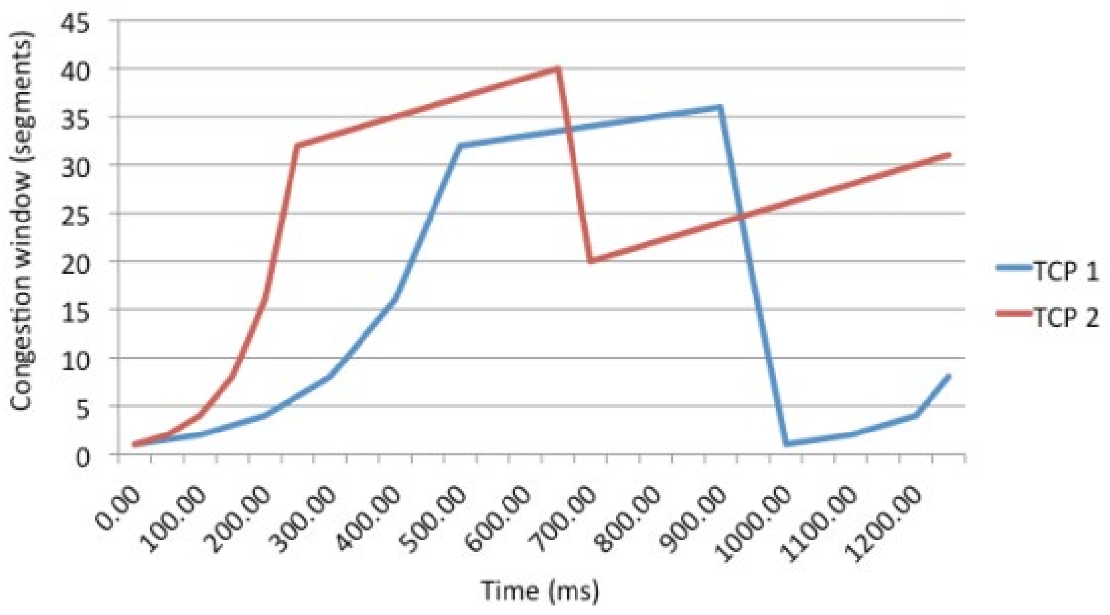


a) [6 points] For the TCP1 transmission, in the provided graph, draw the change in the size of the resulting congestion window, assuming that a packet loss (triple duplicate ACKs) is detected at time t=900ms.

b) [6 points] For the TCP1 transmission, in the same provided graph, draw the change in the size of the resulting congestion window, assuming that a packet loss (triple duplicate ACKs) is detected at time t=650ms.



Solution:



**TCP1:**
**RTT = 100 ms**

| | |
|---|---|
| **Timeline:** | **100ms – window size = 2** |
| | **200ms – window size = 4** |
| | **300ms – window size = 8** |
| | **400ms – window size = 16** |
| | **500ms – window size = 32** |

**TCP2:**
**RTT = 50 ms**

| | |
|---|---|
| **Timeline:** | **50ms – window size = 2** |
| | **100ms – window size = 4** |
| | **150ms – window size = 8** |
| | **200ms – window size = 16** |
| | **250ms – window size = 32** |

**At 650ms, the window size will increase by 400/50=8. Hence, the total window size at 650ms is 32+8 = 40.**
**Consequently, the new sstresh = 20**

**8.2) [5 points]** Show the acknowledgments (i.e., respective ACK numbers) in response to the following TCP segments, arriving in the given order.

Note: the 1st number in a segment is the segment's sequence number, and the 2nd number is the amount of application-layer data carried in the segment.

| | |
|---|---|
| **Seg(2501, 500),** | **respective Ack (_______)** |
| **Seg(4001, 200),** | **respective Ack (_______)** |
| **Seg(4201, 300),** | **respective Ack (_______)** |
| **Seg(3001, 1000),** | **respective Ack (_______)** |
| **Seg(4501, 100),** | **respective Ack (_______)** |

Solution:

| | |
|---|---|
| **Seg(2501, 500),** | **respective Ack (_3001_)** |
| **Seg(4001, 200),** | **respective Ack (_3001_)** |
| **Seg(4201, 300),** | **respective Ack (_3001_)** |
| **Seg(3001, 1000),** | **respective Ack (_4501_)** |
| **Seg(4501, 100),** | **respective Ack (_4601_)** |

**8.3) [5 points]** Assume that a TCP process A first measures the actual round trip time to another TCP process to be 30 ms, and A thus sets its estimated round trim time to be 30 ms. The second actual round trip time that A sees is 60 ms. In response, A increases its estimated round trip to 50 ms. The third actual round trip time that A sees is 40 ms. What is the next (3rd) estimated round trip computed by A? Justify your answer.

