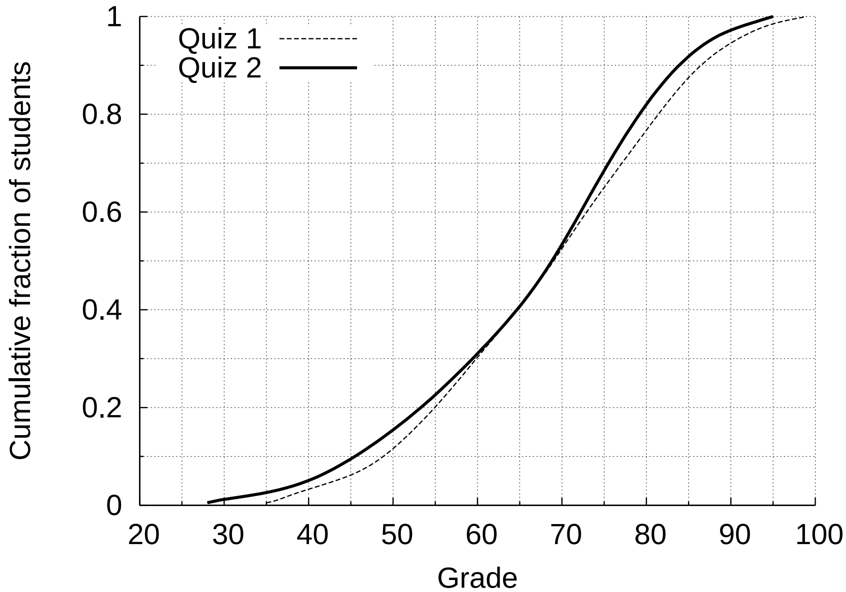
Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2005

Quiz II Solutions



Statistics calculated over all students who took Quiz 2 on April 15, 2005.

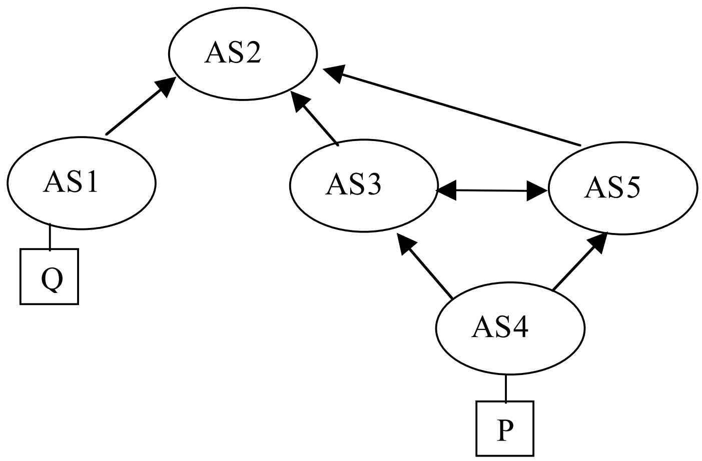
Average: 67.2
Median: 70
Standard deviation: 14.2

Score range	# of students
25–39	7
40-49	17
50-59	30
60-69	37
70-79	59
80-89	32
90-99	6

Name: Alyssa P. Hacker

I Reading Questions

1. [8 points]: The figure below shows 5 autonomous systems (ASes) and their business relationships. The arrows go from customer to provider. Double arrows represent peering relationships. AS1 owns IP prefix Q and AS4 owns IP prefix P. Assume that there are no failures.



Based on the discussion of Internet routing and the Border Gateway Protocol (BGP) described in reading #11, decide if these assertions are true or false.

(Circle True or False for each choice.)

A. True / False AS3 will advertise its route to prefix P to AS2.

TRUE. *An AS advertises customer routes to all other ASes. P belongs to AS3's customer.*

B. True / False AS3 will advertise its route to prefix P to AS5.

TRUE. *An AS advertises customer routes to all other ASes, including those with which it has a peering relationship. P belongs to AS3's customer.*

C. True / False AS3 will advertise its route to prefix Q to AS5.

FALSE. *AS3's route to prefix Q is a route from AS3's provider. AS3 will not advertise that route to any other AS with which it has a peering relationship, such as AS5.*

D. True / False AS4 will hear a route to prefix Q from either AS3 or AS5, but not from both.

FALSE. *In general, AS4 will hear both routes to prefix Q, since both AS3 and AS5 are providers for AS4.*

2. [8 points]: The Network File System (NFS) described in reading #12 allows a client machine to run operations on files that are stored at a remote server. For the version of NFS described in the paper, decide if these assertions true or false.

