Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2003

Handout 26 - Quiz II

All problems on this quiz are multiple-choice questions. In order to receive credit you must fill in the blank(s) or mark the correct answer or answers for each question. You have 50 minutes to answer this quiz.

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

Some questions may be much harder than others. Read them all through first and 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 LAPTOP, NO PDAS, ETC.

CIRCLE your recitation section number:

10:00	1. Lynch/Muthitacharoen	10. Horn/Bodhi	
11:00	2. Lynch/Li	6. Nagpal/Bodhi	11. Horn/Muthitacharoen
12:00	5. Nagpal/Li		
1:00	3. Jackson/Krevat	7. Saltzer/Dabek	8. Rinard/Laas
2:00	4. Jackson/Dabek	9. Saltzer/Laas	12. Rinard/Krevat

Do not write in the boxes below

1-4 (xx/35)	5-9 (xx/40)	10-12 (xx/25)	Total (xx/100)

Name:

I Reading questions

1. [10 points]: Which of the following describe activities that the Google system, as described in reading #12, performs as part of its processing of a particular client-submitted query?

(Circle ALL that apply)

A. Google executes the PageRank algorithm to find a ranking of pages pertinent for that particular query.

B. Pages get indexed in Barrels, in order of identifiers for the words appearing in the query.

C. Google uses proximity of different query words on a page to improve precision.

D. A Web Crawler collects pages containing the query keywords and puts them into a repository.

E. Google extracts lists of occurrences of the query words from pages and organizes them into ”hit lists”.

F. Pages containing all the query words in the client query are found from lists in barrels.

2. [10 points]: How can one be sure a compiler does not have a Trojan horse as described in ”Trusting trust” (reading #14)?

(Circle ALL that apply)

A. Check the source code of the compiler line by line manually

B. Check the signature that came with the compiler to verify who produced the compiler

C. Run the compiler in a ”sand box” separated from the network and other files

D. Use a theorem proving system on the source code of the compiler

E. None of the above

3. [10 points]: The network file system (NFS), reading #11, adds a generation number to a UNIX inode to

(Circle ALL that apply)

A. allow the server to reuse an inode

B. guarantee at-most-once semantics for `write` RPCs

C. avoid tracking the file handles that a server has handed out to clients

D. delay writing modified data to disk till after sending a reply on `write` RPC

4. [5 points]: On a machine configured with the default DNS domain "mit.edu", what is the context for the DNS name "ai."?

(Circle best answer)

A. mit.edu

B. the DNS root

C. ai.mit.edu

D. ai

II Monitoring ants

Alice has learned that ants are a serious problem in the dorms. To monitor the ant population she uses the motes of design project I. She spreads hundreds of motes in her dorm. Each mote can transmit a packet to another mote that is within its radio range. Motes forward messages on behalf of other motes to form a network that covers the whole dorm. The exact details of how this network of motes works are the topic of this question.

Each mote runs a small program that every 1 millisecond senses if there are ants present. Each time the program senses ants, it increments a counter, called *SensorCount*. Every 16 milliseconds the program sends the value of *SensorCount* and the mote's ID to the mote connected to Alice's desktop computer, which has ID A. After sending the message, the mote resets *SensorCount*.

For the rest of the quiz you should assume that only the radio consumes energy. The radio operates at a speed of 19.2 Kbits/s (a 916.50 Mhz transceiver). For transmitting the radio draws 12 milliamp at 3V. Although receiving draws 4.5 millamp, you may assume for this quiz that receiving is for free. The motes have a battery rated at 575 milliamphour (mAh).

5. [5 points]: If a mote transmits continuously, about how many hours can a mote transmit for?
(Circle the best answer)

A. 48 hours

B. 16 hours

C. 192 hours

D. 575 hours

6. [5 points]: How much energy (V x I x Time) does it take to transmit a single bit (1 watt-second = 1 joule)?