**Solution:**

Based on the provided results, it is clear that in this case TCP does not use the simple average, but weighted/smoothed average to estimate RTTs. Thus

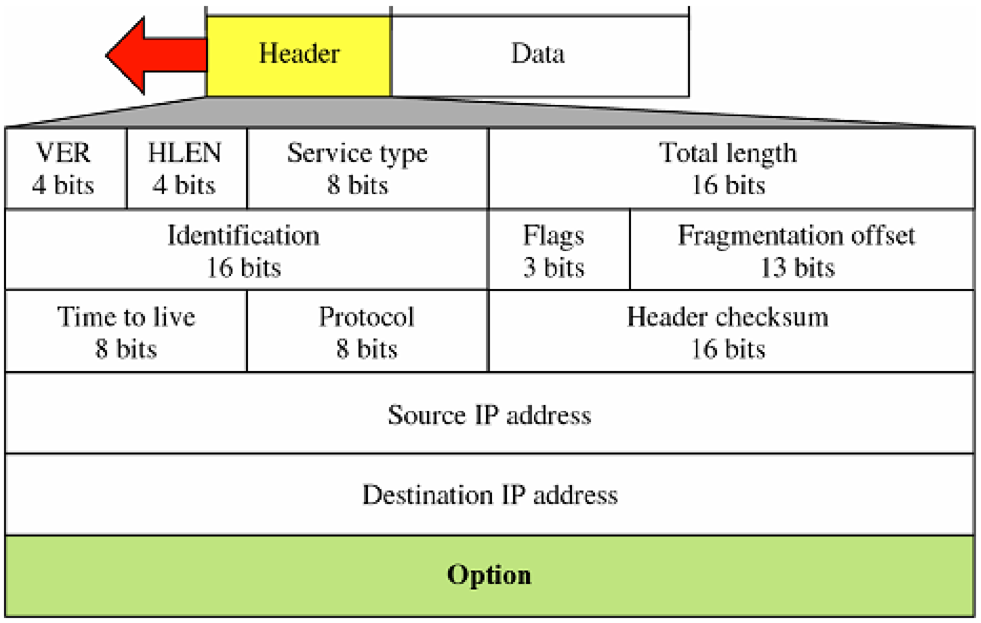
**New-estimated-RTT = x * Old-estimated-RTT + (1-x) * New-observed-RTT**

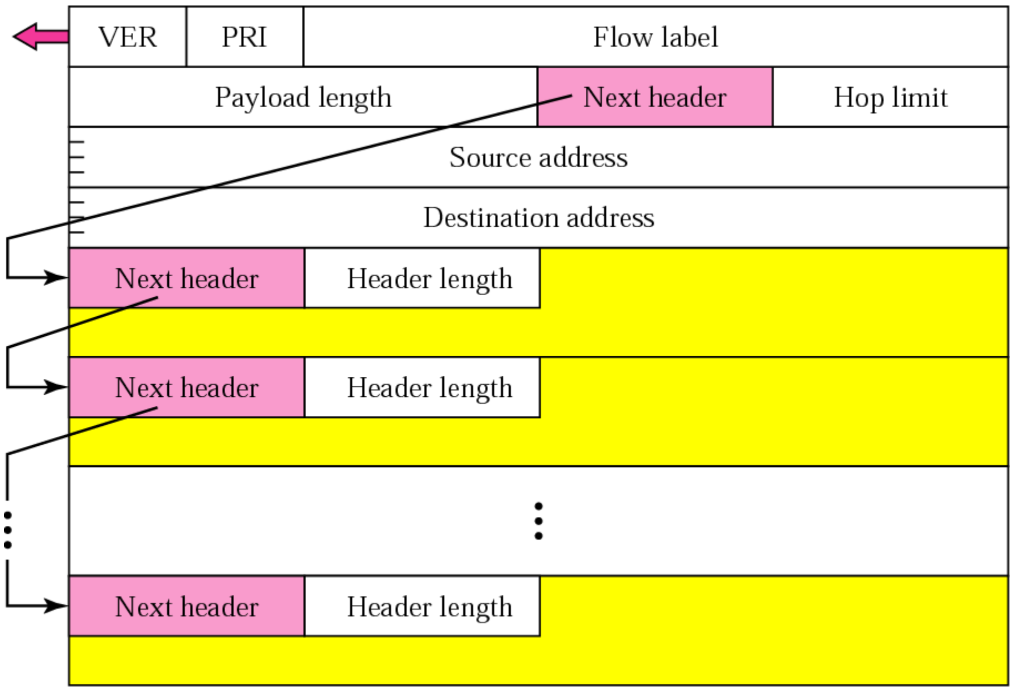
**50 = x * 30 + (1-x)*60 ⇒ 50 = -30*x + 60**

