Department of Computer Science and Engineering

CSE 3214: Computer Network Protocols and Applications

Final Examination

Instructor: N. Vlajic
Date: April 15, 2010

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 8 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	/ 8
3	/ 14
4	/ 15
5	/ 8
6	/ 10
7	/ 17
8	/ 18
Total	/ **100**

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

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

a) The loopback (IP) address is a member of

(a) Class A network.

(b) Class B network.

(c) Class C network.

(d) None of the above.

b) IPv6 addresses are _____ bytes long.

(a) 8

(b) 16

(c) 32

(d) 128

c) 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.

d) Which of the following are true statements about TCP.

(a) The slow-start algorithm increases a source's rate of transmission faster than the additive-increase.

(b) Setting RTO (retransmission timeout value) to a value less than the measured RTT may lead to unnecessary retransmissions.

(c) TCP segments can only be lost when router queues overflow.

(d) TCP connection termination procedure is called two-way handshaking.

e) In a network, after the load reaches and exceeds the capacity, throughput _____ .

(a) Increases sharply.

(b) Increases proportionally with the load.

(c) Declines sharply.

(d) Declines proportionally with the load.

f) The HTTP request line contains a _____ method to get information about a document without retrieving the document itself.

(a) HEAD.

(b) POST.

(c) COPY.

(d) None of the above.

g) DNS can use the services of _____ on the well-know port 53.

(a) UDP.

(b) TCP.

(c) Either (a) or (b).

(d) None of the above.

h) Suppose a Certificate Authority (CA) has Bob's certificate registered with it, binding Bob's public key to Bob. This certificate is signed with:

(a) Bob's public key.

(b) The CA's public key.

(c) Bob's private key.

(d) The CA's private key.

i) Suppose we choose a small value for a fixed playout delay for a real-time interactive multimedia application. This will result i:

(a) Less loss, less interactivity.

(b) Less loss, higher interactivity.

(c) More loss, less interactivity.

(d) More loss, higher interactivity.

j) The _____ (shaping) algorithm allows idle hosts to accumulate credit for the future transmissions.

(a) Leaky bucket.

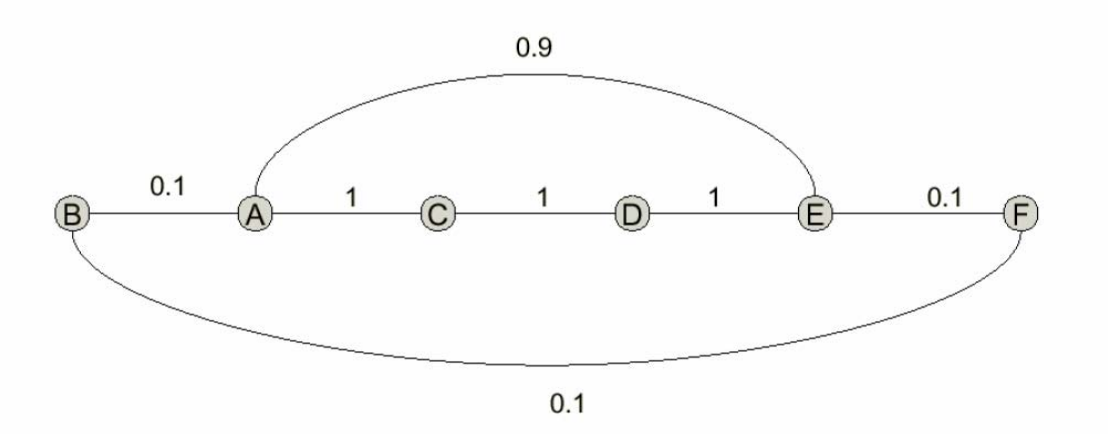
(b) Token bucket.

(c) Early random detection.

(d) None of the above.

