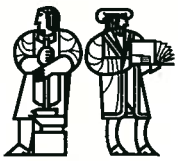


Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2004

Quiz II

There are 13 questions and 9 pages in this quiz booklet. To receive credit for a question, answer it according to the instructions given. You have **50 minutes** to answer the questions.

Write your name on this cover sheet AND at the bottom of each page of this booklet.

Some questions may be harder than others. Attack them in the order that allows you to make the most progress. If you find a question ambiguous, be sure to write down any assumptions you make. Be neat. If we can't understand your answer, we can't give you credit!

THIS IS AN OPEN BOOK, OPEN NOTES QUIZ.
NO PHONES, NO LAPTOPS, NO PDAS, ETC.

CIRCLE your recitation section number:

10:00	1. Ernst/Strauss	2. Madden/Hickey	
11:00	3. Dabek/Hickey	4. Ernst/Chen	5. Madden/Strauss
12:00	6. Dabek/Chen		
1:00	7. Katabi/Bicket	8. Saltzer/Garfinkel	9. Karger/Lesniewski
2:00	10. Saltzer/Bicket	11. Katabi/Lesniewski	12. Karger/Garfinkel

Do not write in the boxes below

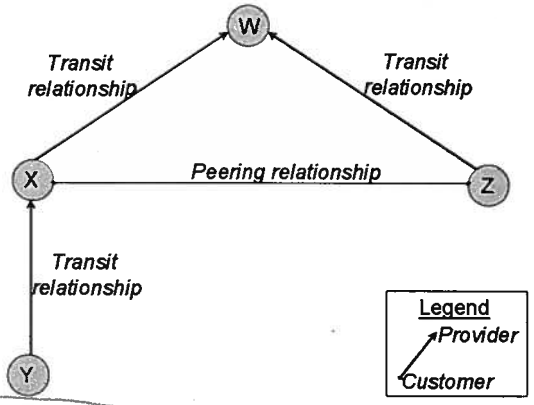
1-4 (xx/32)	5-7 (xx/18)	8-9 (xx/16)	10-11 (xx/16)	12-13 (xx/18)	Total (xx/100)
24	18	15	+12 SAM	13	82.5
42	57	69	82		

Name: DAN PORTS

I Real-world systems

1. [8 points]: The figure below shows four interconnected autonomous systems (AS's) in the Internet, W, X, Y, and Z. For each AS i, let R_i denote the routes for destinations in AS i. If an AS i advertises a route for some destination to an AS j, then we say that "j hears the route from i." Each AS correctly implements the route advertisement and import rules described in the paper, "An Introduction to Wide-Area Internet Routing" (reading #9). Which of the following statements is true?

(Circle ALL that apply)



A. Z will hear R_Y from both X and W.

B. W will hear R_Y from both X and Z.

C. When X gets a packet destined for Z, it will send the packet on the peering link if the packet's source is in X, but will not forward it on the peering link if the packet's source is in Y.

D. If the $W - X$ link fails, then W will not be able to use the path through Z and X to reach Y.

2. [8 points]: Ben Bitdiddle implements a NAT box according to the description in "The IP Network Address Translator (NAT)." Whenever Ben's NAT box sees a packet from a local address, it replaces the source address with one of the box's available global addresses and records the local address to global address mapping. For some reason, Ben is unable to properly use certain TCP-based client applications from behind the NAT box. Which of these reasons is a good explanation for the problem?

(Circle ALL that apply)

A. The client application might be sending its IP address in the payload for the server to process.

B. The server application might be sending its IP address in the payload for the client to process.

C. Ben's client is trying to communicate with a server that is behind the same NAT box, and the NAT does not know how to forward those packets.

D. Ben has forgotten to modify the Ethernet CRC sequence in the NAT after adjusting the IP source address, so packets are being dropped by the switch at the other end of Ben's NAT box.

↳ This is possible, but if so the entire NAT would cease to operate; not just certain TCP apps.

Name: DAN PORTS

3. [8 points]: Which of the following points follows from Ken Thompson's arguments in "Reflections on Trusting Trust" (reading #12)?

(Circle all that apply)

- **A.** It is difficult to discover malicious code in programs, even if you have access to the source code to the entire system, including the compiler.
- **B.** If a C compiler written by someone trustworthy, and all the other system programs are compiled with that compiler, then there can be no malicious behavior.
- **C.** The C programming language is an evil language because it makes it easy to write Trojan horses.
- **D.** It is possible to write a compiler for the C programming language in C.
- **E.** Self-reproducing programs are more important and worthy of a Turing award than UNIX.

