EE 122 Fall 2010 Discussion Section III

12 October 2010

http://www.cs.berkeley.edu/~alspaugh/ee122/fa10/

Question 1: HTTP

a. Describe the process of loading the following web page:

`http://www-inst.eecs.berkeley.edu/~ee122/fa09/class.html`

1. Obtain the IP address of the server associated with `www-inst.eecs.berkeley.edu` through DNS.
2. Initiate a TCP connection with the server.
3. Send an HTTP request.
4. Receive an HTTP response.
5. Request whatever objects are embedded in the response.

b. Specify the first line of each HTTP header. Assume the IP address is unknown, that we are using HTTP 1.1, and that the web page is available.

GET /~ee122/fa09/class.html HTTP/1.1
HTTP/1.1 200 OK

c. Explain how stateless HTTP can implement a stateful user session such as a shopping cart application at amazon.com. Name one security vulnerability of your answered approach.

Use cookies to store state at the client. Subsequent requests from the client will contain the cookie, allowing the server to associate the client with a session. One vulnerability of this approach is that a malicious attacker can impersonate the client if they are able to obtain the client's cookie. In reality, many cross-site scripting attacks are based on this.

d. Your web site prospers but your ISP starts to complain about your web traffic. Where would you put a web cache to handle this problem?

Put a cache close to the clients (this is known as a forward proxy) or actually at the client.

e. Can any web content be cached? If so, justify it. Otherwise, describe what kind of content cannot be cached.

Not all web content can be cached, only static content can. This means content that does not change based on who access it.

Question 2: Ethernet and Multiple Access

a. What is the expected backoff time in a 10 Mbps Ethernet for a host that has just experienced three collisions in a row?

The transmission delay is .1 microseconds. The backoff time is k x 512 bit times, for k in {0, ... 2^m-1}, and m = max(10, number of collisions). Thus the average backoff time is 1/8 (0 x 51.2 + 1 x 51.2 + ... + 7 x 51.2) = 179.2 microseconds.

b. Suppose two hosts are connected via a CSMA/CD network, where the medium transmits at 1 Gbps. If the minimum frame length is fixed to 1000 bits and the propagation speed is 2×10^8 m/s, how far can the hosts be apart?

In such a network, a host must send data for at least 2d, where d is the propagation delay. d = D/V, where D is the distance and V is the propagation speed. Thus the minimum frame size is L = 2dR, where R is the transmission speed. Plugging in D/V for d and solving for D yields 100 m.

c. If the network mentioned in the previous question is Ethernet, what is the approximate efficiency of the network when the hosts are the maximum distance apart? What about if the distance is reduced to 1 m?

Using the formula from the textbook which approximates Ethernet efficiency as $1/(1 + 5d_{prop}/d_{trans})$, and assuming the maximum frame size is 1500 bytes, the transmission delay is 12 x 10^{-6} seconds, the propagation delay is 0.5 x 10^{-6}seconds, so the efficiency is about 0.828 when the distance is 100 m. If the distance is 1 m, the efficiency is .980.

d. Suppose there are *n* hosts sharing a broadcast link. Each host has a frame to send with a probability p. Which multiple access protocol would you use if p is low? What about if p is high?

If p is low, there is little chance for collision, so a random access protocol such as CSMA/CD would be appropriate. Otherwise, a channel partitioning protocol such as TDMA or FDMA is more efficient. One might also consider a taking-turn protocol such as token rings.

Question 3: Hubs and Switches

Assume the topology in Figure 1, where A, B, and C are end hosts that have notC yet sent a packet and whose network interfaces are in promiscuous mode. Initially, the link between U and V is disabled.

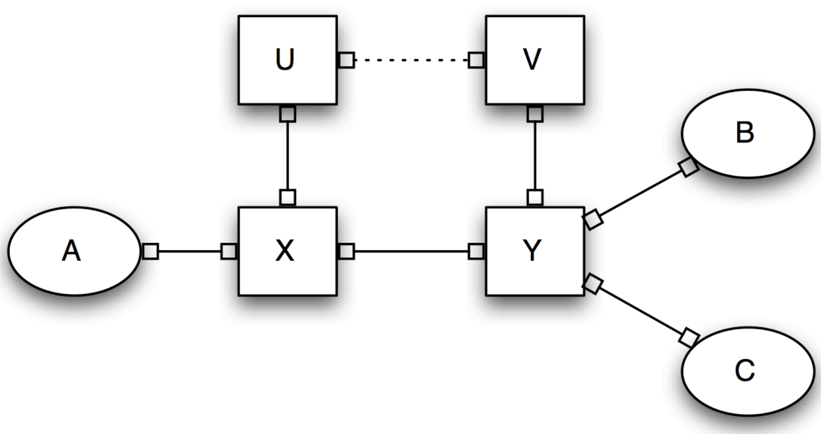

Figure 1

a. What is the process of sending a request message from A to B and getting a reply back from B to A, if U, V, X, and Y are hubs? Can C see any messages? If so, which messages?

A sends a message to X, which floods it out on its other ports. Y receives this message and also floods it out its ports. When B replies, again Y and X flood the message out on all of their outgoing ports. Thus C can see both the message from A and the message from B.

b. Repeat the same question assuming U, V, X, and Y are now switches.

In this case, when A sends the message, X and Y flood it out on their outgoing ports as before, but they also learns which port A is on. Thus, on the reply from B, X and Y do not flood the message, they send it directly out the link to A. So in this case, C only sees the message from A.

c. Imagine the U-V link is now enabled. Can we always guarantee communication between A and B if U, V, X, and Y are hubs? What about if they are switches?

With that link enabled, there is a forwarding loop in both cases. In the case that they are hubs, this loop is unpreventable.

d. If any case in the previous question prevents communication, how would you solve the problem?

There is no way to solve the hub case aside from disabling the link again. In the switch case, we can build a spanning tree and disable the non-tree links.

e. Suppose we disabled the U-V link again and that B has moved and is now connected to U. Can A and B communicate if the squares U, V, X, and Y are hubs? What about if they are switches?

If they are hubs, A and B can communicate because hubs flood packets. If they are switches, X and U have remembered that B is reachable through Y, so messages will not reach B until B's old location information expires or they update B's location information when B sends a new message through U. Messages will still reach A, however.

f. If any case in the previous question prevents communication, how would you solve the problem?

To enable communication when B moves, the location state associated with B must eventually expire so it can be replaced with correct information. (This is known as soft state.)

Question 4: Bridge Spanning Tree Algorithm

"Algorhyme"
I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span.
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me
Then bridges find a spanning tree.

Radia Perlman, inventor of the spanning tree algorithm

a. Given the extended LAN shown in Figure 2, indicate which ports are not selected by the spanning tree algorithm. The LANs in the figure are labeled A – J, and the bridges are labeled B1 – B7. For a hub B*i*, *i* is the hub ID.

The key thing to remember is that every LAN has a designated bridge through which to forward packets, and this bridge should be the one with the shortest path to the root. In the case of ties, the bridge with the lowest ID number is chosen. The root is always the bridge with the lowest ID overall.

Thus, in this case, the links that are pruned from the tree are: B2 – A, B5 – B, B5 – F, and B6 – I.

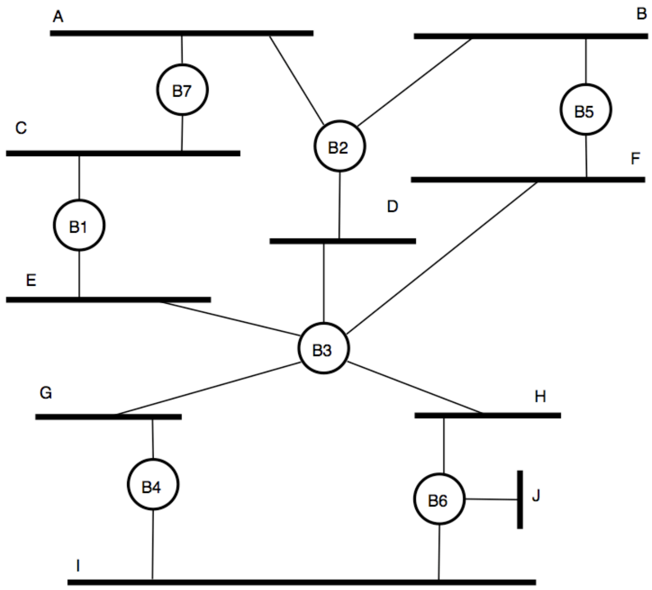

Figure 2

b. Now assume that bridge B1 fails. Indicate which ports are not selected by the spanning tree algorithm after the recovery process completes and a new tree has been formed.

In this case, the new root will be B2, so the links that are pruned are B5 – B, B5 – F, and B6 – I.

Credits: Questions 1 – 3 were adapted from the third homework of the fall 2009 offering of EE 122. Question 4 is from Chapter 3 of Peterson and Davie's textbook.