(Circle True or False for each choice.)

A. True / False When the server responds to a client's write() call, all modifications required by that write will have made it to the server's disk.

TRUE. *Because an NFS server is stateless, it must commit changes to stable storage before returning to the client.*

B. True / False An NFS client might send multiple requests for the same operation to the NFS server.

TRUE. *If a client does not receive a response to a request,* e.g., *because its request was lost, it retransmits it.*

C. True / False NFS requires the underlying RPC protocol to provide "at-most-once" semantics for requests from a client to a server.

FALSE. *NFS clients perform their own retransmissions without relying on any mechanisms for reliable transmission at the lower RPC layer.*

D. True / False When an NFS server crashes and recovers, it must run a recovery procedure from a log to arrive at a consistent state.

FALSE. *NFS servers don't have to run any recovery procedures upon recovering from a crash because the NFS protocol is stateless.*

3. [8 points]: A *port-translating* NAT (NAPT, reading #13) modifies certain fields of the IP and TCP headers. For the two cases below, circle the fields modified by a NAPT box.

(Circle ALL that apply for each case.)

1. Packets *going to* public	A. IP source address	B. IP destination address	C. IP header checksum
(global) Internet	D. TCP source port	E. TCP destination port	F. TCP checksum

ANSWER: *A, C, D, F.*

2. Packets *coming from* public	A. IP source address	B. IP destination address	C. IP header checksum
(global) Internet	D. TCP source port	E. TCP destination port	F. TCP checksum

ANSWER: *B, C, E, F.*

4. [8 points]: While browsing the Web, you click on the link `www.patriots.com`. Your computer asks your Domain Name System (DNS) name server, M, to find an IP address for this domain name. Which of the following is *always* true of the name resolution process, assuming that all name servers are configured correctly and no packets are lost?

(Circle True or False for each choice.)

A. True / False M must contact one of the root name servers to resolve the domain name.

FALSE. *M contacts a root name server only if it has no valid information in its cache for* `www.patriots.com`, `patriots.com`, *or* `com`.

B. True / False M must contact one of the name servers for `patriots.com` to resolve the domain name.

FALSE. *M contacts one of the name servers for* `patriots.com` *only if it has no valid information in its cache for* `www.patriots.com`.

C. True / False If M had answered a query for the IP address corresponding to `www.patriots.com` at some time in the past, then it can respond to the current query without contacting any other name server.

FALSE. *If the binding obtained by M for* `www.patriots.com` *has expired, then M must contact at least one other name server to correctly resolve the domain name.*

D. True / False If M has a valid IP address of a functioning name server for `patriots.com` in its cache, then M will get a response from that name server without any other name servers being contacted.

TRUE. *If M contacts the functioning name server for* `www.patriots.com`, *then it will get an answer without any other name servers being contacted.*

5. [5 points]: You are flying back to MIT from an interview trip to Silicon Valley. Your travel agent gives you the following choice of flights:

A. Flight A uses a plane whose mean time to failure (MTTF) is believed to be 6,000 hours. With this plane, the flight takes 6 hours.

B. Flight B uses a plane whose MTTF is believed to be 5,000 hours. With this plane, the flight takes 5 hours.

The agent assures you that each plane's failures occur according to a memoryless random process (not a "bathtub" curve). Assuming that model, which plane should you take to minimize the chance of your plane failing during the flight.

(Answer in the space below, with brief justification.)

Each plane's failure process is memoryless. The conditional failure probablity, $h(t)$, for plane A is 1/6000 and for plane B is 1/5000. The probability of plane A failing during its flight is 6/6000, and for plane B is 5/5000. In both cases, the probability of failure of a plane during the flight is the same, 1/1000. Therefore, either plane is an equally good choice if you want to minimize the chance of failing during the flight.

II The Wireless EnergyNet

Sara Brum, a UROP in CSAIL, is concerned about energy consumption in the Stata Center and decides to design the EnergyNet, a wireless network of nodes with sensors to monitor the building. Each node has three sensors: a power consumption sensor to monitor the power drawn at the power outlet to which it is attached, a light sensor, and a temperature sensor. Sara plans to have these nodes communicate with each other via radio, forwarding data via each other, to report information to a central monitoring station. That station has a radio-equipped node attached to it, called the *sink*.

