

Name: _____

username: _____

EECS 489 Winter 2010 Midterm Exam

This is an open-book, open-resources exam. Explain or show your work for each question. Your grade will be severely deducted if you don't show your work, even if you get the correct answer.

Name: _____

username: _____

Honor code: I have neither given nor received any help on this exam.

Signature: _____

Problem	Points
1	/15
2	/15
3	/15
4	/20
5	/15
Total	/80

Name: _____

username: _____

QUESTION 1: Big MAC (15 points)

This question explores whether the Internet could be designed using only names and MAC addresses, without a need for IP addresses. Suppose each network adapter, for any link technology, has a unique MAC address from a single address space (such as 48-bit MAC addresses used for Ethernet devices), and these addresses were used Internet-wide for communication between end hosts. Suppose that the Domain Name System (DNS) is changed to return MAC addresses rather than IP addresses, in response to queries.

Make sure you explain your answers below.

(a) In today's Internet, why is it difficult to support continuous communication with a host while it moves? Why would an Internet based on MAC addresses have the potential to make mobile hosts easier to support?

Mobility is difficult to support in today's Internet because a host's IP address changes as it moves, since IP addresses are allocated to institutions in a hierarchical fashion. Dynamic changes in the host's IP address make it difficult to sustain a connection (e.g., TCP or UDP socket) with another host. An Internet based on MAC addresses would allow the host's identifier to stay the same as it moves.

(b) Today's local area networks have several "boot-strapping" techniques, such as DHCP, ARP, and MAC learning. In a world without IP addresses, which of these techniques would still be necessary?

DHCP would no longer be needed to assign an IP address or network mask to the host, though a similar mechanism may still be needed to configure the address of the local DNS server and the gateway router. ARP would no longer be necessary, as there would be no need to map an IP address to a MAC address. MAC learning would still be necessary to enable switches to learn how to forward frames to a host based on its MAC address.

(c) How would moving to an Internet based on MAC addresses affect the size of the forwarding tables in the network nodes (i.e., routers or switches)?

Forwarding tables could grow quite large because MAC addresses are not hierarchical. Today, routers store ~200K IP prefixes that represent a much larger number of hosts. Within a local area network, switch forwarding tables could likely stay the same size, as hosts could direct external traffic (destined to the rest of the Internet) to the gateway router.

(d) How would the new design, based on MAC address, affect users' privacy?

Name: _____

username: _____

Hosts throughout the Internet would know the MAC address of the machine sending the traffic (e.g., to a Web site), leading to less privacy compared to IP addresses. Yet, the remote machine would not necessarily know where the host is located, which may improve privacy.

Name: _____

username: _____

QUESTION 2. A Rose by Any Other Name (15 points)

This question explores the challenges of the Domain Name System.

(a) Before the 9/11 attacks, the top-level domain (TLD) server for South Africa was located in New York City. Explain why the physical destruction on 9/11 disrupted Internet communication *within* South Africa (e.g., for a Web user in South Africa accessing a Web site in South Africa)?

Web transfers within South Africa rely on DNS to resolve domain names for South African Web site names (e.g., www.gov.za). The Web user's local DNS server needs to contact the TLD server for the .za domain as part of translating the Web server name into an IP address, making the Web download fail because the DNS information is not available.

(b) Following up on question (a), explain why the effects in South Africa took place *gradually*, disrupting progressively more communication within the country in the hours (and even days) after connectivity to NYC was lost.

The local DNS servers in South Africa presumably had many name-to-address mappings cached already, particularly for popular sites. However, these cached entries had a time-to-live field that eventually expired, causing the local DNS servers to evict the expired entries. Once the expired entries were evicted, future requests from end hosts would fail due to the inability to contact the TLD server.

(c) How do the local DNS servers know the identity of the root servers? Why are most of the host-to-address queries seen by the root DNS servers for *bogus* or *malformed* names (like **graceland.elvis** or numeric top-level domains)?

The local DNS servers are statically configured with the identity of the root servers.

Most queries do not require the local DNS server to contact the root servers, since typically the TLD servers for common TLDs like .com, .net, .edu, etc. are already cached (and have larger time-to-live values). Bogus or malformed names do not correspond to valid TLDs and, as such, are typically not cached, forcing requests to go to a root server.

(d) Who determines the value of the time-to-live field that determines how long DNS servers cache a name-to-address mapping? What are the pros and cons of using a small value?

The operator of the responding DNS server (e.g., the authoritative DNS server) assigns the TTL value. The advantages of a small TTL are: (i) rapid failover if the IP address associated with a name changes and (ii) enabling content distribution networks to exert fine-grain control (e.g., for load balancing) over which Web server replica handles the Web client request. The disadvantages of a small TTL are: (i) extra DNS requests (which place extra load on the network and the DNS servers) and (ii) extra latency for the user to wait for these DNS queries to complete successfully.

Name: _____

username: _____

(e) A local DNS server typically discards cached name-to-address mappings when the time-to-live expires. Alternatively, the local DNS server could optimistically issue a new query for the cached domain name. Given one advantage and one disadvantage of that approach.

By “prefetching” the name-to-address mapping, the local DNS server can hide the DNS look-up delay, improving the performance experienced by the end user. However, prefetching introduces extra DNS queries and network load to look up name-to-address mappings that may never be needed.

Name: _____

username: _____

QUESTION 3. Finding Your Way (15 points)

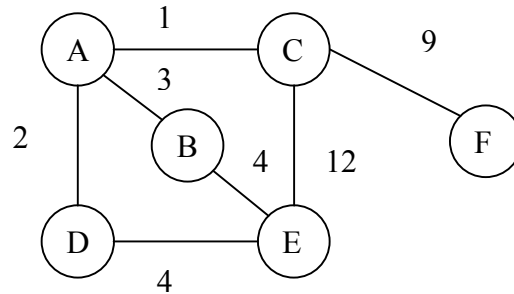


Fig. 3. Yet Another LAN.

(a) Show how the Dijkstra's shortest path first algorithm builds the routing table for node D.

Show the modification to D's table in the intermediate steps as done in lecture. Showing just the table of D without any intermediate steps will not be given full credit.

Step	Set S	A	B	C	D	E	F
0	{}	∞	∞	∞	∞	∞	∞
1	{D}	2	∞	∞	0	4	∞
2	{D,A}	2	5	3	0	4	∞
3	{D,A,C}	2	5	3	0	4	12
4	{D,A,C,E}	2	5	3	0	4	12
5	{D,A,C,E,B}	2	5	3	0	4	12

Name: _____

username: _____

(b) Which routing protocol uses Dijkstra's shortest path first algorithm for route computation: is it inter-domain routing or intra-domain routing? is it link-state routing protocol or distance-vector routing protocol?

Intra-domain. Link-state protocol.

Name: _____

username: _____

QUESTION 4. Dealing with Collisions (20 points)

Suppose the round-trip propagation delay for Ethernet is 50 μ seconds, for a bandwidth of 10Mbps. Assume the minimum frame size is T .

- a. Why is it necessary to have a minimum frame size?

So, that collision can be detected before frame is finished transmission.

- b. What should the minimum frame size be, in terms of T , if the propagation delay is held constant and the bandwidth is increased to 100Mbps? Explain your answer.

Since the bandwidth is increased 10 times, the minimum frame size also increases by 10 folds. It is now $10T$.

- c. Give one drawback of this frame size.

With such a large frame size, smaller packets become very inefficient and actual data is being transmitted a smaller percentage of the time.

Suppose 2 nodes A and B are attached to the opposite ends of a 10Mbps Ethernet, and the propagation delay between the two nodes is 325 bit times (One *bit time* is the duration to transmit one bit, i.e., for 10Mbps links, it's 10^{-7} sec). Suppose node A begins transmitting a frame (of 512+64 bit times), and before A finishes, node B begins transmitting a frame.

