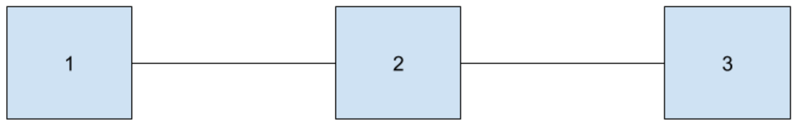
CS 168 Fall 2017 | Introduction to the Internet Scott Shenker | Problem Set

1 Delays



Two packets P_1 and P_2 of size B_1 and B_2, respectively, are sent from router 1 to router 3. Consider the following propagation delays:

- 1 to 2: 0.2 s
- 2 to 3: 0.1 s

and the following transmission speeds:

- 1 to 2: 10000 bits/second
- 2 to 3: 5000 bits/second

In each of the following 3 cases, determine the queueing delay experienced by P_2 at router 2, assuming P_2 begins to be transmitted as soon as P_1 is completely on the wire (at router 1), and that P_1 begins to be forwarded by router 2 only after the complete packet has arrived (i.e., this is not cut-through forwarding).

1. $B_1 = 500$ bytes and $B_2 = 500$ bytes.

   Solution: There is a queueing delay of 0.4 seconds.

2. $B_1 = 1000$ bytes and $B_2 = 500$ bytes.

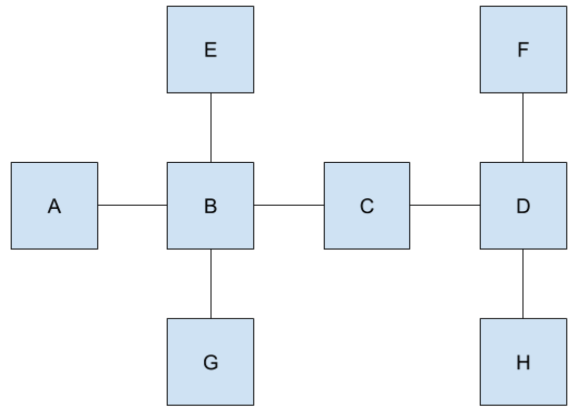
   Solution: There is a queueing delay of 1.2 seconds.

3. $B_1 = 500$ bytes and $B_2 = 1000$ bytes.

   Solution: There is no queueing delay.

2 L2 Routing

Consider the following network of L2 learning switches:



In all parts of this question, assume that the events given are the only events in the network, unless otherwise specified.

In every scenario below, you can observe some set of the switches and note whether or not you see a particular packet. You may assume that no failures occur in the network.

Assume that (i) the events occur one after another, (ii) there is no forwarding state at the start of this scenario, and (iii) no forwarding state times out.

1. You observe a host attached to switch A sending a packet. Is the packet seen at switch F? **Solution:** Yes, the packet is flooded.

2. You observe a host attached to switch G sending a packet. Is it possible for the packet to appear at switches E and F, but not at switch D? **Solution:** No, to get to F, it must pass through D.

3. You observe a host attached to switch A sending a packet. Is it possible for the packet to appear at switches B, D, F and H, but not at switch E? **Solution:** No, only A and G have sent packets, so they are the only destinations known. If the packet appears at D, F and H then it is not destined for G, and the switches will flood, so E must see the packet.

Now, consider the same network and the above events have happened. However, exactly one (unknown) switch loses its forwarding table prior to each event. Assume that the events occur one after another. Say whether each scenario is possible.

1. You observe a host attached to switch A sending a packet. The packet is seen at switches B, D, F, G and H. **Solution:** Yes, it could be flooded.

2. You observe a host attached to switch A sending a packet. The packet is seen at *only* switches B, C and D. **Solution:** No, only A and G are known destinations. Since A is sending, the packet is meant for either G or another switch. If it is meant for G and B still has forwarding state, then the packet would be sent on the path A-B-G, which is not the case. If B lost its state, then the packet would be flooded *at least* to E. Otherwise, if G were not the destination, then the packet would simply be flooded and all switches would see it.

3. You observe a host attached to switch C sending a packet. The packet is seen at *only* switches A, B and D. **Solution:** Yes. If C lost its table, it would flood to neighbors. If the destination was A, then B and D may still have their tables and make efficient decisions (forward to A and drop, respectively).

3 Fun with STP

For each scenario below, indicate the largest range of values that the variable in question can hold (ignoring its dependence on other unknown variables), assuming no problems are encountered when running the spanning tree algorithm. Remember that there are no ordering or timing guarantees in a distributed algorithm. The smallest switch ID is 1, and the largest is 10, and all others are unique integers between these two. All scenarios are independent.