There are two kinds of communication in EnergyNet:

1. *Node-to-sink reports*: A node sends a *report* to the sink via zero or more other nodes.
2. *EnergyNet routing protocol*: The nodes run a distributed routing protocol to determine the next hop for each node to use to forward data to the sink. Each node's next hop en route to the sink is called its *parent*.

EnergyNet is a best-effort network. Sara remembers from 6.033 that layering is a good design principle for network protocols, and decides to adopt a three-layer design similar to the 6.033 reference model.

Our job is to help Sara design the EnergyNet and its network protocols. We will first design the protocols needed for the node-to-sink reports without worrying about how the routing protocol determines the parent for each node.

For now (and until stated otherwise), assume that each node has an unchanging parent, every node has a path to the sink, and nodes do not crash. Nodes may have hardware or software faults, and packets could get corrupted or lost, though.

Sara develops the following simple design for the three-layer EnergyNet stack:

Layer	**Header fields**	**Trailer fields**
E2E report protocol	*location* *time*	*e2e_cksum* (32-bit checksum)
Network	*dstaddr* (16-bit network address of destination)	
Link	*recvid* (32-bit unique ID of link-layer destination) *sendid* (32-bit unique ID of link-layer source)	*ll_cksum* (32-bit checksum)

In addition to these fields, each report packet has a *payload* that contains a report of data observed by a node's sensors. When sending a report packet, the end-to-end layer at the reporting node sets the destination network-layer address to be a well-known 16-bit value, SINK_ADDR. The end-to-end layer at the sink node processes each report. Any node in the network can send a report to the sink.

If a layer has a checksum, it covers that layer's header and the data presented to that layer by the higher layer. Each EnergyNet node has a first-in first-out (FIFO) queue at the network layer for packets waiting to be transmitted.

6. [5 points]: What does an EnergyNet report frame look like when sent over the radio from one node to another? In the rectangle below, show the different header and trailer fields in the correct order, starting with the first field on the left. Make sure to show the payload as well. You do not need to show field sizes.

(Show fields in the rectangle below)

Start of frame	*recvid*	*sndid*	*dstaddr*	*location*	*time*	*payload*	*e2e_cksum*	*ll_cksum*

The double vertical bars demarcate fields belonging to different layers.

7. [6 points]: Sara's goal is to ensure that the end-to-end layer at the sink only passes on (to the application) messages whose end-to-end header and payload are correct. You may assume that the implementation of the functions to set and verify the checksum are correct, and that there are no faults when the end-to-end layer runs.

(Circle True or False for each choice.)

A. True / False Using just *ll_cksum* and not *e2e_cksum* will achieve Sara's goal.

FALSE. *For instance, a fault in the network layer could cause an error that corrupts the payload, but the link-layer checksum at each hop would correctly verify.*

B. True / False Using just *e2e_cksum* and not *ll_cksum* will achieve Sara's goal.

TRUE. *This question is actually somewhat ambiguous, and the statement is generally true for non-malicious faults. However, a malicious node could change both the payload and the corresponding e2e_cksum in a way that is undetectable by the sink. In fact, it is also possible that a corrupt packet matches a corrupt checksum entirely by chance, although that will have a very low probability of occurrence. We decided to ignore this choice while grading.*

C. True / False Each node on the path from the reporting node to the sink must recalculate *e2e_cksum* in order to achieve Sara's goal.

FALSE. *The end-to-end checksum is set and verified by only the end-to-end layer.*

To recover lost frames, Sara decides to implement a link-layer retransmission scheme. When a node receives a frame whose *ll_cksum* is correct, it sends an acknowledgment (ACK) frame to the *sendid* of the frame. If a sender does not receive an ACK within some timeout period, it retransmits the frame. A sender attempts at most three retransmissions for a frame.

8. [8 points]: Which of these statements is true of Sara's link-layer retransmission scheme if no node changes its parent?

(Circle True or False for each choice.)

A. True / False Duplicate error-free frames may be received by a receiver.

TRUE. *For instance, an ACK may be lost, causing the retransmission of a frame that was already received.*

B. True / False Duplicate error-free frames may be received by a receiver even if the sending node's timeout is longer than the maximum possible round trip time between sender and receiver.

TRUE. *For instance, an ACK may be lost, causing the retransmission of a frame that was already received.*

C. True / False If each new frame is sent on a link only after all link-layer retransmissions of previous frames, then the *sink* may receive packets from a given node in a different order from the way in which they were sent.

FALSE. *Link-layer retransmissions don't introduce any reordering, and each node's network-layer queue is FIFO.*

D. True / False If Sara were to implement an end-to-end retransmission scheme in addition to this link-layer scheme, the resulting design would violate an end-to-end argument.

FALSE. *The end-to-end argument does not preclude some duplication of functionality when that duplication leads to performance improvements.*

9. [2 points]: EnergyNet's radios use phase encoding with the Manchester code. Sara finds that if the frequency of level transitions of voltage is set to 500 kiloHertz, the link has an acceptably low bit error rate when there is no radio channel interference, noise, or any other concurrent radio transmissions. What is the data rate corresponding to this level transition frequency (specify the correct units)?

(Answer legibly in the space below.)

250 kilobits per second.

10. [8 points]: Consider the transmission of an error-free frame (that is not retransmitted) over one radio hop from node A to node B. Draw lines between the time duration specified in the left column and the contributor(s) to that delay specified in the right column. There may be multiple lines originating from any given item on the left or right column.

(Draw legible and unambiguous lines.)

1. Time lag between first bit leaving A and that bit reaching B.	A. Processing delay
	B. Propagation delay
2. Time lag between first bit reaching B and last bit reaching B.	C. Queueing delay
3. Time lag between when the entire packet was received at A from previous hop, and the same packet about to be sent by A's link layer to B.	D. Transmission delay

ANSWER: $1 \rightarrow B; 2 \rightarrow D; 3 \rightarrow \{A, C\}$.

11. [6 points]: Sara finds that EnergyNet often suffers from congestion. Based on the principles learned in 6.033, which of the following methods is likely to help reduce EnergyNet's congestion?

(Circle True or False for each choice.)

A. True / False If no link-layer ACK is received, the sender should use exponential backoff before sending the next frame over the radio.

TRUE. *Exponential backoff reduces the offered load on the network.*

B. True / False Provision the network-layer queue at each node to ensure that no packets ever get dropped for lack of queue space.

FALSE. *Provisioning queue sizes to ensure that no packets are dropped for want of space only increases delay and does not alleviate congestion.*

C. True / False On each link-layer ACK, piggyback information about how much queue space is available at a parent, and slow down a node's rate of transmission when its parent's queue occupancy is above some threshold.

TRUE. *Each node reacts to rising congestion by slowing down. This approach is a form of congestion control called "hop-by-hop congestion control" (contrast this approach with the "end-to-end" approach discussed in class).*

For the rest of the quiz, nodes may crash and each node's parent may change with time.

Let us now turn to designing EnergyNet's routing protocol that nodes use to form a routing tree rooted at the sink. Once a second, each node picks a parent by optimizing a "quality" metric and broadcasts a routing advertisement over its radio, as shown in the BROADCAST_ADVERTISEMENT procedure. Each node that receives an advertisement processes it and incorporates some information in its routing table, as shown in the HANDLE_ADVERTISEMENT procedure. These routing advertisements are not acknowledged by their recipients.

An advertisement contains one field in its payload: *quality*, calculated as shown in the pseudocode below. The *quality* of a path is a function of the success probability of frame delivery across each link on the path. The success probability of a link is the probability that a frame is received at the receiver and its ACK received by the sender.

In the pseudocode below, *quality_table* is a table indexed by *sendid* and stores an object with two fields: *quality*, the current estimate of the path quality to the parent via the corresponding *sendid*, and *lasttime*, the last time at which an advertisement was heard from the corresponding *sendid*.

```
procedure BROADCAST_ADVERTISEMENT()
   // runs once per second at each node
  if quality_table is EMPTY and node is not sink then return;
  REMOVE_OLD_ENTRIES(quality_table);    // remove entries older than 5 seconds
  if node is sink then {
    adv.quality ← 1.0;
  } else {
    parent ← PICK_BEST(quality_table);    // returns node with highest quality value
    adv.quality ← quality_table[parent].quality;
  }
  NETWORK_SEND(RTG_BCAST_ADDR, adv);    // broadcasts adv over radio

procedure HANDLE_ADVERTISEMENT(sendid, adv)
 quality_table[sendid].lasttime ← CURRENT_TIME();
 quality_table[sendid].quality ← adv.quality × SUCCESS_PROB(sendid);
```

When BROADCAST_ADVERTISEMENT runs (once per second), it first removes all entries older than 5 seconds in *quality_table*. Then, it finds the best parent by picking the *sendid* with maximum *quality*, and broadcasts an advertisement message out to the network-layer address (RTG_BCAST_ADDR) that corresponds to all nodes within one network hop.