(Circle the best answer)

A. 36 millijoule

B. 1.9 microjoule

C. 1.9 millijoule

D. 36 microjoule

Because the radio range of the motes is only ten meters, the motes must cooperate to form a network that covers Alice's dorm. Motes forward messages on behalf of other motes to provide connectivity to A. To allow the motes to find paths and to adapt to changes in the network (e.g., motes failing because their batteries run down), the motes run a routing protocol. Alice has adopted the path-vector routing protocol from lecture. Each mote runs the following routing algorithm, which finds paths to mote A:

$n \leftarrow myID()$;
if $n = A$ **then** $path \leftarrow []$;
else $path \leftarrow NULL$;

procedure ADVERTISE()
 if $path \neq NULL$ **then** $transmit($**cons** $(n, path))$

procedure RECEIVE(p)
 if $n \in p$ **then return**
 else if $path = NULL \parallel first(p) = first(path) \parallel length(p) < length(path)$
  then $path \leftarrow p$

procedure TIMER()
 if $haveNotHeardFromRecently(first(path))$ **then** $path = NULL$

When a mote starts it initializes its variables n and $path$. Each mode has a unique ID, which the mote stores in the local variable n. Each mote stores its path to A into the $path$ variable. A path contains a list of mote IDs; the last element of this list is A. The first element of the list ($first(path)$) is the first mote on the path to A. When a mote starts it sets $path$ to NULL, except for Alice's mote, which sets *path* to the empty path ([]).

Every t seconds a mote creates a path that contains its own ID concatenated with the value of $path$, and transmits that path (see *advertise*) using its radio. Motes in radio range may receive this packet. Motes outside radio range will not receive this packet. When a mote receives a routing packet, it invokes the procedure *receive* with the argument set to the path stored in the routing packet. If a path contains n, then this routing packet circled back to n and the procedure *receive* just returns, rejecting the path. Otherwise, *receive* updates $path$ in three cases:

- if $path$ is NULL, because n doesn't have any path to A yet.
- if the first mote on $path$ is the mote from which we are receiving p, because that mote might have a new path to A.
- if p is a shorter path than $path$, because p is a better path to A than $path$. (For this quiz we assume shorter paths are better.)

A mote also has a timer. If the timer goes off, it invokes the procedure *Timer*. If since the last invocation of `Timer` a mote hasn't heard from the mote at the head of the path to A, it resets $path$ to NULL, because apparently the first node on $path$ is not reachable anymore.

Name:

The forwarding protocol uses the paths found by the routing protocol to forward reports from a mote to A. Since A may not be in radio range, a report message may have to be forwarded through multiple motes to reach A. This forwarding process works as follows:

procedure SEND$(counter)$
 if $path \neq NULL$ **then** $transmit(first(path), (n, counter))$

procedure FORWARD$(nexthop, report)$
 if $n \neq nexthop$ **then** $return$
 if $n = A$ **then** $deliver(report)$
 else $transmit(first(path), report)$

The procedure *send* creates a *report* (the mote's ID and the counter value) and transmits a report packet. The report packet contains the first hop on $path$, that is $first(path)$, and the report:

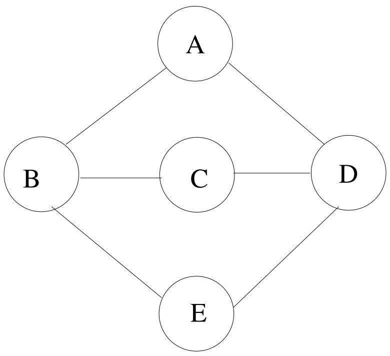
Nexthop	Source	SensorCount

report

The *nexthop* field contains the ID of the mote to which this packet is directed. The *source* field contains the ID of the mote that originated the message. The *sensorcount* field contains the sensor count. (If path is set to NULL, *send* has no mote to forward the report to and thus the mote drops the packet.)

When a mote receives a report packet, it calls *forward*. If a mote receives a report packet for which it is not the next hop, the mote ignores the packet. If a report packet has reached its final destination (A), *forward* delivers it to Alice's computer. Otherwise, the mote forwards the report by setting *nexthop* to the first mote on its $path$ variable. This process repeats until the report reaches A, which delivers the reports to Alice's computer.

For the subsequent questions assume the following arrangement of motes:



The circles represent motes with their node IDs. The edges connect motes that are in radio range of one another. For example, when mote A transmits a packet, it may be received by B and D, but not by C and E. Packets may be lost and motes may also fail (e.g., if their batteries run down). If a mote fails, it just stops sensing, transmitting, and receiving packets.

7. [10 points]: If motes may fail and if packets may be lost, what values could the variable $path$ contain at node E:

(Circle ALL that apply)

A. NULL

B. [B, C]

C. [D, A]

D. [B, A]

E. [B, C, D, A]

F. [C, D, A]

G. [D, A, B, C, D, A]

8. [10 points]: If no motes fail and packets are not lost, what properties hold for Alice's routing and forwarding protocols with the given arrangement of motes?

(Circle ALL that apply)

A. Every mote's $path$ variable will contain a path to A.

B. After the routing algorithm reaches a stable state, every mote's $path$ variable will contain a shortest path (in hops) to A.

C. After the routing algorithm reaches a stable state, the routing algorithm may have constructed a forwarding cycle.

D. After routing algorithm reaches a stable state, the longest path is two hops.

9. [10 points]: If packets may be lost but motes don't fail, what properties hold for Alice's routing and forwarding protocols with the given arrangement of motes?

(Circle ALL that apply)

A. Every mote's $path$ variable will contain a path to A.

B. The routing algorithm may construct forwarding cycles.

C. The routing algorithm may never reach a stable state.

D. When a mote sends a report packet using *send*, the report may or may not reach A.

A report is 13 bits: an 8-bit node ID and a 5-bit counter. A report packet is 21 bits: an 8-bit next hop and a report. Assume that your answer to question 6 (the energy for transmitting one bit) is j joules. Further assume that to start the radio for transmission takes s Joules. Thus, transmitting a packet with r bits takes $s + rj$ joules.

10. [5 points]: Assuming the routing algorithm reaches a stable state, no node and packet failures, how much *total* energy does it take for every node to send one report to A (ignoring routing packets)?

(Circle best answer)

A. $6(s + 21j)$

B. $4(s + 21j)$

C. $8(s + 21j)$

D. $5(s + 21j)$

11. [10 points]: What changes to the Alice's system will reduce the amount of energy needed for each node to send one report to A?

(Circle ALL that apply)

A. Add a nonce to a report so that Alice's computer can detect duplicates

B. Delay forwarding messages and piggyback them on a report message from this mote.

C. Use 4-bit node IDs.

D. Use a stop-and-wait protocol to transmit each report reliably from one mote to the next.

To be able to verify the integrity of the reports, Alice creates for each mote a public-key pair. She manually loads the private key for mote i into mote i, and keeps the corresponding public keys on her computer. A mote signs the contents of the report (the counter and the source) with its private key and then transmits it. Thus, a signed report consists of a 5-bit counter, a node ID, and a signature. When Alice's computer receives a signed report, it verifies the signed report and rejects the ones that don't check out.

12. [10 points]: Assuming that the private keys on the motes are not compromised, and the `sign` and `verify` procedures are correctly implemented, what properties hold for Alice's plan:

(Circle ALL that apply)

A. A mote that forwards a report is not able to read the report's content.

B. A mote that forwards a report may be able to duplicate a report without Alice's computer rejecting the duplicated report.

C. A mote that forwards a report can use its private key to verify the report.

D. When Alice receives a report that verifies correctly, she knows that the report was signed by the mote that is listed in the report and that the report is fresh.

End of Quiz II