- a. Can A finish transmitting before it detects that B has transmitted? Why or why not?

At $t=0$, A transmits. At $t=576$, A would finish transmitting. In the worst case, B begins transmitting at time $t=324$, which is the time right before the first bit of A's frame arrives at B. At time $t=324+325=649$ B's first bit arrives at A. Because $649 > 576$, A finishes transmitting before it detects that B has transmitted. So A incorrectly thinks that its frame was successfully transmitted without a collision.

- b. If A finishes transmitting before it detects that B has transmitted, what is the implication? How does one fix this problem?

A incorrectly thinks that its frame was successfully transmitted without a collision. To fix the problem, the minimum frame size needs to be increased or the maximum propagation delay needs to be constrained by restricting the maximum length of the Ethernet cable.

Name: _____

username: _____

PROBLEM 5. For Good Measure (15 points)

(a) Traceroute capitalizes on the “Time To Live” field in the IP packet header to measure the forwarding path from one host to another. Give two reasons why a hop in Traceroute might show a “*” (i.e., no IP address for the router at that hop in the path).

The packet may have been lost on the forward path.

The TIME_EXCEEDED packet may have been lost on the reverse path.

The router may have been configured not to send TIME_EXCEEDED messages.

The packet may have encountered a firewall or NAT that drops the packet.

(b) Traceroute may report a path that does not exist. For example, Traceroute could return a path A-B-C (where A, B, and C are IP addresses) when there is no link between routers B and C. Give one example where that could happen. (Draw a picture if useful.)

The route to the remote destination may have changed during the measurement process.

For example, suppose the route from A to Z changed from A-B-Y-Z to A-D-C-Z. Then, the first traceroute probe would return “B” and the second would return “C”, even though nodes B and C are not directly connected.

(c) The delay between a TCP sender transmitting a SYN packet and receiving the SYN-ACK packet provides an estimate of the round-trip time between a client and a server. Give one reason why, on average, the SYN/SYN-ACK delay is *larger* than a typical round-trip delay between a client and server. Assume all DNS and ARP look-ups have already completed.

The server may have a listen queue that stores pending requests to establish a TCP connection. As such, the SYN packet may sit in the queue for some time before the SYN-ACK is sent.

(d) Determining the geographic location of an Internet host, based on its IP address, is challenging. Propose two techniques that could be used to infer the rough location.

The DNS name, found by performing an address-to-name look-up, may reveal the location, e.g., “nslookup(12.122.16.158)” returns “tbr1.n54ny.ip.att.net”, which is at 54th street in Manhattan.

The whois database may reveal the location, e.g., “whois -h whois.arin.net 128.112.12.58” returns an entry with 87 Prospect Avenue in Princeton, New Jersey.

Sending probes (such as “ping”) from a host can reveal the round-trip latency to the remote destination. Sending such probes from many vantage points can be used to “triangulate” the location of the remote host.

Name: _____

username: _____

Sending a traceroute to the remote host would reveal the names of many of the routers along the path. The names of the routers near the destination would reveal the likely location of the host.

(e) Suppose a host's *local DNS server* records the round-trip time for a successful query to an *authoritative DNS server*. This could be used as an estimate of the expected round-trip time for the host to communicate with machines at the remote site. Give one reason why this is a good heuristic for estimating the round-trip time between two hosts, and one reason why it is not.

A host is typically close to its local DNS server (usually learned via DHCP). Similarly, a host is typically close to its authoritative DNS server (which is configured with the host's name-to-address mapping). As such, the round-trip time from host A's local DNS server to host B's authoritative DNS server is often a reasonable estimate of the round-trip time from A to B.

However, the estimate may be quite inaccurate if either of the assumptions is violated, e.g., because host A is configured to use a far-away local DNS server.