2. Routing [8 points]

In this problem you will be asked to compute distance vector(s) using the Bellman Ford algorithm for the network below:

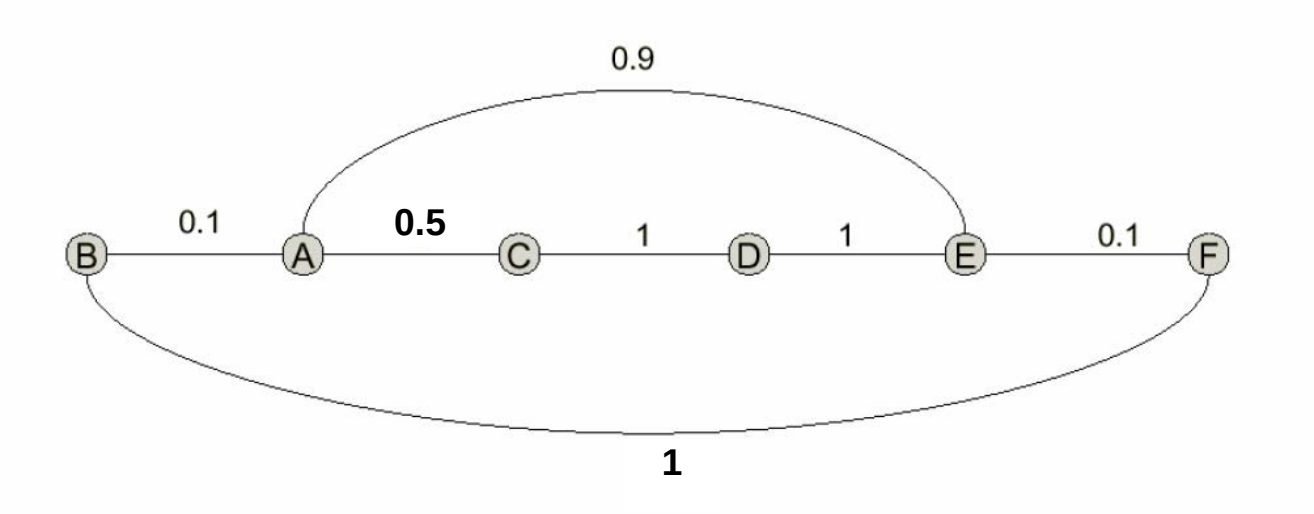


Assume that time is slotted (t=1, 2, 3, …) and that a node sends its distance vector estimates to its neighbors at the beginning of each slot. A distance vector estimate sent at the beginning of a slot arrives at the end of that slot. All distance estimates are computed using the most recently available estimates.

2.1) [5 points] For the above stated problem, what are node A's distance vectors at the beginning of the time slots 1, 2, 3, and 4?

	A	B	C	D	E	F
1	0	0.1	1	∞	0.9	∞
2	0	0.1	1	**1.9**	0.9	**0.2**
3	0	0.1	1	**1.9**	**0.3**	**0.2**
4	0	0.1	1	**1.3**	**0.3**	**0.2**

2.2) [3 points] This time assume that the cost/weight of link AC is 0.5, while the cost/weight of link BF is 1. How many iterations are required, in this case, for A's distance vectors to converge?



	A	B	C	D	E	F
1	0	0.1	0.5	∞	0.9	∞
2	0	0.1	1	**1.5**	0.9	**1**

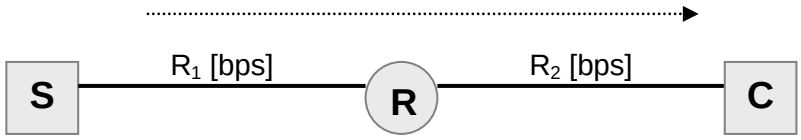
3. Packet Switching / Network Layer Potpourri [14 points]

3.1 [9 points] Assume there is one router and two links between the file server and client, as shown in the figure below. The first link has transmission rate R_1 and the second link has transmission rate R_2. Assume the file gets broken into three packets, each of size L. Ignore all propagation and processing delays. Answer the following thee questions:

(a) How long does it take from when the server starts sending the file until the client has received the whole file if $R_1 \leq R_2$?

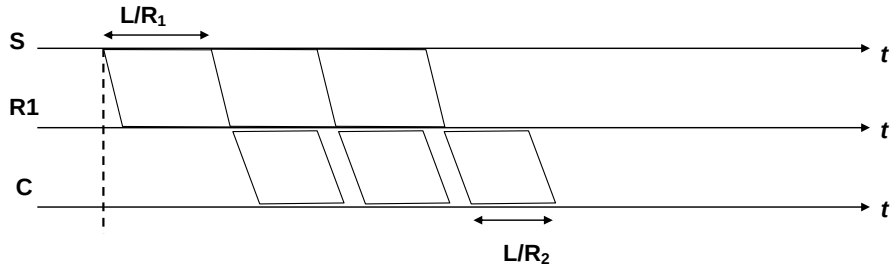
(b) How long does it take from when the server starts sending the file until the cline has received the whole file if $R_1 > R_2$?