4. [8 points]: Which of the following points follows from Ross Anderson's arguments in "Why Cyptosystems Fail" (reading #13)?

(Circle ALL that apply)

- **A.** Cryptographic protocols are the weakest link in practical systems where security is important.
- **B.** Getting the cryptography right is important, but not sufficient to achieve security.
- **C.** Insider attacks are a significant security problem in many practical systems.
- **D.** Secure systems should not be designed like safety critical systems.

Name: DAN PORTS

II The OttoNet

Inspired by the recent political success of his Austrian compatriot, "Arnie," in Caleeforneea, Otto Pilot decides to emigrate to Boston. After several months, he finds the local accent impenetrable, and the local politics extremely murky, but what really irks him are the traffic nightmares and long driving delays in the area.

After some research, he concludes that the traffic problems can be alleviated if cars were able to discover up-to-date information about traffic conditions at any specified location, and use this information as input to software that can dynamically suggest good paths to use to go from one place to another. He jettisons his fledgling political career to start a company whose modest goal is to solve Boston's traffic problems.

After talking to car manufacturers, Otto determines the following:

1. All cars have an on-board computer on which he can install his software. All cars have a variety of sensors that can be processed in the car to provide traffic status, including current traffic speed, traffic density, evidence of accidents, construction delays, etc.

2. It is easy to equip a car with a Global Positioning System (GPS) receiver (in fact, an increasing number of cars already have one built-in). With GPS, software in the car can determine the car's location in a well-known coordinate system (assume that the location information is sufficiently precise for the purposes of this quiz).

3. Each car's computer can be networked using an inexpensive 10 Megabits/second radio. You may assume that each radio has a spherical range, R, of 250 meters; *i.e.*, assume that a radio transmission from a car has a non-zero probability of directly reaching any other car within 250 meters, and no chance of directly reaching any car outside that range.

Otto sets out to design the *OttoNet*, a network system to provide traffic status information to applications. OttoNet is an *ad hoc* wireless network formed by cars communicating with each other using cheap radios, cooperatively forwarding packets for one another.

Each car in OttoNet has a client application and a server application running on its computer. OttoNet provides two functions that run on every car, which the client and server applications can use (read these specifications carefully!):

1. QUERY(*location*): When the client application running on a car calls QUERY(*location*), OttoNet delivers a *query* packet to at least one car within distance R (the radio range) of the specified location, according to a best-effort contract. The query packet is 1,000 bits in size.

2. RESPOND(*status_info, query_pkt*): When the server application running on a car receives a query packet, it processes the query and calls RESPOND(*status_info, query_packet*). RESPOND() causes a *response* packet to be delivered to the client that performed the query, according to a best-effort contract. The response packet summarizes local traffic information (*status_info*) collected from the car's sensors and is 10,000 bits in size.

For both query and response packets, the cars will forward the packet cooperatively in best-effort fashion toward the desired destination location or car. Cars may move arbitrarily, alternating between motion and rest. The maximum speed of a car is 30 meters/second (108 kilometers/hour or 67.5 miles/hour).

Name: DAN PORTS

5. [8 points]: Which of the following properties is true of the OttoNet, *as described thus far*?
(Circle ALL that apply)

+8

A. Because the OttoNet is "best-effort," it will attempt to deliver query and response packets between client and server cars, but packets may be lost and may arrive out-of-order.

B. Because the OttoNet is "best-effort," it will ensure that as long as there is some uncongested path between the client and server cars, query and response packets will be successfully delivered between them.

C. Because the OttoNet is "best-effort," it makes no guarantees on the delay encountered by a query or response packet before it reaches the intended destination.

D. An OttoNet client may receive multiple responses to a query, even if no packet retransmissions occur in the system.

Otto develops the following packet format for OttoNet (all fields except `payload` are part of the packet header):

```
structure packet {
    GPS dst_loc;          // intended destination location
    int_128 dst_id;       // car's 128-bit unique ID picked at random
    GPS src_loc;          // location of car where packet originated
    int_128 src_id;       // unique ID of car where packet originated
    int hop_limit;        // number of hops remaining (initialized to 100)
    int type;             // query or response
    int size;             // size of packet
    char *payload;        // query request string or response status info
};

structure packet pkt; // pkt is an instance of ''structure packet''
```

Each car has a 128-bit unique ID, picked entirely at random. Each car's current location is given by its GPS coordinates. If the sender application does not know the intended receiver's unique ID, it sets the `dst_id` field to 0 (no valid car has an ID of 0).