**So, x = 1 / 3**

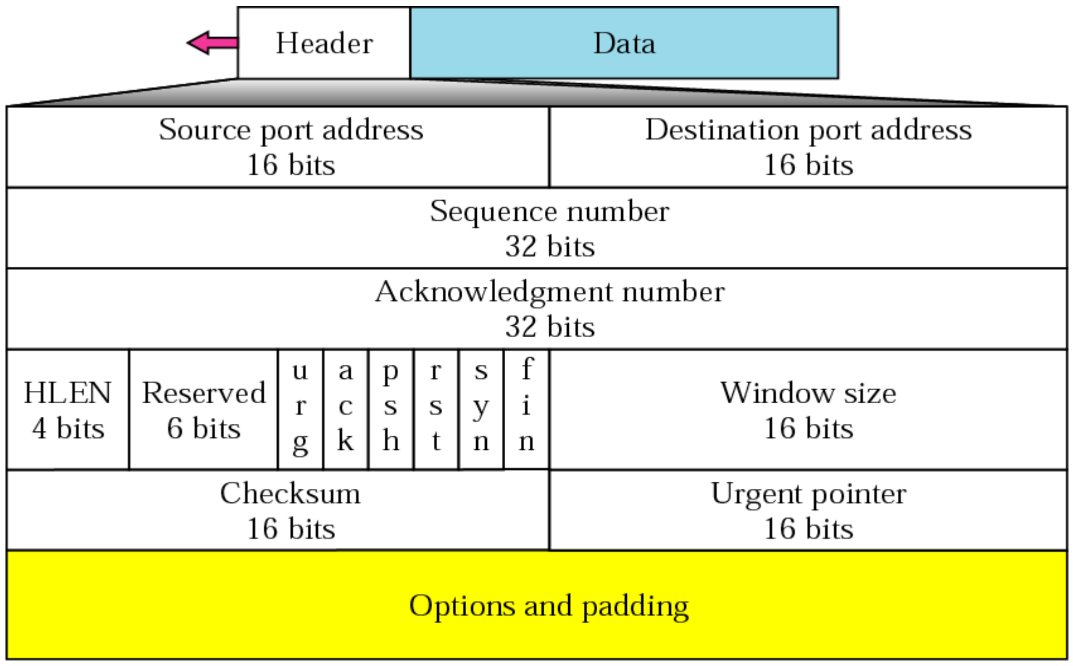
**And New-estimated-RTT = 1/3 * 50 + 2/3 * 40 = 130 / 3 = 43.33 [msec]**

**IPv4 and IPv6 PACKET FORMATs:**





**TCP PACKET FORMAT:**


Header
Data
Source port address
16 bits
Destination port address
16 bits
Sequence number
32 bits
Acknowledgment number
32 bits
HLEN
4 bits
Reserved
6 bits
urg
ack
psh
rst
syn
fin
Window size
16 bits
Checksum
16 bits
Urgent pointer
16 bits
Options and padding