(c) In case (b), how long does the second packet spend in the router's queue?

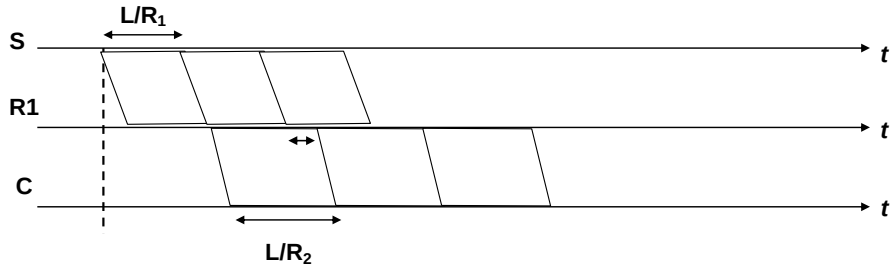


Solution:

(a) delay = 3*L / R_1 + L/R_2



(b) delay = L / R_1 + 3*L/R_2



(b) queueing delay = L / R_2 - *L/R_1

3.2 [5 points] A router has the following CIDR entries in its routing table:

Address/mask	Next hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Router 1
default	Router 2

For each of the following IP addresses, what does the router do if a packet with that address arrives?

(a) 135.46.63.10

Taking the first 22 bits of 135.46.63.10 as network address, we have 135.46.60.0. It matches the network address of the 2nd row. The packet will be forwarded to **Interface 1.**

(b) 135.46.57.14

Taking the first 22 bits of the above IP address as network address, we have 135.45.56.0. It matches the network address of the first row. The packet will be forwarded to **Interface 0.**

(c) 135.46.52.2

Taking the first 22 bits of the above IP address as network address, we have 135.45.52.0. It dose not match the network addresses of the first three rows. The packet will be forwarded to default gateway which is **Router 2.**

(d) 192.53.40.7

Taking the first 23 bits of the above IP address as network address, we have 192.53.40.0. It matches the network address of the third row. The packet will be forwarded to **Router 1.**

(e) 192.53.56.7

Taking the first 23 bits of the above IP address as network address, we have 192.53.56.0. It does not match the network addresses of the first three rows. The packet will be forwarded to default gateway which is **Router 2.**

4. TCP Potpourri [15 points]

4.1) [5 points] Consider a TCP connection between two machines (A and B) in an environment with 0% packet loss. Assume the round trip time (RTT) between the two machines is 4 [seconds], and the segment size is 3 [Kbytes]. The bandwidth of the connection is 500 [kbps]. What is the smallest TCP window size for which there will be no stalling? (We say a TCP connection experiences no stalling if the acknowledgments arrive back to the sending machine before the sliding window over the send buffer close to zero. I.e., TCP packets are continuously, back-to-back, sent out of the sending machine.)

Solution:

There will be no stalling if

time-to-send-entire-window ≤ time-for-first-ack-to-arrive-back

That is:

$$W*S / R \leq RTT + S / R$$

Hence: $W \leq RTT * R / S + 1$

In this particular case: $W \leq 4 \text{ sec} * 500 \text{ kbps} / 24 \text{ kbits} + 1 = 83.3 + 1 = 84.3$

That is: $W \leq 84$

4.2) [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 next actual round trip time that A sees is 60 ms. In response, A increases its estimated round trip to 50 ms. The next actual round trip time that A sees is 40 ms. What is the next 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, $x = 1 / 3$

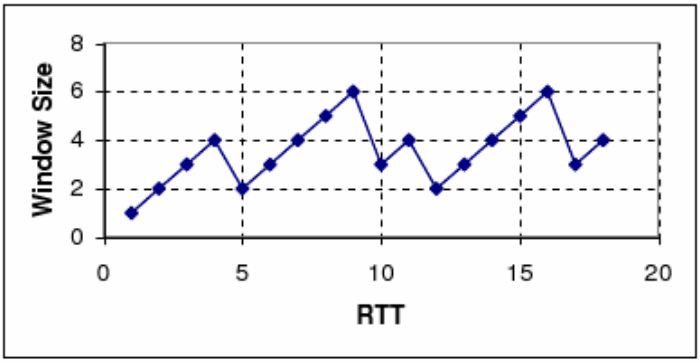
And **New-estimated-RTT = 1/3 * 50 + 2/3 * 40 = 130 / 3 = 43.33 [msec]**

4.3) [5 points] Consider an additive-increase multiplicative-decrease congestion control algorithm (window size increases linearly and it is halved when congestion is detected), with no slow start, that also works in units of packets rather than bytes. The algorithm starts each connection with a congestion window equal to one packet. Assume that the delay is only due to propagation (infinite transmission capacity) and when a group of packets is sent, only a single cumulative acknowledgment is returned.

Fill out the table below, and consequently plot the size of congestion window as a function of the round-trip times for the situation in which the following packets are lost: 9, 25, 30, 49. For simplicity assume a perfect timeout mechanism that detects a lost packet exactly 1 RTT after it is transmitted.

RTT	1	2	3	4	5	6	7	8	9
Window size	1	2	3	4	2	3	4	5	6
Packet sent	1	2,3	4,5,6	7,8,9,10	9,10	11,12,13	14-17	18-22	23-28

RTT	1	2	3	4	5	6	7	8	9
Window size	3	4	2	3	4	5	6	3	4
Packet sent	25-27	28-31	30, 31	32-34	35-38	39-43	44-49	49-51	52-55



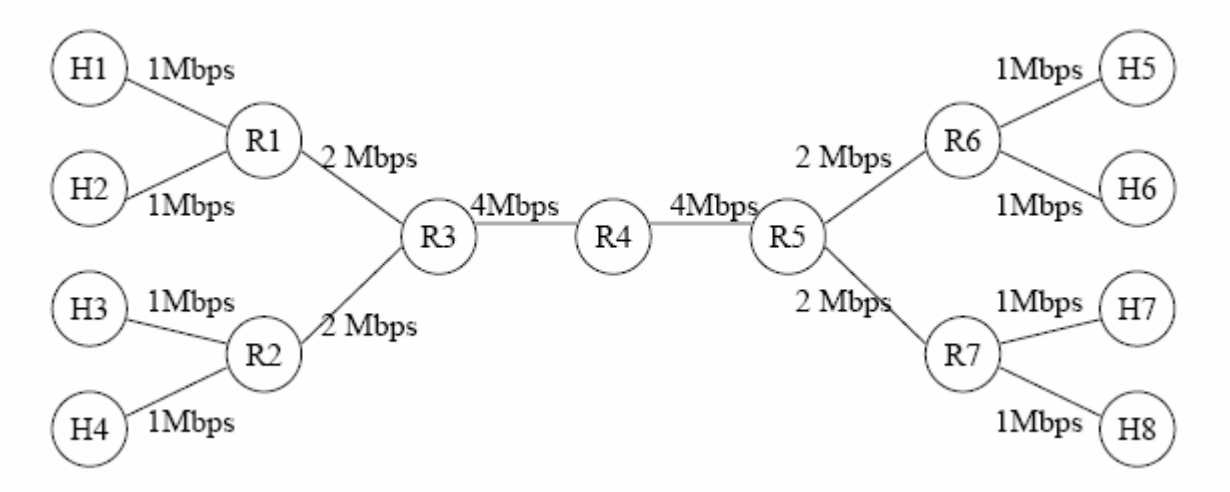
5. Congestion / Scheduling [8 points]

5.1) [4 points] Consider the network below, with eight hosts H1, .., H8, and seven routers R1, .., R7, each of which is much faster (in terms of processing) than any of the links. All links are full-duplex with bandwidths as shown in the figure. (A link is full-duplex if it allows that data be transmitted in both directions, at the same specified rate.) Answer the following:

a) Which routers can never be congested?

b) Which routers are vulnerable to congestion?

For the routers that can get congested, show specific traffic patterns that could cause congestion.

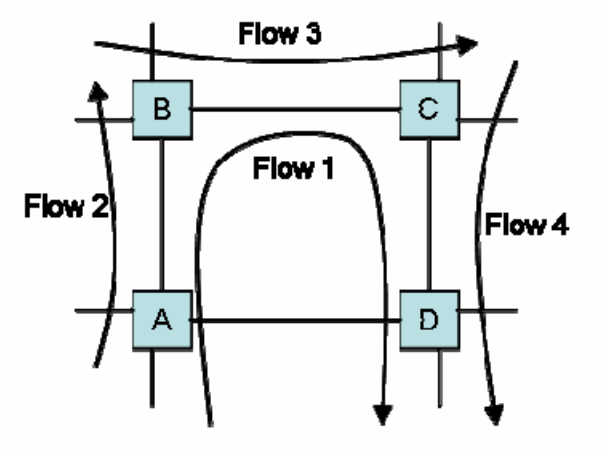


Solution:

Router R4 can never be congested. The worst case scenarios are either H1,…,H4 want to send 4 Mbps to either of H5,…,H8 or vice versa. In either case, there is enough link capacity on both sides.

All other routers can become congested.
• Suppose H2, H3 and H4 want to send 1Mbps each to H1. In this case, R1 becomes congested (similar scenarios apply to R2, R6 and R7, e.g. suppose H1, H2, and H3 want to send 1Mbps each to H4. In this case, R2 becomes congested).
• Suppose that all H2,…,H8 want to send 1Mbps each to H1. In this case, R3 also becomes congested.

5.2) [4 points] Consider the network below consisting of four routers. Every link has capacity of 1 Mbps, and every flow sends data at 1 Mbps. Assume that all flows are UDP and use the same packet size.



a) What is the throughput of each flow (i.e. actual rate at which packets of this flow arrive at its respective destination), if all routers implement FIFO scheduling. FIFO scheduling implies that in the case of congestion, each packet is dropped with the same probability.

Solution:

Flow 1: 1/4Mbps, Flow 2: 1/2Mbps, Flow 3: 2/3Mbps, Flow 4: 3/4Mbps. Because each link is congested (the sum of the arrival rates of the flows at each link is greater than 1Mbps), ach flow gets a throughput proportional to its arrival rate. On link AB, the arrival rates of both Flow 1and Flow 2 is 1Mbps, so each flow gets 0.5Mbps. On link AB, the arrival rate of Flow 1 is 0.5Mbps while the arrival rate of Flow 3 is 1Mbps, so Flow 3 will get 2/3Mbps, while Flow 1 will get 1/3Mbps. Finally, on link CD, the arrival rate of Flow 1 is 1/3Mbps, while the arrival rate of Flow 4 is 1Mbps, so Flow 4 will get 3/4Mbps, while Flow 1 will get 1/4Mbps.

b) What is the throughput of each flow if all routers implement Weighted Fair Queueing, and each Flow i has weight i?

Solution:

Flow 1: 1/5Mbps, Flow 2: 2/3Mbps, Flow 3: 3/4Mbps, Flow 4: 4/5Mbps. On link AB, Flow 1 gets 1/3Mbps while Flow 2 gets 2/3Mbps; on link BC Flow 2 gets 1/4Mbps, while Flow 3 gets 3/4Mbps; on link CD Flow 1 gets 1/5Mbps, and Flow 4 gets 4/5Mbps.

6. Multimedia [10 points]

6.1) [6 points] Recall the two FEC schemes for recovery of packet loss in multimedia applications: Redundant XOR-ing and Redundant Streaming. Suppose the first scheme generates a redundant chunk for every four original chunks. Suppose the second scheme uses a low-bit rate encoding whose transmission rate is 25 percent of the transmission rate of the nominal stream.

a) How much additional bandwidth does each scheme require?

Solution:

25%

b) How much playback delay does each scheme add?

Solution:

Playback delay of 1st scheme is 5 packets. Playback delay of 2nd scheme is 2 packets

c) How do the two schemes perform if the first packet is lost in every group of 5 packets. Which scheme will have better audio quality?

Solution:

The first scheme will be able to completely reconstruct the original audio stream. The second scheme will produce lower quality audio.

c) How do the two schemes perform if the first packet is lost in every group of 2 packets. Which scheme will have better audio quality?

Solution:

In this case, the first scheme will not be able to reconstruct many of the original packets; hence the quality of the audio will be very poor. The second scheme will be able to replace all lost packets with their lower-quality version; hence, it will produce audio of overall better quality than the first scheme.

6.2) [4 points] Suppose a server transmits one frame of a video every second, and the client starts playing the video at one frame per second as soon as the first frame arrives. Suppose the first ten frames arrive at times 0, 1.2, 1.99, 4.17, 4.01, 5.03, 8.05, 7.50, 8.90, 8.99, all in seconds.

a) Which frames reach the client too late for playout?

Solution:

Starting with frame 0 at time 0, frame i should play at time i. Hence, the frames arriving too late to play are frame **1** (1.2 instead of 1), **3** (4.17 instead of 3), **4** (4.01 instead of 4), **5** (5.03 instead of 5), **6** (8.05 instead of 6), **7** (7.50 instead of 7), and **8** (8.90 instead of 8). Only frames **2** (1.99 instead of 2) and **9** (8.99 instead of 9) arrive in time.

b) How much extra playout delay is needed to ensure that all frames have arrived by the time the client needs to play them?

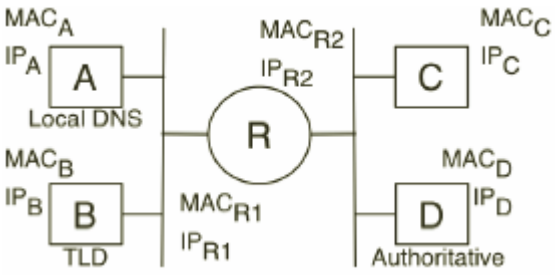
Solution:

The worse offender is frame 6 that arrive 2.05 seconds early. A playout time of **2.05 seconds** is enough to ensure that all frames arrive in time for playout.

7. DNS and HTTP Potpourri [17 points]

7.1) [4 points] Consider the following network where Local DNS Server A receives a request to resolve host C's address. A already has the IP address of the appropriate (nearest) top-level DNS server (B) cached. The authoritative DNS server for C is D.

Circle R represents a router connecting two network segments. Router R has two NICs – one NIC (R_1) is connected to the left network, and the other NIC (R_2) is connected to the right network. Assume the Address Resolution Protocol (ARP) table at each host and router is pre-populated with the MAC and IP addresses of hosts in the same local area network. The MAC address and the IP address of any particular host X (i.e. NIC) are denoted as MAC_X and IP_X respectively.



Now, if B does not have a record for C, and it does NOT support DNS recursive queries, write down the sequence of Ethernet frames during the DNS lookup process by filling the following table. (The number of rows is not necessarily equal to the number of frames during the process.)

Frame Number	Source MAC address	Destination MAC address	Source IP address	Destination IP address
1	MAC_A	MAC_B	IP_A	IP_B
2	MAC_B	MAC_A	IP_B	IP_A
3	MAC_A	MAC_{R1}	IP_A	IP_D
4	MAC_{R2}	MAC_D	IP_A	IP_D
5	MAC_D	MAC_{R2}	IP_D	IP_A
6	MAC_{R1}	MAC_A	IP_D	IP_A
7				

7.2) [4 points] Repeat question (a) assuming B now supports DNS recursive queries.

Frame Number	Source MAC address	Destination MAC address	Source IP address	Destination IP address
1	MAC_A	MAC_B	IP_A	IP_B
2	MAC_B	MAC_{R1}	IP_B	IP_D
3	MAC_{R2}	MAC_D	IP_B	IP_D
4	MAC_D	MAC_{R2}	IP_D	IP_B
5	MAC_{R1}	MAC_B	IP_D	IP_B
6	MAC_B	MAC_A	IP_B	IP_A
7				

7.3) [4 points]

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 host, so a DNS look up is necessary to obtain the given IP address. Further suppose that 3 DNS servers are visited before your host receives the IP address form DNS. The round trip time between the client and the i^{th} DNS server is T_i $(1 \le i \le 3)$, and that between the j^{th} DNS server and the $(j+1)^{th}$ DNS server is t_j $(1 \le j \le 2)$.

Further suppose that the page contains four objects (including the webpage) and that the web browser has to fetch the webpage (the .html file) before it knows 3 (other) image objects are contained in the webpage. Assume that it takes negligible time to transmit the .html file (i.e., the transmission time for the file is zero), but O_i time is required to transmit the i^{th} image object. Let T_0 denote the propagation delay (i.e., RTT excluding transmission time) between the local host and the server containing the objects.

How much time elapses from the time instant when the client clicks on the link until the time instant when the client receives a complete webpage? **Assume iterative DNS query service and persistent HTTP with pipelining.**

Solution:

Overall delay (download time) = T_1 (for DNS1) + T_2 (for DNS2) + T_3 (for DNS3) + + T_0 (for TCP connection with server) + T_0 (for obtaining html file) + T_0 (for all images at once) + O_1 + O_2 + O_3

Overall delay (download time) = $T_1 + T_2 + T_3 + 3*T_0 + O_1 + O_2 + O_3$

7.4) [5 points]

Repeat 7.3) but this time assuming that **recursive DNS query service and non-persistent HTTP with parallel connections** are used. (In case of parallel HTTP, the client is allowed to establish multiple TCP connections with the server in order to speed up the download of a particular web page.)

Solution:

Overall delay (download time) = T_1 (for DNS1) + t_1 (between DNS1 and DNS2) + t_2 (between DNS2 and DNS3) + T_0 (for TCP connection for html file) + T_0 (for obtaining html file) + T_0 (for TCP connection for images) + T_0 (for obtaining images) + max (O_1 + O_2 + O_3)

Overall delay (download time) = $T_1 + t_1 + t_2 + 4*T_0 + \max\{O_1 + O_2 + O_3\}$

8. Security [18 points]

8.1) [10 points] Assume Alice wants to send confidential emails to Bob. K_B^+ is Bob's public key, and K_A^- is Alice's private key for signing. For each of the following, discuss whether the given option provides sufficient message confidentiality, sender authenticity, and data integrity. I.e., in each case (a) to (c), circle the options that apply.

Note: When considering this problem, it is reasonable to assume that Alice will not send an email with the same content (i.e. she will not send the same M) twice.

a) Alice sends $[K_B^+(M), K_A^+(M)]$

- Message confidentiality
- Sender authenticity
- Data integrity

b) Alice sends $[K_B^+(M), K_A^-(K_B^+)]$

- **Message confidentiality**
- Sender authenticity – No. Trudy could have overheard a previous transmission, steal the signature, and reuse it for her own / injected messages.
- Data integrity – No. In the case of man-in-the-middle attack, $K_B^+(M)$ could be changed to $K_B^+(M')$

c) Alice sends $[K_B^+(M), K_A^-(M)]$

- Message confidentiality
- **Sender authenticity**
- **Data-integrity**

d) Alice sends $[K_B^+(M), K_A^-(K_B^+(M))]$

- **Message confidentiality**
- **Sender authenticity**
- **Data integrity**

e) Alice sends $[K_B^+(M), K_A^+(K_B^+(M))]$

- **Message confidentiality**
- Sender authenticity
- Data integrity

8.2) [4 points] The software company PerfectSecurity is selling a new defence software against DDoS attacks. Their software looks at the source IP address of all incoming packets, and if it finds any IP address that accounts for more than 1% of overall traffic over the last hour, it installs an entry in the router that blocks all packets from that address for the next 24 hours. PerfectSecurity's marketing people are claiming that this will stop all DDoS attacks effectively.

(a) Name at least two reasons why PerfectSecurity's software is not a good solution to the problem?

(b) Explain how PerfectSecurity's solution could be mis-used (by a malicious third party) to prevent a legitimate user from accessing a web-site protected by their software.

Solution:

a.1) It's too easy to break: with more than 100 zombies, you can flood the victim's network link without any zombie consuming more than 1% of traffic.

a.2) It's too easy to evade detection with forged source addresses. You just use a different (forged) IP address on every packet.

b) Attackers could exploit this to cause collateral damage to innocent third parties. If CNN is using
this, an attacker could prevent Joe from being able to reach CNN by sending a large number of
packets whose IP address has been forged to look like they came from Joe.

8.3) [4 points] Suppose two directly connected routers A and B exchange their routing information by means of IP packets. A third party C, several hops away, could conceivably launch a denial-of-service attack on (e.g.) router B by sending unwanted packets to this router. To defend B from such attacks, the network operator might install a packet filter that discards all packets destined to B, except for packets sent from IP address A. However, as a countermeasure, C could now easily send "spoofed" packets to B, and get to place unwanted load on router B in spite of the presence of the packet filter.

Propose a measure that would effectively defend B against such (spoofed-packet) type of attacks by exclusively relying on the inherent features of IP protocol.

Solution:

The 8-bit TTL field has a maximum value of 255. C cannot set a value higher than 255, and each hop along the path from C to A decrements the value (discarding when 0 is reached). As such, it is impossible for C to direct a packet to A that will have a TTL value of 254 when it reaches A.

IP PACKET FORMAT: