Solution for Homework 2

May 30, 1999

Summary

The solution for each problem consists of the problem text in helvetica, a description of the grading scheme, a list of common mistakes, and a sample answer that was given full credit, even if it's a little bit wrong, in `courier`. A 1.0 represents full credit for each part of problem 1, 2.0 represents full credit for problems 2 through 5, for a grand total of 11 points for this assignment.

Again, I've scripted up some cumulative distribution functions (the little graphs) for each of the problems and the overall grade. A cumulative distribution is like an integrated histogram. You can count the number of students who got a better or worse grade than you by finding your score on the x axis, and seeing what percentile that corresponds to on the y axis. This seemed more informative than just an average score or a median.

The cdf for the entire homework set is below. I've also added the cumulative score (from hw1 and hw2) for the curious.

The median score is where the y-value is 0.5. You can see the lowest score at the far left, and the highest score on the far right. If you have trouble reading these, let me know.

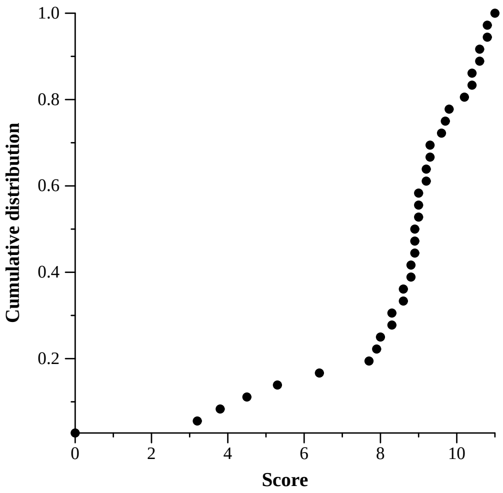


Figure 1: CDF over the entire homework

Problem 1a

Under what circumstances, if any, is it possible for router A to forward packets for router B through router C, even though routers A and B are neighbors.

Most everyone got this right.

Grading Scheme

Full credit to those answers describing a situation where the cost of the route A-C-B is lower than the cost of A-B. If you decided that distance vector approaches only use hop-count to measure distance but explained a way to make path-based distance vector route around a direct link (as a policy), that was ok too.

Sample Correct Answer

```
i. Assuming that the network links are
   weighted, it may be cheaper (con-
```

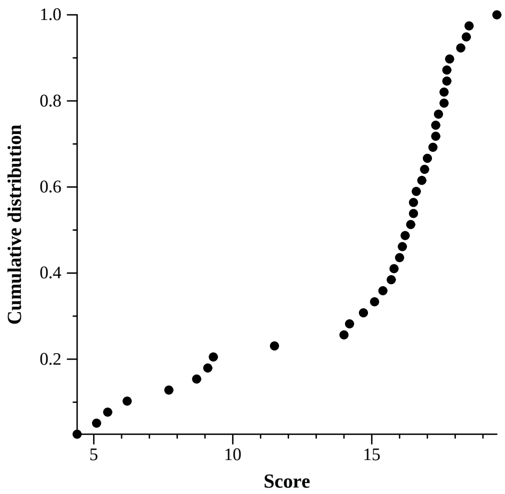
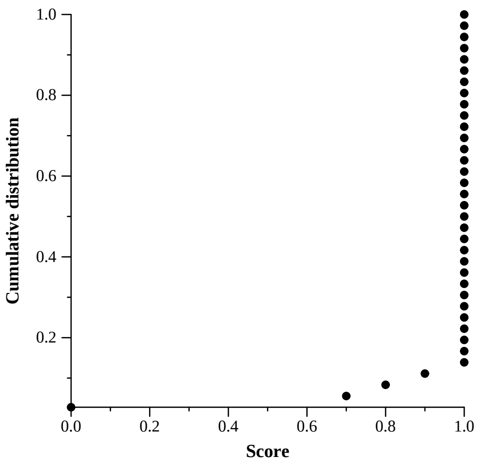


Figure 2: CDF over the both homeworks, hw1score + hw2score.