The function FORWARD(`pkt`) runs in each car, and is called whenever a packet arrives from the network or when a packet needs to be sent by the application. FORWARD() maintains a table of the cars within radio range (R) and their locations (using broadcasts every second to determine the locations of neighboring cars), and implements the following steps:

F1. If the car's ID is `pkt.dst_id`, then deliver to application (using `pkt.type` to identify whether the packet should be delivered to the client or server application), and stop forwarding the packet.

F2. If the car is within R of `pkt.dst_loc` and `pkt.type` is "query", then deliver to server application, and forward to any one neighbor that is even closer to `dst_loc`.

F3. *Geographic forwarding step:* If neither F1 nor F2 is applicable, then among the cars that are closer to `pkt.dst_loc`, forward the packet to some car that is closer in distance to `pkt.dst_loc`. If no such car exists, drop the packet.

Name: DAN PORTS

The OttoNet's QUERY(*location*) and RESPOND(*status_info, query_packet*) functions have the following pseudocode:

```
procedure QUERY(location)
  pkt.dst_loc = location;
  pkt.dst_id = X; //see question 6.
  pkt.src_loc = my_gps;
  pkt.src_id = my_id;
  pkt.payload = "What's the traffic status near you?";
  send(pkt)

procedure RESPOND(status_info, query_pkt)
  pkt.dst_loc = query_pkt.src_loc;
  pkt.dst_id = Y; //see question 6.
  pkt.src_loc = my_gps;
  pkt.src_id = my_id;
  pkt.payload = "My traffic status is: " + status_info   // "+" concatenates strings
  send(pkt);
```

6. [4 points]: Give suitable values for **X** and **Y** in the pseudo-code above, such that the pseudo-code conforms to the specification of QUERY() and RESPOND() given earlier.

(Fill in the blanks)

+4

X = 0

Y = query_pkt.src_id

7. [6 points]: What kinds of names are the ID and the GPS location used in the OttoNet packets?

(Circle ALL that apply)

+6

X **A.** The ID and GPS location are both pure names because either of them uniquely identifies a car.

✓ **B.** On response packets, the destination car's ID (`pkt.dst_id`) is a pure name; the destination car's GPS location is an address.

X **C.** On response packets, the destination car's ID and its GPS location are both addresses.

Name: DAN PORTS

8

8. [8 points]: Otto outsources the implementation of the OttoNet according to these ideas and finds that there are times when a QUERY() gets no response packet, and times when a receiver receives packets that are corrupted. Which of the following mechanisms is an example of an application of an *end-to-end* technique to cope with these problems?

(Circle ALL that apply)

√ **A.** Upon not receiving a response for a QUERY(), retry the QUERY() from the client after a timeout.

B. If FORWARD() fails to deliver a packet because no neighboring car is closer to the destination, store the packet at that car and deliver it to a closer neighboring car a little while later.

√ **C.** Implement a checksum in the client and server applications to verify if a packet has been corrupted.

D. Run distinct TCP connections between each pair of cars along the path between a client and server to ensure reliable end-to-end packet delivery.

9. [8 points]: Suppose Otto decides to retry queries that don't receive a response. The speed of the radio in each car is 10 Megabits/second, and the response and request sizes are 10,000 bits and 1,000 bits respectively. The car's computer is involved in both processing the packet, which takes 0.1 μs per bit, and in transmitting it out on the radio (*i.e.*, there's no pipelining of packet processing and transmission). Assume that each car's radio can transmit and receive packets at the same time.

Suppose that the maximum queue size is 4 packets in each car, the maximum radio range for a single hop is 250 meters, and that the maximum possible number of hops in OttoNet is 100. Ignore media access protocol delays. Assume that the server application takes negligible time to process a request and generate a response to be sent.

What is the smallest "safe" timeout setting that ensures that the retry of a query will happen *only* when the original query or response packet is guaranteed not to still be in transit in the network? *Answer this question in the space provided below, showing your work. Be neat and legible!*

7

Need to find maximum transmission delay on network.

= request_xmit_delay + response_xmit_delay

= (100 hops · (1000 bits / 10 Mbit/s + 1000 bits · 0.1 µs) · 4)

+ (100 hops · (10,000 bits / 10 Mbit/s + 10000 bits · 0.1 µs))

× because not worst case

The multiplicative factors of 4 are due to the queue length of 4: 3 other packets could be sent first at each hop, then this packet. See note on back.

+ (3 × 100 · 1.2 ms) + (3 × 100 (2 ms))

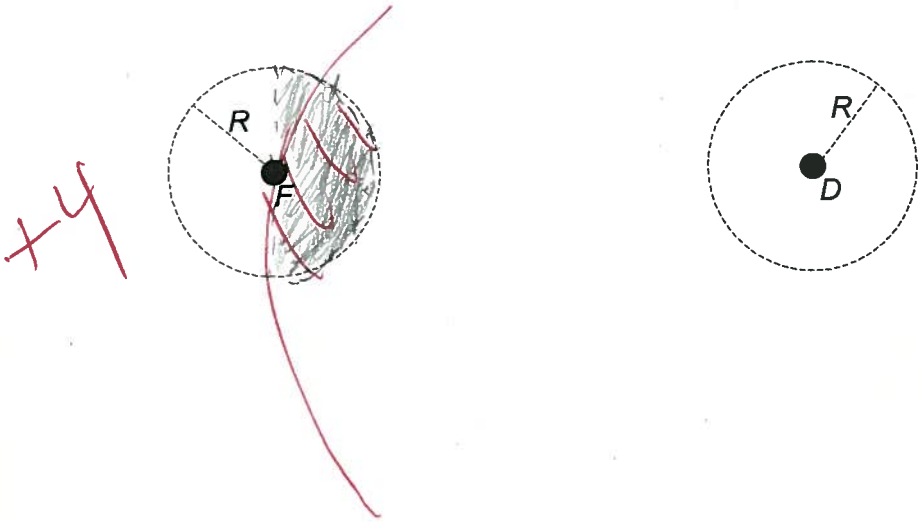
= 1.88 s

Name: DAN PORTS

Otto now proceeds to investigate why FORWARD() sometimes has to drop a packet between a client and server, even though it appears that there is a sequence of nodes forming a path between them. The problem is that geographic forwarding does not always work, in that a car may have to drop a packet (rule F3) even though there is some path to the destination present in the network.

10. [8 points]: In the figure below, suppose the car at F is successfully able to forward a packet destined to location D using rule F3 via some neighbor, N. Assuming that neither F or N has moved, clearly mark the region in the figure where N must be located. (Assume a two-dimensional network.)

Warning: Be reasonably sure of your answer before drawing it on this picture. Use a blank side for scratch space. Or use a pencil so you can erase it and redraw if you make a mistake. If you have messed the picture up, please draw another on a blank side and tell us which page it's on!



+4

11. [8 points]: Otto decides to modify the client software to make "pipelined" QUERY() calls in quick succession, sending a query before it gets a response to an earlier one. The client now needs to match each response it receives with the corresponding query. Which of these statements is correct?

(Circle ALL that apply)

X **A.** As long as no two pipelined queries are addressed to the same destination location (the `dst_loc` field in the OttoNet header), the client can correctly identify the specific query that caused any given response it receives.

✓ **B.** Suppose the OttoNet packet header includes a nonce set by the client, and the server acknowledges the nonce in its response, and the client maintains state to match nonces to queries. This approach can always correctly match a response to a query, including when two pipelined queries are sent to the same destination location.

+8

X **C.** *Both* the client and the server need to set nonces that the other side acknowledges (*i.e.*, both sides need to implement the mechanism in choice B above), to ensure that a response can always be correctly matched to the corresponding query.

D. None of the above.

Name: DAN PORTS

12. [8 points]: After running the OttoNet for a few days, Otto notices that network congestion occasionally causes a congestion collapse because too many packets are sent into the network, only to be dropped before reaching the eventual destination. These packets consume valuable resources. Which of the following techniques is likely to reduce the likelihood of a congestion collapse?

(Circle ALL that apply)

- **A.** Increase the size of the queue in each car from 4 packets to 8 packets.
- **B.** Use exponential backoff in the timeout mechanism while retrying queries.
- **C.** If a query is not answered within a timeout interval, multiplicatively reduce the maximum rate at which the client application sends OttoNet query packets.
- **D.** Use a flow control window at each receiver to prevent buffer overruns.

13. [10 points]: The OttoNet is not a secure system. Otto has an idea—he observes that the 128-bit unique ID of a car can be set to be the public key of the car! He proposes the following protocol. On a query packet, sign the packet with the client car's private key. On a response packet, seal the packet with the client car's public key (that public key is in the query packet). To allow the response packets to be forwarded through the network, the server does not seal the destination location and ID fields of those packets. Assume that each car's private key is not compromised.

Which of these statements is true of this scheme?

(Circle ALL that apply)

- **A.** A car that just forwards a query packet can read that packet's payload and verify it.
- **B.** The only car in the network that can unseal the response packet from a server is the car specified in the destination field.
- **C.** The client cannot always verify the message integrity of a response packet, even though it is sealed.
- **D.** If every server at some queried location is honest and not compromised, the client can be sure that a sealed response it receives for a query actually contains the correct traffic status information.
- **E.** None of the above.

End of Quiz II

Name: DAN PORTS