Whenever a node receives an advertisement from another node, *sendid*, it runs HANDLE_ADVERTISEMENT(). This procedure updates *quality_table*[*sendid*]. It calculates the path quality to reach the sink via *sendid* by multiplying the advertised quality with the success probability to this *sendid*, SUCCESS_PROB(*sendid*). The implementation details of SUCCESS_PROB() are not important here, and you can assume that all the link success probabilities are estimated correctly.

You should assume that no "link" is perfect; i.e., $\forall i, j, p_{ij} < 1$ (strictly less). Assume that every received advertisement is processed within 100 ms after it was broadcast.

Name:

12. [5 points]: Ben Bitdiddle steps on and destroys the parent of a node N at time $t = 10$ seconds. At what time would node N remove the entry for its parent from its *quality_table*? Round your answer to the nearest second.

Answer: Between $t =$ ___10___ seconds and $t =$ ___16___ seconds.

Suppose that the entry for N's parent is present in N's quality_table at slightly before $t = 10$ seconds. It is possible for the previous few advertisements from the parent to have been lost, so N might remove the entry for the parent at $t = 10$ seconds. (In fact, depending on your interpretation of the question, the earliest time at which N might remove the node is any time before $t = 10$ seconds.)

To obtain the largest possible time at which N would remove the entry for its parent, observe that the parent could have sent an advertisement right before being destroyed, with that advertisement reaching N at slightly after $t = 10$ seconds. If N's per-second check, as part of BROADCAST_ADVERTISEMENT, *runs at $t = 10, 11, 12, \ldots$ seconds, then N will be able to remove the entry only at $t = 16$ seconds. That is the latest possible time.*

We've rounded off all answers to the nearest second, so the exact time taken for the advertisement to be received and processed has been ignored.

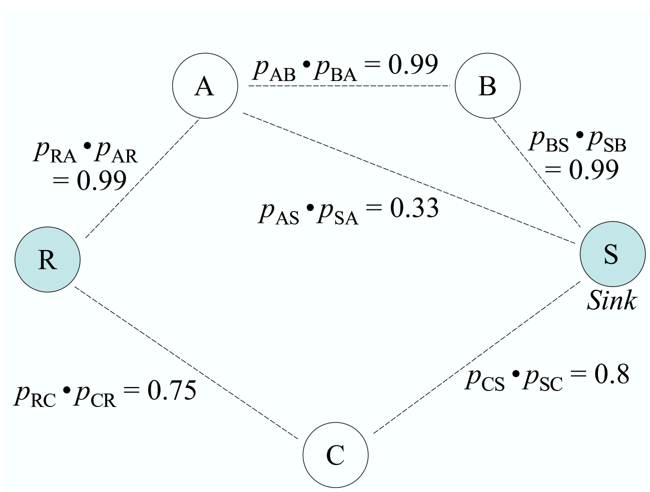


Figure 1: Network topology for some EnergyNet questions.

See Figure 1. The picture shows the success probability for each pair of transmissions (only non-zero probabilities are shown). The number next to each radio link is the link's success probability, the probability of a frame being received by a receiver and its ACK being received successfully by the sender.

13. [6 points]: In Figure 1, suppose B is A's parent and B fails. Louis Reasoner asserts that as long as no routing advertisements are lost and there are no software or hardware bugs or failures, a routing loop can never form in the network. As usual, he is wrong. In the space below, explain why. Give a clear scenario and/or sequence of events.

(Answer legibly in the space below.)

Name:

Consider the following scenario. Soon after B fails, the first advertisement heard by A is from R, saying that R can reach the sink with a certain quality. When A removes B as its parent, it will pick R, because the quality will be higher than any other choice. We now have a routing loop, because A and R each have the other as its parent.