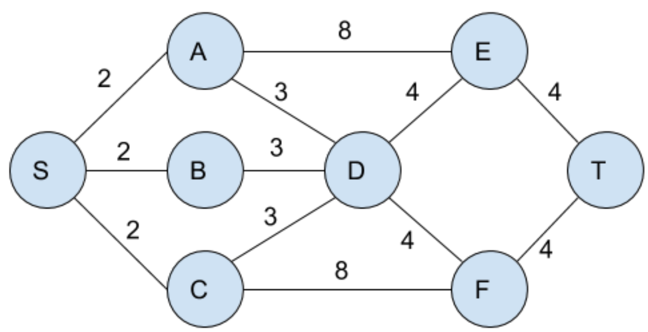
1. Switch S sends (x, 0, z) to its neighbors. What values can x and z take? **Solution:** x = z, x $\in$ [1,10]

2. Switch S receives (x, y, z) from its neighbor. It then advertises (q, r, 5) to its neighbors. What values can x, y, z, q, and r take? **Solution:** $x \in [1,z]$, $y \in [0,9]$, $z \in [1,4] \cup [6,10]$, $q \in [1,5]$, $r \in [0,9]$

3. Switch S receives (1, x, y) from a neighbor. It then advertises (1, t, 3). What values can t take? **Solution:** $t \in [1, x+1]$

4. Switch S has sent (x, y, z) to its neighbors. Upon receiving (a, b, c) from a neighbor, it sends (a, s, r). What values can a and x take (assume that $a \neq x$)? **Solution:** $x \in [2,10]$, $a < x$, $a \in [1,9]$

4 L3 Routing



Suppose we have an L3 network with the topology shown above, and the routing algorithm used is link state.

1. After convergence, what is the path cost from S to T, and what are all the possible paths with this cost? **Solution:** The cost is 13. All paths are S $\rightarrow$ [A, B, C] $\rightarrow$ D $\rightarrow$ [E, F] $\rightarrow$ T

2. Suppose that a control message (a message used by the routing algorithm) takes 1 second to propagate along a link. What individual link failure(s) would cause the longest delay to reconvergence, and what is that delay? **Solution:** Link SB causes a 3 second delay.

3. Suppose you have the ability to take down individual nodes. Which nodes would you take down in order to partition the network? If you can't partition the network, which nodes would you take down to increase path costs maximally? In each part, suppose you can take down:

   (a) a single node. **Solution:** D. Can't partition, and removing D increases path costs the most.

   (b) two nodes. **Solution:** E and F. Removing them will partition T from the network.

4. Which single link's cost (if any) should you double in order to increase the path cost from S to T? **Solution:** It's not possible, there is always an alternate, cheaper path.

5 Reliable Transport and Congestion Control

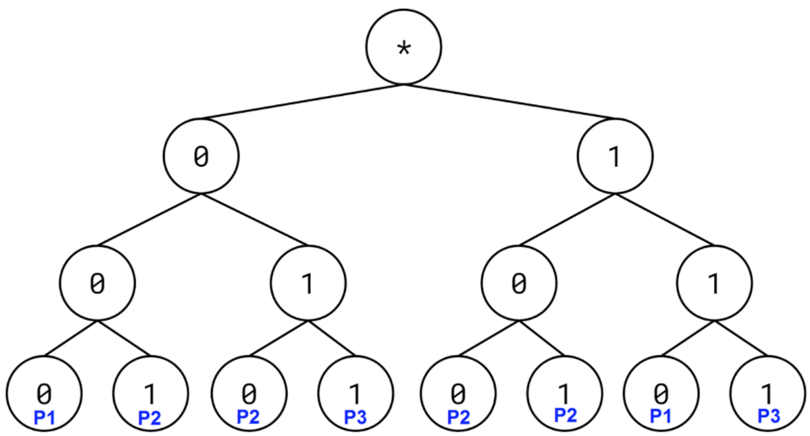
Your company is considering possible changes to the networking stack used on their machines, but they have some questions.

1. In the first approach, when a packet drop is detected by duplicate ACKs, CWND becomes 0.25 CWND, and when a full window's worth of packets is ACKed, the window is increased by 2MSS. Is this more aggressive than normal TCP, or less aggressive, or is this TCP-friendly? **Solution:** More aggressive.

2. Another proposal is that the receiver only send an ACK once two packets have been received (these packets can contain the same payload). Is this reliable? Would this interoperate with existing TCP implementations? **Solution:** Yes to both.

3. The last proposal is to save cycles by not setting any timers, and rely exclusively on duplicate ACKs for loss detection and retransmissions. Is this reliable? **Solution:** No.

6 Forwarding



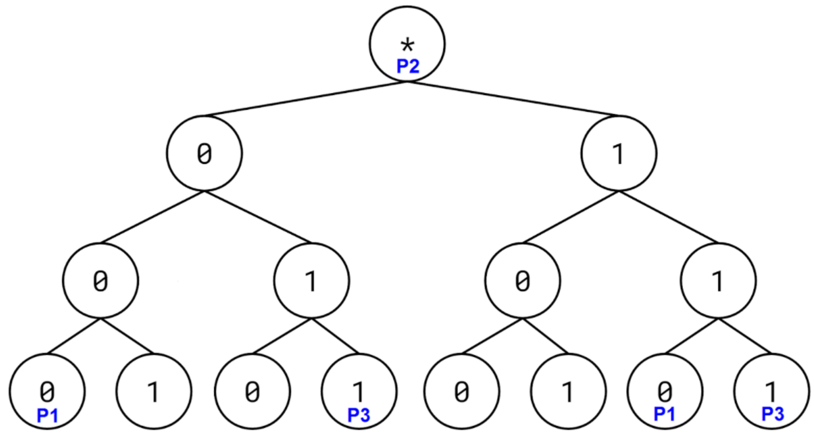
You have a router with 3 outgoing ports, with forwarding entries represented by the above prefix tree.

1. Convert the above prefix tree into a forwarding table with minimal entry count.

   Solution:
   *** $\rightarrow$ P2
   000 $\rightarrow$ P1
   011 $\rightarrow$ P3
   110 $\rightarrow$ P1
   111 $\rightarrow$ P3

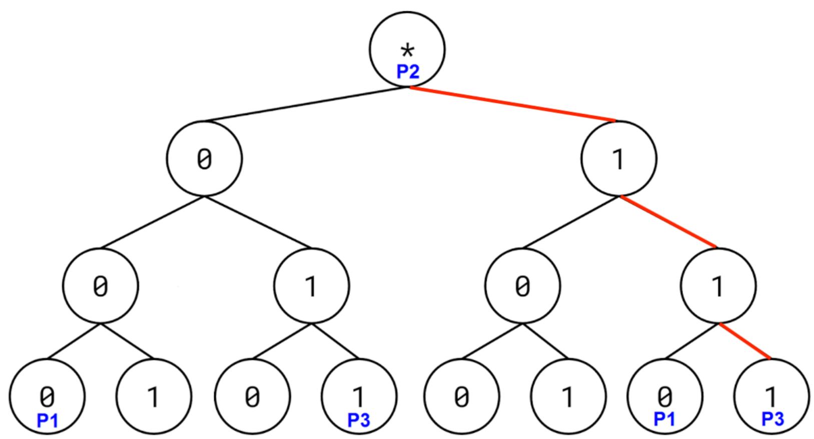
2. Convert the forwarding table you just constructed back into a prefix tree for use with LPM.

   Solution:

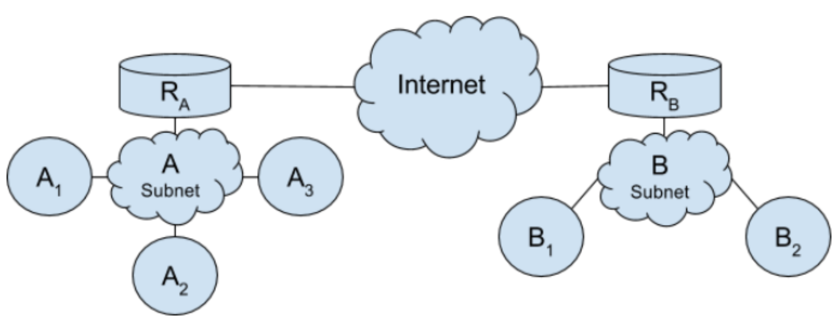


3. For this problem, assume IP network addresses are 3 bits long. What path would be traversed in the LPM tree to match the IP address 7?

Solution:



7 ARP



Consider the topology above. ARP caches start off empty. Circles refer to hosts, whereas Cylinders refer to routers. Everyone knows each others IP addresses from previous DNS lookups. Assume switches learn promiscuously from broadcasts and other packets it receives.

1. Host A_1 sends a message to host A_3. Which entities see the ARP request? **Solution:** R_A, A_2, A_3
2. A_3 responds. Who knows its MAC address? **Solution:** A_1
3. Host B_1 sends a message to A_3. Whose MAC address does the it need first? **Solution:** R_B
4. When R_A gets the packet from the last part, does it need to send an ARP request? **Solution:** Yes, as only A_1 knows where A_3 is
5. Host A_3 sends a message to host B_2. Whose MAC does A_3 need first, and does it have it? **Solution:** R_A; it does from the previous interaction
6. After B_2 gets the packet, which end hosts does R_B have in its ARP cache? **Solution:** B_1 and B_2
7. What about R_A? **Solution:** A_1 and A_3

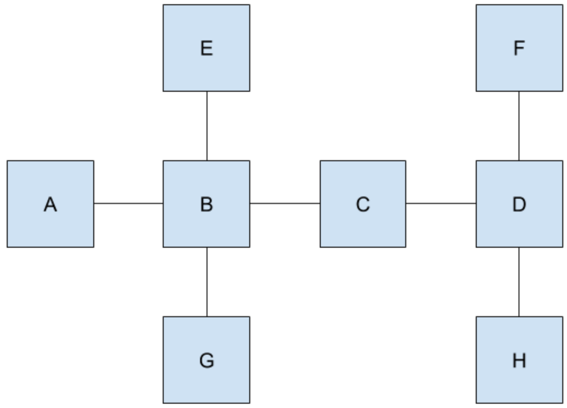
8 DHCP

Consider an L2 network with D DHCP servers and H hosts (none of whom have an IP address yet).

How many of each of the following packets would you expect to see after all hosts get IP addresses? Assume no packets are sent besides DHCP messages, and assume everything works correctly.

1. Packets with source IP 0.0.0.0 and destination IP 255.255.255.255 **Solution:** 2H
2. Packets with source IP X and destination IP Y, where X is an offered IP address and Y is any DHCP server's IP. **Solution:** 0
3. Packets with source IP 0.0.0.0 and destination IP Y, where Y is any DHCP server's IP. **Solution:** 0
4. Packets with source IP X and destination IP 255.255.255.255 where X is the IP of any DHCP server. **Solution:** DH+H
5. Packets with source MAC X and destination MAC FF:FF:FF:FF:FF:FF, where X is the MAC of any host. **Solution:** 2H
6. How many ARP requests were sent? **Solution:** 0

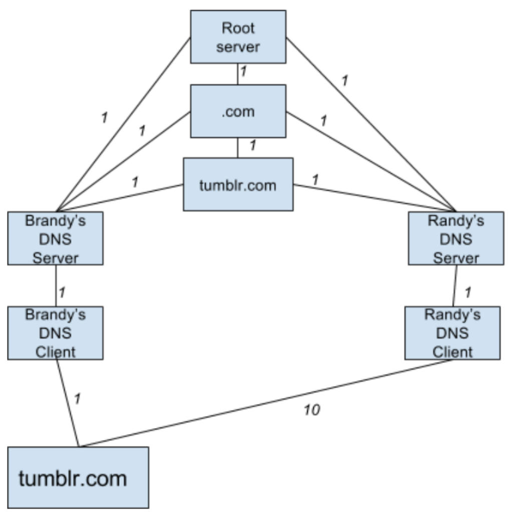
Suppose the network is configured as shown below:



Assume all hosts are attached to switch A and all DHCP servers are attached to switch D. If there is not enough information to answer the question, please indicate.

1. How many packets would Switch B see?
2. How many packets would Switch G see? **Solution:** Both would see H Discovery packets, HD Offer packets, H Request packet, and H ACK Packets.

9 DNS



Consider two siblings, Brandy and Randy, who are trying to access tumblr.com. Brandy issues requests from the US, while Randy requests the page from China. Assume that the latency between nodes is given by the edge weights on the topology above, and that neither siblings' local DNS server caches any data. The box labeled ".com" is the authoritative server for the TLD .com, and the box labeled "tumblr.com" is the authoritative server for the domain tumblr.com.

1. If iterative DNS resolution is being used, what is the total latency of a request for each sibling? **Solution:** 10 for Brandy, 28 for Randy

2. If recursive DNS resolution is being used, what is the total latency of a request for each sibling? **Solution:** 10 for Brandy, 28 for Randy

3. Based on the latencies given in the topology, is Randy being directed to a server in China or the United States? **Solution:** United States

4. Seeing as the latency to tumblr.com's server hurts Randy's performance, what could be done to improve his user experience? Assume that DNS servers can determine the geological region of a requestor. **Solution:** The authoritative DNS server could direct Randy to a tumblr server in China.

10 Congestion Control

For this problem, assume that the state is maintained throughout all the parts of the problems (i.e., assume the congestion control state for part 2 is the state resulting from part 1, and so forth). For each part, state what CWND is at the end of the specified event, expressed in terms of MSS. We are initializing from Slow Start and CWND=1 and SSTHRESH is 50. Below, ACK is an ACK for new data. You must use the rounding strategy from discussion where:

Instead of using CWND/2, use $\lfloor \frac{CWND}{2} \rfloor$

Instead of using 1/CWND, use $\frac{1}{\lfloor CWND \rfloor}$

Note that in this problem, and in this class, we do not care whether an ACK of new data ACKs one or multiple MSSs when calculating CWND. This is not the recommended approach in TCP implementations, but simplifies our presentation of TCP.

1. 9 ACKs are received. CWND = **Solution:** 10
2. 8 ACKs are received. CWND = **Solution:** 18
3. A timeout occurs. CWND = **Solution:** 1
4. 45 ACKs are received. CWND = **Solution:** 13 $\frac{3}{13}$
5. 10 ACKs are received. CWND = **Solution:** 14
6. 9 dupACKs are received. CWND = **Solution:** 16
7. 35 ACKs are received. CWND = **Solution:** 11
8. A timeout occurs. CWND = **Solution:** 1
9. 8 ACKs are received. CWND = **Solution:** 6 $\frac{1}{2}$
10. 2 dupACKs are received. CWND = **Solution:** 6 $\frac{1}{2}$
11. 4 ACKs are received. CWND = **Solution:** 7 $\frac{1}{7}$

11 Fair Queueing

Consider four flows traversing a single link. The four flows pass through a router implementing Fair Queueing before they reach the link. At their sources, the four flows are sending at the following rates:

Flow A: 3 Mbps
Flow B: 4 Mbps
Flow C: 5 Mbps
Flow D: 6 Mbps

There are no other flows on the link.

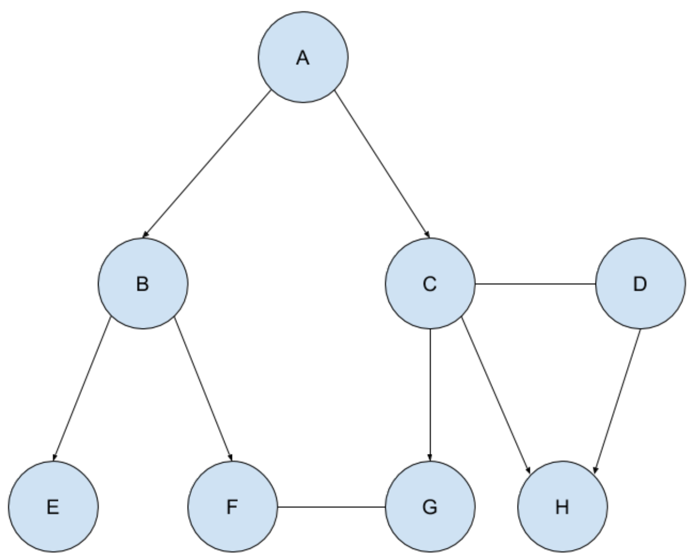
1. If the link has a capacity of 15 Mbps, how much bandwidth does each flow get after passing through the router? **Solution:** 3, 4, 4, 4

2. If the link has a capacity of 17 Mbps, how much bandwidth does each flow get after passing through the router? **Solution:** 3, 4, 5, 5

3. Now continue assume a capacity of 17 Mbps, but now the router drops packets from flow i with probability p_i instead of using fair queueing. What values of p_i should it use to achieve the same (fair) effective rates as from the previous part? **Solution:** 0, 0, 0, $\frac{1}{6}$

12 BGP

Consider the AS topology below:



Assuming that we are using standard Gao Rexford policies. Arrows represent provider→consumer relationships, while horizontal lines represent peering.

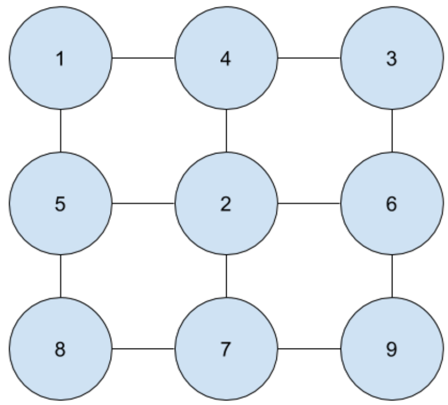
1. What is the path(s) from H to D, if any? **Solution:** H-D, or H-C-D

2. What is the path(s) from E to G, if any? **Solution:** E-B-A-C-G

3. What is the path(s) from E to D, if any? **Solution:** None

4. What is the path(s) from F to H, if any? **Solution:** F-B-A-C-H

Consider another AS topology below:



Assume that instead of Gao Rexford policy, we use a new policy.

An AS will export:

- A path to itself to all neighbors.
- All paths advertised by odd domains to even domains.
- All paths advertised by even domains to odd domains.

An AS will select:

- Choose routes advertised by even domains over those advertised by odd ones.
- Choose routes advertised by lower valued domains over higher valued ones.
- Never select routes that cause loops.

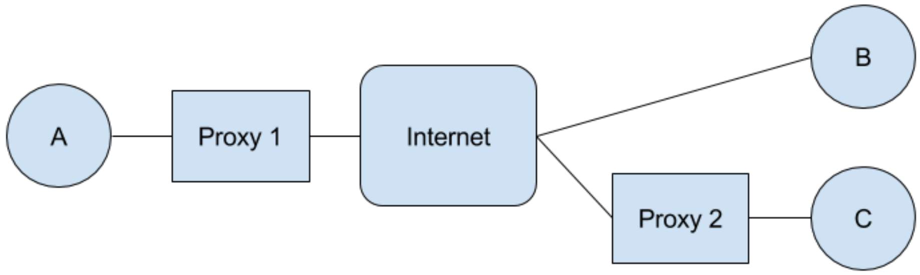
Using these new policies, answer the following questions:

1. What is the path(s) from 2 to 3, if any? **Solution:** 2-4-3

2. What is the path(s) from 1 to 9, if any? **Solution:** 1-5-2-6-9

3. What would happen if the third selection criteria was removed, and domain 2 tried to send to domain 6? **Solution:** A loop would occur: 2-4-1-5-2

13 HTTP



Consider the (abstracted) network topology above. Hosts A and C are connected to HTTP proxies that cache the results of the last 2 HTTP requests they've seen to improve performance. The proxy will perform any TCP handshakes or teardowns with the client and server concurrently. That is, when it gets a SYN from a client, it will respond and immediately send a SYN to the server, and similarly for other messages.

As an example, let's suppose that A sends a request to B. A sends a SYN to B, which is intercepted by the proxy. The proxy sees the request is going to B, and initiates its own TCP handshake with B, all the while completing the origin, separate handshake with A. By the time this has completed, there will be 2 TCP sessions. The first between A and the proxy, and the second between the proxy and B. When A sends a request, the proxy will forward it to B if it is not cached, or respond if it is cached. A similar process to the handshake is followed when A tears the connection down.

For the purposes of this problem, assume that the latency on each link is L, and the latency through the internet is I. Processing delay at all points and packet size is assumed to be negligible (don't consider transmission and processing delay). Assume that TCP connections use the 3 message teardown, and that data is not sent in ACK packets.

Suppose that Host A issues the following list of requests (in order):

- berkeley.edu to Host C
- eecs.berkeley.edu to Host C
- stanford.edu to Host B
- mit.edu to Host B
- stanford.edu to Host B
- berkeley.edu to Host C

1. What is the total time to complete all requests if they are issued one at a time? That is, each completes before the next is started with a separate TCP session (no need to wait for the session to be torn down).

   Solution: To set up a connection to B, A will initiate a TCP connection with P1, which will establish a connection to B. The total time to establish the connection is the time for the SYN to reach P1 plus the time it takes P1 to establish a connection to B. This is $L+2(2L+I)$, since the proxy will send the request from A right after sending the ACK.

   To set up a connection to C, A will initiate a TCP connection with P1, which will do so with P2 (who it thinks is C), which will do so with P. Similarly to before, the time this takes is $L+2(2L+I)$, since the longest delay is between the two proxies, thats all we need to consider. Since this time is the same for both servers, well call it S, the setup time.

In all cases, A will complete its handshake with its proxy before the proxy finishes its handshake with the server, and so the HTTP request will have already arrived and be sent immediately after. This means that the latency from A to P1 doesnt matter, just the latency from P1 to hosts B and C.

The latency from P1 to B is $2L+I$, well call this L_B.

The latency from P1 to C is $3L+I$, well call this L_C.

The first and second requests are different, and won't be helped by the cache. Therefore they take $S+2L_C+L$ time each.

The third and fourth requests are similar, taking $S+2L_B+L$ time each.

The fifth request hits P1s cache, only taking $2L+2L$ time.

The sixth request hits P2s cache, taking $S+2(L_C-L)+L$ time.

Therefore the total time is:

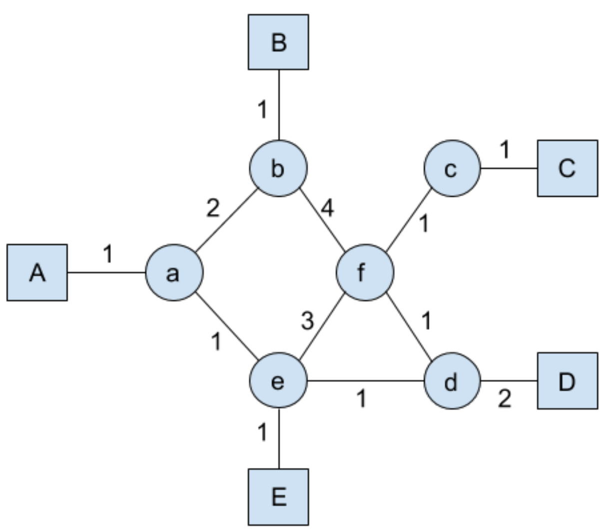
$$
\begin{aligned}
&2(S+2L_C+L)+2(S+2L_B+L)+4L+S+2(L_C-L)+L= \\
&2S+4L_C+2L+2S+4L_B+2L+4L+S+2L_C-2L+L= \\
&5S+6L_C+4L_B+7L= \\
&5(5L+2I)+6(3L+I)+4(2L+I)+7L= \\
&25L+10I+18L+6I+8L+4I+7L= \\
&58L+20I
\end{aligned}
$$

2. What is the total time to complete all requests if they are performed concurrently?

Solution: Since all the requests occur concurrently, there is no opportunity for caching by the proxies. Therefore the total time is just the max of all the times for the individual requests, which is the first request. It takes:

$$
\begin{aligned}
&S+2L_C+L= \\
&5L+2I+2(3L+I)+L= \\
&5L+2I+6L+2I+L= \\
&12L+4I
\end{aligned}
$$

14 Multicast



Consider the network topology above for both parts of this question. Hosts are squares with capital letters, routers are circles with lower case letters. Link cost is latency in seconds.

14.1 DVMRP

All parts of this section are cumulative. At the start, A and E are members of group 1. Assume that all NMR reports (non-membership reports) have been processed before each part.

1. A sends a message to group 1. After all the NMR reports have propagated, which links remain available (i.e. are used for forwarding packets) for (Source A, Group 1)? **Solution:** A-a, a-e, e-E
2. Suppose C adds itself to group 1. Immediately after this, who knows C joined group 1? **Solution:** c. Only the first hop router is notified at first.
3. E sends a message to group 1. Which links remain available for (Source E, Group 1)? **Solution:** A-a, a-e, e-E, d-e, d-f, c-f, c-C
4. B sends a message to group 1, is this possible? **Solution:** Yes, non members can send messages to a group.
5. What is the route that the packet takes from B to E? **Solution:** B-b-a-e-E
6. How long does it take for B's message to reach all members of group 1? **Solution:** 7 seconds, the path to C (B-b-f-c-C) is the longest.
7. Suppose that the pruning information expires. A, B and D form a new group 2. A then sends messages to groups 1 and 2, B sends a message to group 2, and D sends a message to group 1. What state does router f have? (Give a list of (Source x, Group y)). **Solution:** Original (Explicit (S, G) that didn't NMR): (A, 1), (D, 1). Correction/Lecture-Style ((S, G) that are NMR'd): (A, 2), (B, 2)
8. What message(s) took the longest time to send? **Solution:** B→D takes 7 seconds.
9. What message(s) took the least time to send? **Solution:** A→E took only 3 seconds.

14.2 CBT

All parts of this section are cumulative, but independent from the previous section. A is the core for group 1.

1. C decides to join group 1. What routers does its request go through? **Solution:** c, f, d, e, a
2. D decides to join group 1. What routers does its request go through? **Solution:** d
3. B sends a message to group 1. What long does it take to reach all members? **Solution:** 10 seconds. 4 seconds to reach A, and then another 6 to reach C from A (reaching D only took 5 seconds).
4. C sends a message to group 1. How long does it take to reach all members? **Solution:** 5 seconds. It takes 6 seconds to reach A, but A is the core and not a member. It takes 5 sec to reach D.
5. The link between A and a goes down, but no other changes in control plane state occur. Who can still send/receive messages to/from group 1, if any? **Solution:** Only members of group 1.

15 SDN

You are a chief network engineer at CS168DB, a company that specializes in databases. Currently, your networking infrastructure still relies on hardware switches that converge on routing with various L2/L3 protocols. You have tens of thousands of servers that need to communicate with each other. As the chief network engineer, you are thinking about making the switch to software defined networking.

1. What are your current control plane abstractions? **Solution:** None!
2. What are the required control plane's abstractions if you make the switch to SDN? **Solution:** There are three. The first is an abstraction of the forwarding model of switches, OpenFlow is an example. The second is an abstraction of network state, in the form of a global network view via a Network Operating System. The third abstraction is an abstraction of the specification for network behavior, which is used to specify the *goals* of an operator on an abstract view of the network.
3. What happens when a link goes down in your new SDN setup? **Solution:** The software switches connected to the link will communicate the topology change to the controller. The controller will then recompute routing state and disseminate that information to the routers.
4. Is OpenFlow equivalent to SDN? **Solution:** No.

16 TCP

Your scores for CS61A midterm 2 have just been released. You want to check your score on Gradescope and you know that you must connect to Gradescope over TCP but unfortunately your Mac does not have TCP support because Apple believes TCP is dated and useless. So you must manually manage your TCP connection.

16.1 Handshakes

Consider the following statements:

1. Packet has TCP ACK flag set
2. Packet has TCP SYN flag set
3. Packet has TCP FIN flag set
4. Packet has TCP RST flag set
5. Packet carries HTTP request
6. Packet carries HTTP response

You have to remember the process from establishment of the TCP connection to tearing the TCP connection down (which is initiated by the host) in order to connect to Gradescope. This process takes, in this case, seven ordered messages. For each of these packets, list which of the above properties apply (list all that apply). Assume that in this part, we are piggybacking HTTP messages on ACKs. We will expect you to understand both cases (with piggybacking, and without), but we will be explicit in which case to use.

(a) Packet 1 **Solution:** 2 (SYN)

(b) Packet 2 **Solution:** 1, 2 (SYN+ACK)

(c) Packet 3 **Solution:** 1, 5 (ACK+HTTP request)

(d) Packet 4 **Solution:** 1, 6 (ACK+HTTP response)

(e) Packet 5 **Solution:** 1, 3 (ACK+FIN)

(f) Packet 6 **Solution:** 1, 3 (FIN+ACK)

(g) Packet 7 **Solution:** 1 (ACK)

16.2 Data Transmission without Congestion Control

You send the TCP establishment handshake and receive a 20 Byte SYN packet back that has sequence number 2250 and advertised window of 1250 bytes. You send a HTTP get on your graded midterm and begin to receive packets.

Assume that the transmission delay is negligible.

Note: Without Congestion Control, TCP uses a constant sliding window, and retransmits the packet containing the next expected byte on a finite timeout or the 3rd duplicate ACK.

(a) Fill in the blanks in the table below. Assume that all packets arrive in order.

#	Packet Received	Payload Size	Sent on dupACK	Sent on timeout	ACK sent
1	D2270	225	No	No	A2495
2	D2495	225	No	No	A2720
3	D2945	225	No	No	A2720
4	D3170	225	No	No	A2720
5	D3395	225	No	No	A2720
6	D2720	225	Yes	No	A3620
7	D3845	225	No	No	A3620
8	D4070	225	No	No	A3620
9	D4295	225	No	No	A3620
10	D4520	225	No	No	A3620
11	D3620	225	Yes	No	A4745
12	D4745	225	No	No	A4970
13	D4970	225	No	No	A5195
14	D5195	225	No	No	A5420

(b) If the RTT of the link is 100ms and the timeout is initially 4 seconds, what is the total time needed for you to receive your graded midterm (the 14 packets in the above table)? Assume negligible transmission delay and negligible processing time, and that the estimates that go into the RTO remain constant during the events below.

Solution: There are 6 separate RTTs that contribute to the total delay.

These are from sending of:

- the first window of packets (1-4)
- D3395 due to the sliding window (5)

- the retransmit of D2720 (6)
- the next window of packets (7-10)
- the retransmit of D3620 (11)
- the last window of packets (12-14)

Thus the total delay is $6 \cdot RTT = 600ms$.