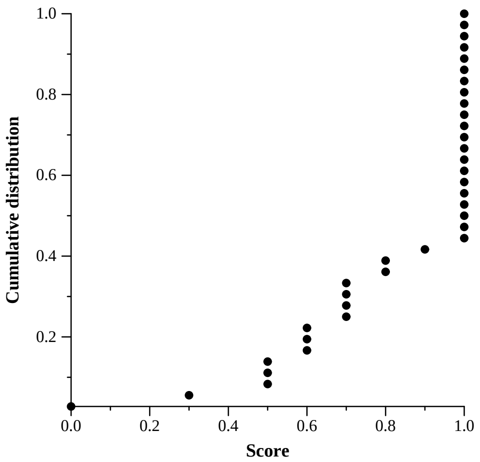
gestion, drop rate, etc.) to take a different route that results in an additional hop. Even if the weights are proportional to distance, the fact that A and B are neighbors does not guarantee that the A-B route is shortest. It still may be shorter to take an additional hop, and Dijkstra's algorithm will find that shorter path.

ii. Similar arguments can be made for a traditional distance vector algorithm. For whatever reason, A to B through C may be cheaper than A to B, direct. When A receives forwarding info from C, it will decrease its A-B distance value by pointing to C as its "next hop".

iii. Assuming BGP, as the path-based distance vector algorithm, it's easy to come up with a situation that would allow A to B through C. Namely, the BGP protocol at A is literally told not to route directly to B, or perhaps is told to favor C over B. We don't know why, since BGP strives for reachable nodes vs. optimal paths.



Problem 1a



Problem 1b

Problem 1b

Suppose that A and B are neighbors, but A thinks the cost of the link to B is x, and B thinks the cost to the link to A is y.Will this cause any problems?Explain in terms of an example.

This is likely to cause asymmetric routes. Not loops, oscillations, or instability.

Grading Scheme

Full credit to those that using the word "asymmetric," and those who were very close to it.

Common Mistakes:

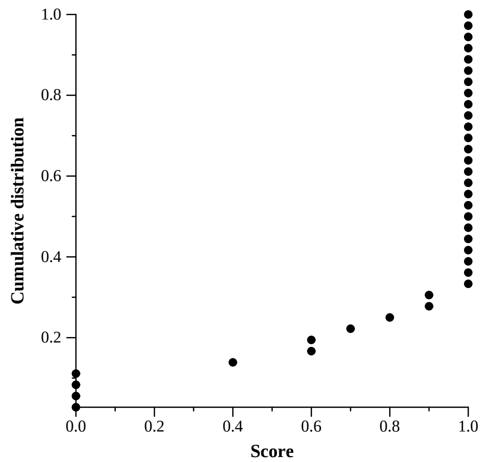
An asymmetric route *does not* imply any problems in estimating round trip time. Round trip time, by it's very nature (round trip) doesn't care about asymmetry. Taking the travel example, lets say you're going to Belize. (I wish I was going to Belize) On the outbound leg, assume you go through Dallas. Coming back, assume you go through Chicago. Let's say it takes longer to go through Chicago. The total time you spend in the airplane (round trip time) is pretty much the same, no matter how many times you do this, ignoring wind speed, snow in chicago, etc. So, it's pretty easy to estimate how much plane time you need to spend to get back and forth from Belize. As an added bonus, people in Belize who want to go to Seattle for vacation (right) spend the same amount of time in an airplane for their round trip. Belabored enough for you?

Sample Correct Answer

```
If two neighboring routers believe their
shared network link has asymmetrical
cost, then no harm is done.  The network
link may actually have an asymmetrical
cost.  Using any routing system, the
routers in the network will not be con-
fused because their routing tables will
only consider the asymmetrical link in
one direction.
```

Problem 1c

Give a topology in which, due to an inconsistency in the information about a single link, many routes are disrupted.Give a topology in which, despite an inconsistency in the information about a single link, no routes are disrupted.



Problem 1c

Grading Scheme

I accepted most answers reasonably presented. Interestingly, some topologies people used for the many-routes-disrupted cases were also used for other's no-routes-disrupted case, with the argument that if there's no other way to go than through the single link connecting two networks, there's no disruption, since the route stays the same.

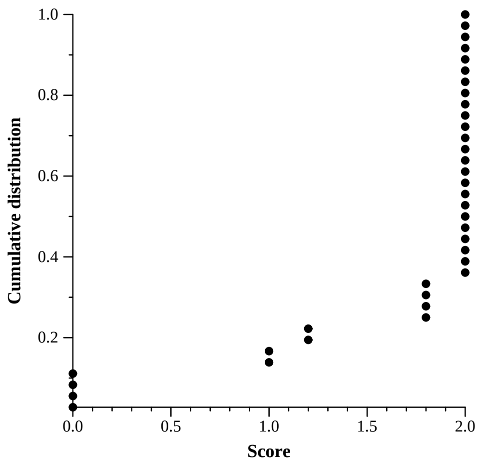
So, the two acceptable readings of this problem were, given inconsistent information,

- Routes have to be changed to go somewhere else.
- Routes could be blocked.

The easy way to get the no-routes disrupted case was to point to an unused (high cost, or out of the way, or redundant) link.

Problem 2

Consider an arbitrarily large network in the form of a fully balanced binary tree of L levels.There are 2^k nodes at level k (for $k = 0..L-1$) and there are $2^L - 1$ nodes altogether. Each leaf at the bottom of the tree represents a host and each internal node represents a router. Your task is to devise a scheme for assigning addresses to nodes that is optimal in the sense of minimizing the maximum size of the routing tables across all routers,



Problem 2

assuming each entry in the routing tables store only prefixes with unique routes (in other words, the routers are CIDR-based). Indicate what it stored in each router's forwarding table.

This seemed pretty obvious to people. Common variations were:

- to give the root of the tree a bit, that then prefixed everyone elses bit and was somewhat wasted.
- to make routers addressible (*difficult!*)

One thing to note about such a topology is how much contention you'd expect at the root of the tree. You could also conclude that the root node itself isn't really necessary.

Common mistakes:

- Not including the default / parent / up-the-tree route.
- Not using correct addresses for the children.
- Not using CIDR.

Grading Scheme

Full credit for a tree, an addressing scheme, and a routing table. Generally less if the routing table was missing entries or didn't make sense.

Sample Correct Answer

```
The length of the network address of the
nodes may be of 1 to k+1.  The address
of k-1 level nodes are the prefix of the
address of k level nodes.  The last bit
value of the network address of a k level
child node is 0 if it is the left child
and 1 if it is the right child of its
k-1 level parent.  The network address
for the root node is 1.  Each entry in
the routing table represented by [prefix,
unique route] and the routing table for
each internal node has three entries.
Assume the network address of an inter-
nal node is "m", its routing table will
be like ["m0", left child], ["m1", right
child], [other, up].  The routing table
of the root node should be [10, left],
[11, right].
```

Problem 3

Consider the following proposal for hierarchical source routing based on the Internet's "loose source routing". With loose source routing, a host specifies a list of routers to visit on the way to the destination; the network is free to use any means of getting between the routers in the loose source route. A hierarchical source routing scheme could be built on top of this, by saying that when a packet is delivered to a router specified in the loose source route list, it can in turn "push on the stack" a loose source route to the next router in the original list. Explain why this might be better or worse than the way hierarchical routing is done in the Internet today. Where appropriate, give concrete examples.

The key realization for this problem is that it moves routing state from the routers to the end nodes. This has two (related) problems:

- State has to get to the hosts, and be kept up-to-date. Perhaps hosts won't realize the best way to get their packet across because they use obsolete information.

- Failure is not transparent to end hosts. Perhaps hosts will choose a path that won't work, and what is a router to do?

Lesser problems include

- Packet processing overhead. This is significant if we believe routers are cpu-bound.

- Unbounded packet header size. If a convention is adopted where "pushed" routers "pop" themselves off the packets stack, which is how I read the question, one could arbitrarily limit the size of the source route. A source routing header of size 1 would degenerate to normal internet routing.

- Integration with existing hardware.

Common mistakes:

Loops aren't any more likely with loose source routing than without.

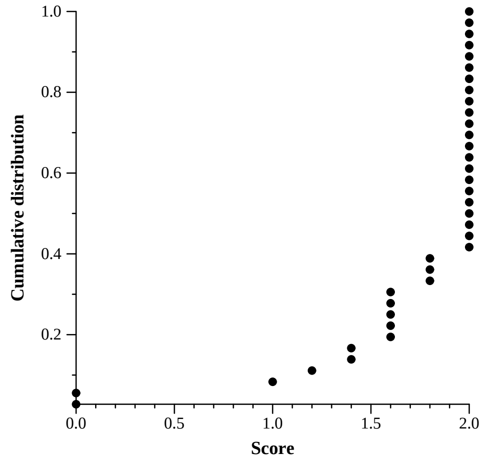
Router complexity isn't reduced, it still needs to know everything to deliver the loose-est of source routes.

Sample Correct Answer

```
pro - A congested or downed link could
    be effiently routed around before
    getting near the problem.

con - The source has to know the net-
    work topology and state at any given
    moment to be effecient.

pro - It would be a useful diagnos-
    tic/debugging tool.  e.g.  doing a
    traceroute from any arbitrary point
    to any arbitrary point.
```



Problem 3

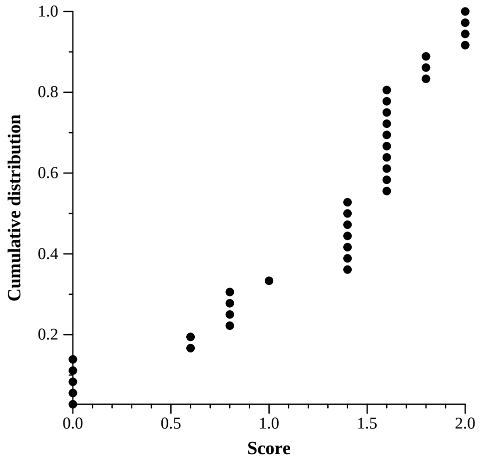
```
con - The extra information in the packet
    increases it's size and the complex-
    ity of handling it.

con - It would make denial of service
    attacks easier.  The attacker could
    compromise a single router and all
    traffic through it could be sent
    to/away from the attackee.  The ex-
    isting scheme seems more likely to be
    self correcting.

  On the whole it seems like extra work
for little gain.  Thumbs down.
```

Problem 4

Draw a timing diagram illustrating all the packets that are sent using TCP between a client and server, when the client requests a 10KB web page, and the previously determined maximum transfer unit without fragmentation is 512 bytes.Include connection setup and teardown, slow start, delayed acks, etc. Roughly how many round trips does it take?What happens if the third data packet from the server is dropped?How many round trips in this case?



Problem 4

Grading Scheme

0.2	delaying the first ack 200ms
0.4	delaying other acks (as in sending every other)
0.2	setup of the connection (syn/ack)
0.2	teardown of the connection (fin/ack)
0.8	slow start (increase by one per ack)
0.4	round trip estimate around 8 (between 6-12)
0.2	round trip estimate in 3rd packet loss slightly greater than previous.
-0.2	not doing a web transfer (as in client sends request)
-0.4	not having 20 (or more) packets

The attentive reader will notice that the total adds up to more than 2. I added up all the features you got right, with an upper limit of 2, then subtracted all the negative features to arrive at a grade for this problem.

Common mistakes:

As you can tell, two common mistakes were:

- Drawing a sender and receiver instead of a client and a server, then showing a 10k ftp-like transfer instead of a web transfer.
- Not drawing as many packets as were required.

It's much easier for a TA to grade based on whether you followed directions than whether you understand what you're talking about.

Sample Correct Answer

A sample is shown in Figure 3, which is large enough for latex to bounce somewhere random in the file.

Problem 5

Suppose we modified TCP to use the DECbit constant for multiplicative decrease (0.875 instead of 0.5).Assuming packets are lost periodically at the tip of the sawtooth, how would this change the formula in Mathis et al.?Is there any downside?

Key to getting credit to this problem was looking like you understood what you were doing. In particular, the area under the sawtooth should have a width of $\frac{W}{8}$, and consequently, a total area of $\frac{15}{128}W$.

Common mistakes:

- Taking the 0.875 as a subtractive factor. For example, if the window size was 8 when a conestion loss occurs, going back to 1 (incorrect) instead of 7 (correct). I tried to be generous for this one, since it's an honest mistake, but it is described at the bottom of page 416 in the text.
- Not getting the area under the sawtooth correct
- Not coming to a reasonable conclusion about the difference in performance. TCP only misses $\frac{1}{4}$ of the W^2 area, it would be very difficult for anything to get more than a 30% gain over TCP.

Grading Scheme

The problem was broken into derivation and analysis. For derivation, a reasonable explanation with the correct sawtooth equation was given full credit. For the analysis, any of the following observations weighed highly:

- Interaction with slow start: it would be cruel to increase by a factor of two each time, then backoff by this small factor. So you'd have to slow start slower, which would probably hurt slow connection performance.
- The expected loss rate would increase, since you reach the limit, W, four times more often, so you'd have more timeouts and retransmissions.
- Interaction with existing technology: a sender like this would get much more than its fair share competing with TCP flows.

Sample Correct Answer

If we use the DECbit constant (0.875) for TCP, it will have the following effect on the Mathis et. al. formula:

- The equilibrium point will become (7/8)W
- Assuming that the receiver acks every segment, the window will open by one segment per round trip and the time period of the saw tooth will become W/8 (Since it will take W/8 rtts to reach from (7/8)W to W).
- Given that we loose a packet at the tip of the saw tooth and the link delivers aprrox. 1/p consecutive packets followed by one drop, each cycle must be RTT*W/8 seconds.

Total data delivered (area under the saw tooth) = (7/8)*W*(W/8) + (W/8*W/8)/2 = (15/128)*W*W

Since this is equivalent to 1/p, we can solve for W

W = square_root(128/(15p))

BW (Bandwidth) = (data per cycle) /(time per cycle) = (MSS*(15/128)*W*W)/(RTT*w/8))

Replacing W by p we get : BW = (MSS/p)/(RTT* (square_root(2)/square_root(15p)))

The implications of using DEEbit constant 0.875 for Multiplicative decrease:

1. By decreasing the congestion window size by a small fraction of the original window size when packet loss occurs, we will see more oscillations around the point of equilibrium. Thus the system takes a longer time to stabilize. The effects are more pronounced when the congestion in the network is severe - in such cases, the traditional TCP constant of 0.5 works much better.
2. DECbit constant works better if a single packet loss usually doesn't mean that there is severe congestion in the network.
3. The DECbit constant also works better for networks that use explicit congestion notification schemes to alert the end host that there is congestion in the network. In such cases, an action as drastic as cutting the congestion window size in half is not required.

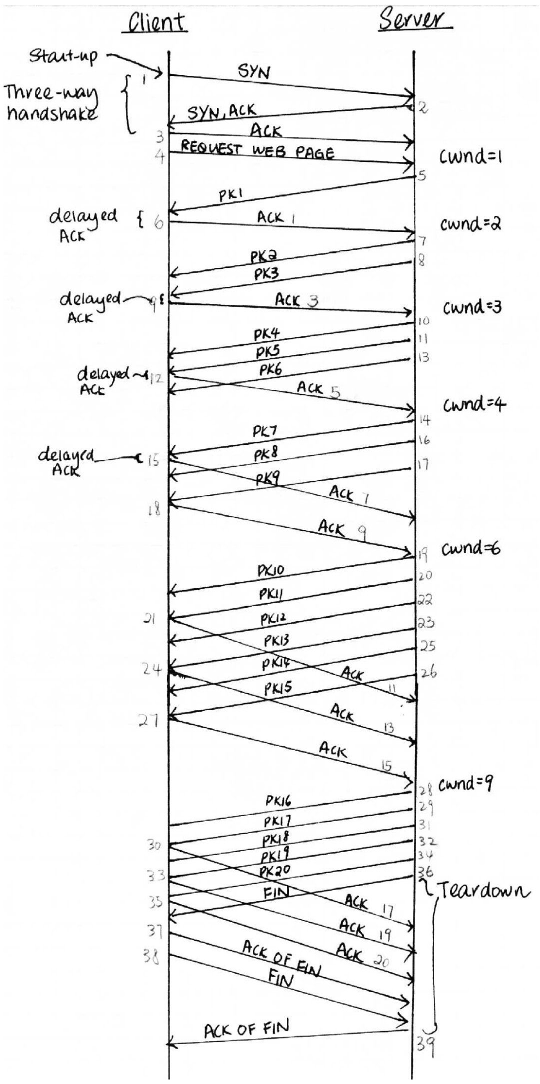
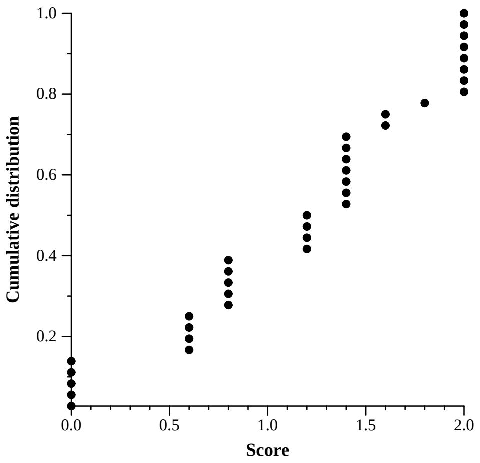


Figure 3: Sample Answer for Problem 4



Problem 5