14. [3 points]: In one short sentence, mention a modification to EnergyNet's routing advertisement that can prevent routing loops from forming in any EnergyNet deployment

(Answer legibly in the space below.)

*The simplest way to eliminate this routing loop is to use for each advertisement to include the path to the sink (*i.e., *use a path-vector approach). A node ignores an advertisement whose path already includes it.*

On the face of it, a different way to solve the problem might be for each node to include its current parent in its advertisement, and for a node to ignore any advertisement on which it is the named parent. (There are colorful terms given to variants of this approach in the networking literature, including "poison reverse" and "split horizon".) Unfortunately, this method only prevents two-hop routing loops and does not prevent loops involving three or more nodes.

One might also think that adding a "hop limit" or "time to live" field in the packet header, decrementing that field on each hop, and discarding a packet when this value reaches zero, might eliminate routing loops. That is not correct; this approach only ensures that no packet circulates in the network forever, but does nothing to prevent the routing loop.

15. [2 points]: See Figure 1 again. Which path between R and S would be chosen by Sara's routing protocol? Name the path as a sequence of nodes starting with R and ending with S.

(Answer legibly in the space below.)

Sara's protocol picks the path with the largest end-to-end packet delivery probability. In this picture, that would be R-A-B-S, a path with metric 0.99^3*, bigger than the metrics for the other paths.*

16. [6 points]: See Figure 1 again. Recall that the nodes use link-layer retransmissions for report packets. If you want to minimize the *total expected number of non-ACK radio transmissions* needed to successfully deliver the packet from R to S, which path should you choose? You may assume that frames are lost independently over each link and that the link success probabilities are independent of each other.

Hint: If a coin has a probability p *of landing "heads", then the expected number of tosses before you see "heads" is* $\frac{1}{p}$.

(Explain your answer in the space below.)

The path that minimizes the expected number of radio transmissions is R-C-S. There are three paths to consider.

R-A-B-S: The expected number for this path is $3/0.99 \approx 3$.
R-A-S: The expected number for this path is $1/0.99 + 1/0/.33 \approx 4$.
R-C-S: The expected number for this path is $1/0.75 + 1/0.8 = 31/12 < 3$.

Name:

Sara finds that each sensor's reported data is noisy, and that to obtain the correct data from a room, she needs to deploy $\ell > 1$ sensors in the room and take the average of the ℓ reported values. However, she also finds that sensor nodes may fail in fail-fast fashion. Whenever there are fewer than ℓ working sensors in a room, the room is considered to have "failed", and its data is "unavailable". When that occurs, an administrator has to go and replace the faulty sensors for the room to be "available" again, which takes time T_r. T_r is smaller than the MTTF of each sensor, but non-zero.

Assume that the sensor nodes fail independently and that Sara is able to detect the failure of a sensor node within a time much smaller than the node's MTTF.

Sara deploys $m > \ell$ sensors in each room.

17. [6 points]: Sara comes up with three strategies to deploy and replace sensors in a room:

A. Fix each faulty sensor as soon as it fails.

B. Fix the faulty sensors as soon as all but one fail.

C. Fix each faulty sensor as soon as data from the room becomes unavailable.

Rank these strategies in the order of highest to lowest availability for the room's sensor data.

A C B. *Suppose x is the number of sensor nodes that fail before they are replaced. Since $m > \ell > 1$, the availability of the room's data is a decreasing function of x; replacing a faulty sensor after a smaller number of failures improves availability.*

End of Quiz II!

Bonus question: Answer this question only if you have time. We will give extra credit for a correctly explained answer, but you can score 100 even without answering this question!

18. [2 points]: Suppose that each sensor node's failure process is memoryless and that sensors fail independently. Sara picks strategy C from the choices in the previous question. What is the resulting MTTF of the room?

Let T_f be the MTTF of a sensor node. The expected time for the first of m nodes to fail is $\frac{T_f}{m}$. When that happens, the expected time for the next sensor node to fail is $\frac{T_f}{m-1}$, because the failure process is memoryless. The room becomes unavailable when only $l-1$ sensor nodes are up, which means that the room's MTTF is

$$T_f(\frac{1}{m} + \frac{1}{m-1} + \ldots + \frac{1}{l}).$$

Notice that the argument underlying this calculation is essentially equivalent to the one made for plane engines in Chapter 8.

Name: