EE122 FINAL EXAM: 2003-12-13

Scott Shenker, Ion Stoica

Last name ____________________ **First name** ____________________

Student ID ____________________ **Login: ee122-**____

Please circle the last two letters of your login.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Discussion section meeting time ____________________TA's name __________

This is a closed-book exam. This booklet has X numbered pages including the cover page, and contains Y questions. You have 3 hours to complete the exam. Good luck.

Question	Name	Max Score	Your Score
1	Alphabet Soup + Short Questions	20	
2	Potpourri	20	
3	Link-State Routing	15	
4	Metrics + QoS	20	
5	RED	15	
6	CAN of Worms	10	
7	Blanche Dubois	truffle	
		Total	

Question 1 (20 pts) ***Alphabet Soup + Short Questions:***

(a) Alphabet soup: **(11 pts)**
Here are a list of acronyms: A, CAN, CNAME, CTS, DD, DHCP, DNS, DVMRP, ECN, MIMD, MX, NS, PIM, RTS, SYN, ToS.
You are to fill out the following table, by writing each of these acronyms in its corresponding row. Note that some rows will contain more than one acronym.

The four kinds of DNS records (4 answers)	A CNAME MX NS
A distributed hash table algorithm	CAN
A way to explicitly signal congestion in packet headers	ECN
Bits in IP packet header now used for Diffserv	ToS
A protocol used to automatically allocate IP addresses	DHCP
A packet sent to initiate a transmission in 802.11, and its response (2 answers)	RTS
A method of window increase and decrease *not* used in TCP	MIMD
The system used to translate host names into IP addresses	DNS
A packet sent to initiate a TCP connection	SYN
Two multicast routing protocols (2 answers)	PIM DVMRP
The initials of a method used in sensornets to retrieve information	DD

(b) Short questions: **(9 pts)**
Please be as concise as possible.

(i) Why is AIMD (Additive Increase, Multiplicative Decrease) used instead of MIMD or AIAD in TCP?
Soln: Additive increase: is used when we already have some idea of the capacity of the link (ssthresh is set). In this case, we want to gently explore to see if link capacity has increased, so as to increase the period of time between drops as much as possible. Multiplicative increase would cause a potentially much larger rate of packet loss, incurring more retranmissions.
Multiplicative decrease: is used to react quickly when congestion is detected. Since congestion is detected thru packet loss, the queues along the link have already reached capacity and have begun to overflow, and hence it is critical to back off (send at a slower rate) as quickly as possible. Additive decrease would back off too slowly.
(ii) What is the difference between distance vector, path vector, and link state routing algorithms?
Soln: Distance vector and path vector are ”path oriented”, in the sense that advertisements refer to the state of complete paths. Link state is ”link oriented”.
Soln: Link state suffers from fewer convergence problems (eg count to infinity), making it ideal for intra-domain routing. Path vector and distance vector allow providers to hide information about topologies (eg. which AS's an AS is peered to) making it ideal for inter-domain routing.
(iii) Which floods requests: Chord or Gnutella? Which can execute arbitrary wildcard searches more easily and why?
Soln: Gnutella floods requests. This allows it to more easily execute wildcard searches, since every node receives a copy of the request. (a wildcard search is a search for all files with a substring in the title, for example, all files containing the string “Britney Spears”)
(iv) What is “backscatter” and how can one use it to measure the rate of denial-of-service attacks?
Soln: Denial-of-service attacks send a large stream of data to a particular host under attack. If the hosts sending the attack traffic insert their IP address in the “source IP” field, the host under attack will know who's sending the traffic, and they can then contact the ISP containing the attacker and ask them to disable the attacker's account. However, attackers may mask their location by inserting a false value in the source IP address field.

Some of the packets sent by the attacker may “bounce back” towards the attacker. For example, if the destination host is unplugged and hence becomes unreachable, or if the port under attack gets shut down, the host under attack (or an intermediate router) may start sending packets back towards the source... but if the attacker lied about the source IP in the packet, this packet will get sent to some arbitrary IP address!

Some ISPs and universities that joined the Internet very early own large portions of the IP address space, but may only use a small fraction of it. These entities can obeserve traffic destined to IP addresses that they own that have not been allocated. Packets to these addresses are likely to be the result of “backscatter” from the attacks described above.

(v) How does Traceroute work?
Soln: It sends out an ICMP packet with a fixed TTL. It repeats the process, with increasing TTLs. When the TTL of the packet expires, an ICMP packet will be returned to the source, which contains the address of the machine that last incremented the TTL. By doing this process, Traceroute can learn the IP address of each host along the path to a given destination.

(vi) What is the purpose of SYN-cookies? (you don't have to tell us how they work, just what their purpose is)

Soln: They prevent DoS attacks caused by an attacker sending a large number of SYN packets. As discussed in "Practical UNIX and Internet Security," by Garfinkel and Spafford: The recipient will be left with multiple half-open connections that are occupying limited resources. Usually, these connection requests have forged source addresses that specify nonexistent or unreachable hosts that cannot be contacted. Thus, there is also no way to trace the connections back."
(vii) How does your mail program (or, more accurately, your smtp server) find the right mail server to which to send a piece of email?

Soln: It discovers the MX record for the domain of the destination address. For example if you want to email foo@xyz.com, your smtp server would do a DNS lookup for the MX record of xyz.com.
(viii) Integrated Services (IntServ) proposed two services: guaranteed service and controlled load service. There are also two different kinds of admission control proposed: measurement-based and parameter-based (or worst-case). Which admission control is used with guaranteed service and why? Which admission control is usually used with controlled load service and why?

Soln: Parameter-based is appropriate for guaranteed service, since they are both focused on providing *hard* QoS guarantees. Controlled load service provides softer guarantees, and is hence measurement-based is more appropriate.
(ix) There are two key distinctions among multicast routing algorithms. There is single-source multicast, in which each multicast group has a single sender, and general-source multicast, in which any host can send to the group. Among general-source multicast algorithms, there are shared-tree and source-specific tree routing algorithms. How does single-source multicast build the multicast tree?

Soln: In single-source multicast: hosts send grafts toward the source to join. This fixes the root of the tree at the source.
General-source multicast can be said to offer a "rendezvous" service that single-source does not; what does this mean?
The group can exist when any subset of group members is present. This isn't true for single-source multicast, as the source is fixed and hence must be present for the tree to be formed.

Question 2 (20 pts) ***Potpourri***

Please answer these questions in no more than a few sentences. Focus in the key points, don't worry about filling in all the details.

(a) Architectural questions: **(6 pts)**
(i) Why is reliability implemented at the transport layer (layer 4) and not at the network layer (layer 3)?

Soln: Routers perform operations on IP packets at layer 3. These operations could potentially corrupt packet contents. Hence we need an end-to-end mechanism to catch these errors, which reliability at layer 4 provides. In fact, layer 4 is the lowest layer that can provide end-to-end reliability.
(ii) Why does the Internet only have one protocol at the network layer (IP)?

Soln: Having a single general-purpose forwarding protocol improves modularity. We can implement arbitrary applications on top of IP, and have it run over arbitrary media. With this layering we don't need to re-implement every application for every transmission media technology.
In addition, having a single network layer protocol makes it easy for networks to peer and interoperate.
(iii) Why should designers not embed application knowledge into level 3 (and below) protocols?
Soln: Other applications may not need this knowledge. These applications shouldn't have to suffer maintaining and distributing this knowledge if they don't need to use it.
(b) Transport Protocols: **(8 pts)**
(i) At what layer are UDP and TCP implemented, network (layer 3) or transport (layer 4)?
Soln: Layer 4.
(ii) Which of UDP and TCP would be better for voice-over-IP, and why? Which of UDP and TCP would be better for electronic money transfers, and why?
Soln: UDP is typically preferred over TCP for voice-over-IP, as delaying a packet is typically much worse than simply losing it. That is, TCP ensures recovery of packets, but the recovery mechanism introduces delay (due to ARQ), which is highly undesirable for voice-over-IP (e.g. imagine conversing with someone over the phone. Losing a packet is not a big deal, as you can ask the person to repeat themselves. However, having a fixed delay of a few seconds between when they speak and when you hear them can make conversations unpleasent, as you will often start talking at the same time).
TCP would be preferable for electronic money transfers, as it is highly desirable for transfer requests to get through without being lost, even if they are delayed (e.g. it is ok to delay a request to depost $100 into an account a few seconds, as long as it gets executed eventually).
(iii) Name a field that UDP and TCP have in common, and what is it used for?

Soln: Checksum (detecting bit errors). Also, port (knowing which application to deliver the packet to).
(iv) TCP offers reliability and congestion control. In what way does the congestion control algorithm make use of the reliability algorithm?
Soln: Through ACK's. ACKs are used to signal two things (a) the network is not overly congested, so we can increase our sending speed, and (b) the packet got through successfully, so no need to retransmit it.

(c) 802.11 and Ethernet: **(6 pts)**
(i) 802.11 and Ethernet differ from traditional wired network technologies in that they have "shared media" where collisions lead to lost data. However, 802.11 and Ethernet use very different methods to deal with, or avoid, collisions. Of the three techniques of carrier sense, collision detect, and collision avoidance, each is used by only one of Ethernet and 802.11. Which ones are used by Ethernet? Which ones are used by 802.11?

Soln: Ethernet: carrier sense, collision detect 802.11: carrier sense, collision avoidance

(ii) Explain how the RTS/CTS packet exchange works in 802.11. In particular, what do other nodes do when they hear:

Soln: just an RTS:

If you hear an RTS but not the corresponding CTS, you can send. This may cause interference at the source.

just a CTS:

Keep quiet until the scheduled transmission is over (hear an ACK). In this case you're not hearing the RTS perhaps because you're out of radio range.

both an RTS and the corresponding CTS:

Keep quiet until the scheduled transmission is over (hear an ACK).

(iii) Why does 802.11 use RTS/CTS packets while Ethernet does not?
Soln: Because in 802.11, you may not be able to hear every host that can interfere with a packet you're transmitting. The RTS/CTS exchange allows the receiver to arbitrate its airspace, so as to inform other hosts outside of the sender's radio range that they shouldn't transmit.

Question 3 (15 pts) ***Link-State Routing***

All nodes in a network use link state routing (which uses Dijkstra's algorithm to compute shortest paths). Initially the routing tables are empty. Node A gets the following link state packets.

Node ID	Neighbor node	Cost
B	A	5
B	C	1
B	E	1

Node ID	Neighbor node	Cost
C	A	6
C	B	1

Node ID	Neighbor node	Cost
D	A	1
D	E	2

Node ID	Neighbor node	Cost
E	B	1
E	D	2

(a) **(7 pts)** Draw the network topology.

(b) **(8 pts)** Apply the link state algorithm for node A. Show each step in the table below. Your solution need not use all entries of the table.

Step	start S	d(B),p(B)	d(C),p(C)	d(D),p(D)	d(E),p(E)

Soln:
(a) Construct the network graph.
E-B with cost 1
B-C with cost 1
E-D with cost 2
B-A with cost 5
B-C with cost 1
D-A with cost 1

(b) Apply the link state algorithm for node A. Show each step. (1 pt for each step; 5 pts total)

Step	start S	d(B),p(B)	d(C),p(C)	d(D),p(D)	d(E),p(E)
0	A	5,A	6,A	1,A	infinity
1	AD		6,A		3,D
2	ADE	4,E	6,A		
3	ADEB		5,B		
4	ADEBC				

Question 4 (20 points) ***Metrics + QoS***

Suppose we have two hosts connected via an intermediate router:

S——-R——D

R implements a token bucket to rate limit flows between S and D with buffer size B. Suppose S starts a single flow to D at time t=0.

Link S-R has bandwidth R_1=10Kbps, delay d_1=200ms
Link R-D has bandwidth R_2=5Kbps, delay d_2=200ms

(a) **(6 pts)** First, suppose S sends to D with a constant bit rate UDP stream, so as to saturate the link S-R. What time does the first packet drop occur?

Soln:
Draw the throughput curve. We can see the first packet loss corresponds to the knee in the curve. This happens after $(b * R_1)/(R_1 - R_2)$ bits are sent. The time corresponds to the amount of time it takes to send this many bits, or $(b * R_1)/(R_1 - R_2)/10Kbps$.

(b) **(6 pts)** First, consider a simplified version of TCP: the source S does not advance the window until it receives ACKs for every packet in the window. At what time will the first packet drop occur?

Soln:
The first packet drop occurs when the TCP burst exceeds the size of the buffer, given the queue drain rate. In particular, it occurs at iteration i where $2^i * (packetsize)$ first exceeds $(b * R_1)/(R_1 - R_2)$. Hence, the time T when the packet drop occurs is:

$$T = \sum_{i=0}^{k} i * RTT(i) + i * p/R_1 + (2^i - 1) * p/R_2 \tag{0.1}$$

where

$$k = \log_2 B * R_1/(2 * R_1 - 2 * R_2) \tag{0.2}$$

and

$$RTT(i) = propdelay + storeandforwarddelay = 2 * d + P/R_1 + 2^{i-1}P/R_2 \tag{0.3}$$

(c) **(4 pts)** Next, suppose regular TCP Reno is used. Does this change your answer? If so, what is the time the first packet loss will occur if TCP Reno is used? If not, why not?

Soln:
S can send pkts at the rate of 10kbps, but they will traverse link R-D at the rate 5kbps. So, the ACKs returning to S will be spaced out such that S will never send more than R-D's link capacity. Hence, there will be no packet loss.

(d) **(4 pts)** Suppose instead there are two routers between S and D:
S—–R_a—–R_b—–D, where
link $S - R_a$ has bandwidth 10Kbps, delay 200ms,
link $R_a - R_b$ has bandwidth 5Kbps, delay 200ms,

link $R_b - D$ has bandwidth 10Kbps, delay 200ms.

Suppose R_a has a buffer of size 7, and R_b has a buffer of size 3. Suppose S starts sending to D with a constant bit rate UDP stream, so as to saturate the link $S - R_a$.

(d-i) What time does the first packet loss occur?
(d-ii) Does the first packet loss occur at router R_a or R_b?

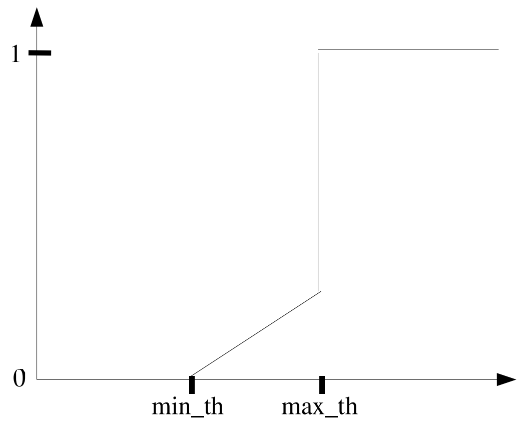
Soln:

(i) We can use the same formula as in part (a), but simply plug in a buffer size of 7 in place of B.

(ii) The first packet loss must occur at router R_a. Packet loss can't occur at R_b since it's output bandwidth exceeds its input bandwidth!

Question 5 (15 points) ***RED***

(a) **(6 pts)**
(a-i) If the traffic consists solely of UDP sources, is it likely that using RED will result in fair bandwidth allocation between flows?
(a-ii) Does RED typically avoid bursts of losses? Why or why not?
(b) **(3 pts)**
Describe how to implement congestion control with Explicit Congestion Notification (ECN) In particular,
(b-i) What modifications (if any) are needed in RED?
(b-ii) What modifications (if any) are needed in the TCP source?
(b-iii) What modifications (if any) are needed in the TCP receiver?
(c) **(6 pts)**

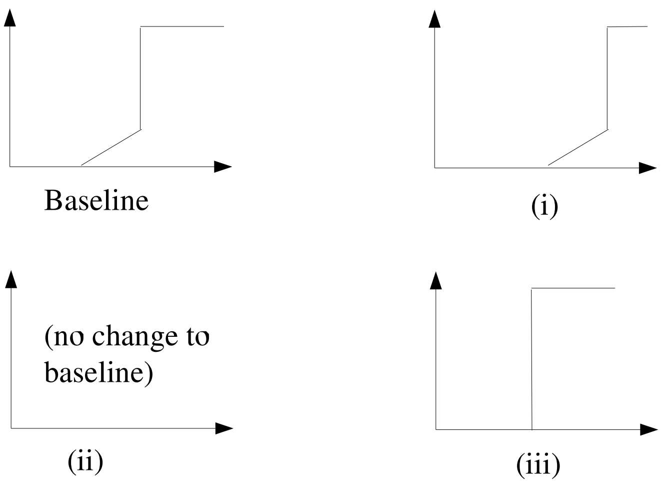


Your task in this problem is to describe how you would configure RED for each of the scenarios listed below. For each scenario, you are allowed only to change ONE of the *parameters* to RED (min_th, max_th, or the filter gain α used in the Exponentially-weighted Moving Average (EWMA) to compute the average queue length). There may be more than one right answer for which parameter to choose. You are not allowed to change the RED algorithm itself.
Make sure to clearly motivate the parameter change you are suggesting, and explain why your change improves performance.
(c-i) How would you configure RED to achieve higher link utilization (by incurring higher average queue length)?
(c-ii) How would you configure RED to make it more responsive to short-term bursts (which give rise to rapid changes in queue length)?
(c-iii) How would you make RED behave as much as possible like Drop-Tail queueing?
Soln:

(a) (i) RED is not appropriate, as UDP flows do not back off after a loss. (ii) RED is appropriate, as it can reduce synchronization and can spread out the effects of bursts by dropping early, before the queue becomes fully utilized.

(b) If the queue is completely full the router drops the packet. If the queue has spare room, RED probabilistically marks the packet. The receiver needs to be modified to forward the notification back to the sender.

The sender needs to be modified to treat the markings as signals of congestion, and back off.



(c)
(i) We increase max_{th} (or min_{th}).
(ii) We change α, the filter gain in the EWMA of the queue size, so as to more heavily weight the measured queue length.
(iii) We set $max_{th} = min_{th}$.

Question 6 (10 points) ***CAN of Worms***

Please choose **ONLY ONE** of the following two questions to do (you do not need to do both questions). Clearly indicate which of the two questions you are answering.

(a) Consider a CAN distributed hash table with N nodes. Assume that the nodes are arranged in a perfect grid. Show that the average path length is roughly $d * n^{(1/d)}$.

Soln: Use the fact that, in the expected sense, these n nodes divide the space equally, and that the volume of a $d-$dimensional hypercube of side s is equal to s^d).

In more detail: start by assuming we have a 2 dimensional CAN. The namespace can be represented by a square (a 2 dimensional grid of nodes approximately arranged as a square). The number of nodes in the network corresponds to the area of the square. To reach a given destination, we need to route at most (i) horizontally the length of the square, then (ii) vertically the height of the square. Each edge of a square of size n is $sqrt(n)$. So, we need to route at most $2 * sqrt(n)$ hops.
We can generalize this to higher dimensions by assuming we have a $d-$dimensional hypercube. We need to traverse (at most) each side of this hypercube. Each side is of length $n^{(1/d)}$, and there are d total sides. So, the average expected path length is roughly $d * n^{(1/d)}$.

(b) Consider the process of a worm spreading in a network with N nodes. The worm spreads in rounds. In each round, every infected node contacts K other random nodes in the network (note that some of these nodes may have been already infected), and infects them. Assume that in round 0 there is only one node infected.
Let m be the round by which half of the nodes are infected, i.e., after round m, the expected number of infected nodes is $N/2$. Compute an approximate upper and lower bound for the value of m.
Solution (b):
Let n_i be the number of infected nodes after round i, where $n_0 = 1$. Then the expect number of infected nodes after round $i + 1$ is

$$n_{i+1} = n_i + n_i * k * (n - n_i)/n, \tag{0.4}$$

where the second term represents the expected number of newly infected nodes. Each infected node in round i contacts k random nodes, out of which only a fraction $(n - n_i)/n$ are not already infected. (NOTE: There is an inaccuracy here as we ignore nodes that are infected by more than one nodes during round $i + 1$, but this shouldn't matter).
Because $n_i < N/2$, Eq. (1) can be trivially bounded as:

$$n_i + n_i * k/2 < n_{i+1} < n_i + n_i * k \rightarrow \tag{0.5}$$

$$n_i * (1 + k/2) < n_{i+1} < n_i * (1 + k) \tag{0.6}$$

This means that it will take at most m_1 rounds, and at least m_2 rounds, to infect $N/2$ nodes, where

$$n_0 * (1 + k/2)^m 1 = N/2 => (1 + k/2)^m 1 => m1 = log_{1+k/2}(N/2), and \tag{0.7}$$

$$n_0 * (1 + k)^m 2 = N/2 => (1 + k)^m 2 => m2 = log_{1+k}(N/2). \tag{0.8}$$

Thus, we obtain

$$log_{1+k}(N/2) < m < log_{1+k/2}(N/2). \tag{0.9}$$

Question 7 ***Extra Credit Question (redeemable for a chocolate truffle)***

Why is TCP congesion control like Blanche Dubois?
Soln:
Because it “relies on the kindness of strangers.”