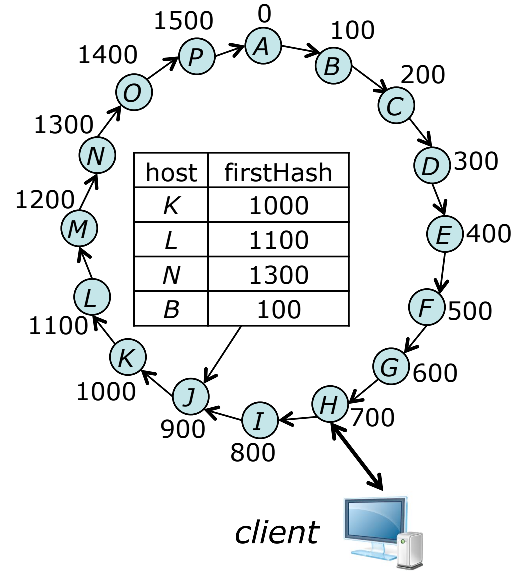
CSE 473 – Introduction to Computer Networks *Jon Turner*

Exam 2 Solutions

11/7/2012



1. (10 points). The diagram at right shows a DHT with 16 nodes. Each node is labeled with the first value in its range values (so for example, *B* is responsible for hash values 100-199. The routing table for node *J* is shown in the figure. Note that *J* has routes to the node that is 1 hop away, the one that is 2 hops away, the one that is 4 hops away and the one that is 8 hops away. Assume that all nodes have routing tables that are configured similarly.

   Suppose the client shown in the diagram sends a get request to node *H* with a key string of "flapjack", and that *hash*("flapjack")=513. List the servers through which this request would pass, assuming that the key string does not appear in any node's cache.

   It would go through nodes P, D and F before returning to H and then the client.

   What servers would the request pass through if the key string appears in node *M*'s cache?

   The same set.

   What servers would it pass through if the key string appears in node *D's* cache?

   P and D.

   Suppose that "flapjack" is requested frequently. Specifically, each DHT node receives a get request for "flapjack" about once per second. If the system is operated without caches, how many requests per second must the "responsible server" process?

   16 requests per second.

   If caching is enabled, and each cache entry expires 60 seconds after being placed in the cache, approximately how often does the responsible server receive a get request from another server? (Hint: how many other servers send directly to the "responsible server"?)

   There are 4 servers that send packets directly to the responsible server. These 4 servers will each send the responsible server a new request every 60 seconds, so the responsible server will receive a request about once every 15 seconds from another server (in addition to the one per second that it receives directly from clients).

2. (10 points). Consider a pipelined, reliable transport protocol that uses go-back-*N* with cumulative acknowledgments. Assume that timeouts trigger retransmissions (duplicate ACKs do not) and that the receiver does not maintain any receive buffer. If the one-way delay between the sender and receiver is 50 ms and every packet is 10,000 bits long, how big must the window be to allow the sender to send at a steady rate of 1 Gb/s under ideal conditions?

   RTT=.1 second, so a 1 Gb/s link sends 100M bits per RTT or 10K packets per RTT. So the window size must be at least 10,000 to support a 1 Gb/s rate.

   Suppose that approximately one packet in 100,000 is lost. If the sender uses a timeout of 500 ms and a window size of 20,000 packets, how often does sender experience a timeout? How many packets will it retransmit when a time out occurs?

   Assuming that the bottleneck link rate is 1 Gb/s, the sender can still only send 10K packets per RTT. After each loss, it takes half a second for the sender to detect the loss and all packets sent in that half second are effectively wasted (since the receiver discards them in go-back-N). But the window size limits the number of packets sent following the lost packet to 20K. So immediately after each loss, the sender sends 20K packets, pauses for .3 seconds then re-sends the first 20K packets before sending another 60K, at which point it loses another packet. So, the sender experiences a timeout every 1.3 seconds.

   If we do not assume a 1 Gb/s bottleneck link rate, the sender can send 20K packets per RTT. In this case, after sending the packet that gets lost, it again times out after half a second, before re-sending 20K packets plus 60K new ones (which takes 400 ms). So, the sender has a timeout every 900 ms.

   Assume that after the connection starts at time 0, the 100,000-th packet (call it *p*) is lost. At what time was *p* sent by the sender?

   At time 1 second if we assume a 1 G/bs bottleneck link rate, 0.5 seconds if we do not.

   At what time does the sender re-transmit *p*?

   1.5 seconds if we assume a 1 Gb/s bottleneck link rate, 1 second if we do not.

   What happens to the packets sent between the time *p* is sent the first time and the time it is retransmitted?

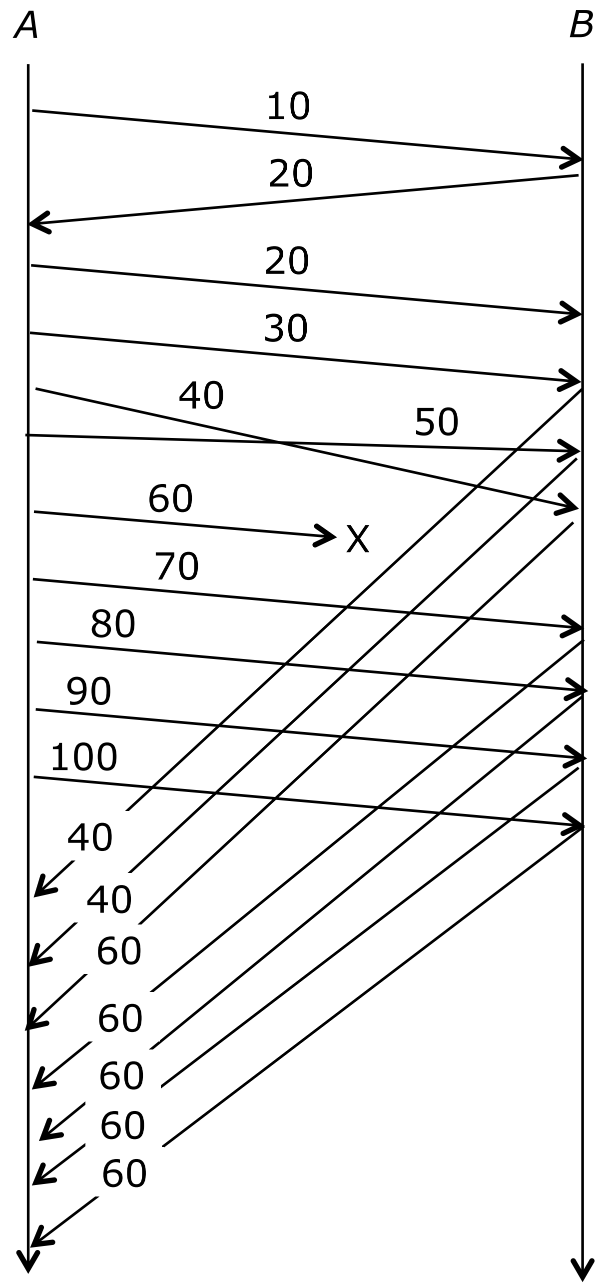
   They are discarded by the receiver and retransmitted by the sender

   Estimate the throughput for this connection, assuming one packet in 100,000 is lost.

   Assuming a 1 Gb/s bottleneck link rate, the receiver gets 80K new packets every 1.3 seconds, so the throughput is (8/13) Gb/s which is about 620 Mb/s.

   If we do not assume a 1 Gb/s bottleneck link rate, the receiver gets 80K new packets every .9 seconds, so the throughput is (8/9) Gb/s, which about 890 Mb/s.

3. (10 points). The diagram at right shows a TCP segment being sent from host *A* to host *B* and an ACK being returned. The numbers on the arrows are the sequence numbers of the data segments and the ACK numbers. Suppose that after receiving the ACK with ack number 20, *A* sends packets with sequence numbers 20, 30, 40, 50, 60, 70, 80, 90 and 100. Some time later, it receives ACKs with sequence numbers 40, 40, 60, 60, 60, 60, 60. (Assume that *A* sends no additional data segments in the meantime.) Complete the diagram in a way that is consistent with the given information and what you know about the way TCP behaves.



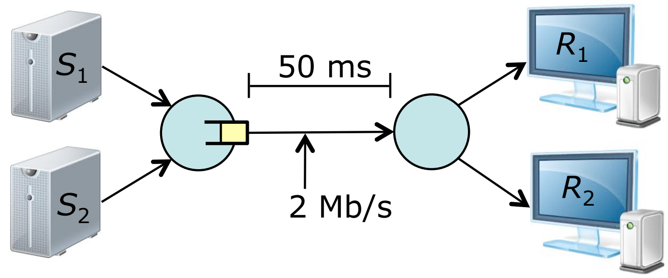
What sequence number would you expect to see in the next packet sent by *A*?

60

What ACK number would you expect in the next ACK? You may assume that all packets sent by *A* carry 10 bytes of data.

110

4. (10 points) The diagram at right shows two TCP senders at left and the corresponding receivers at right. Both senders use TCP Reno. Assume that the MSS is 1 KB, that the one-way propagation delay for both connections is 50 ms and that the link joining the two routers has a bandwidth of 2 Mb/s. Let $cwnd_1$ and $cwnd_2$ be the values of the senders' congestion windows and assume that $cwnd_1 = cwnd_2$. What is the smallest value of $cwnd_i$ for which the link joining the two routers stays busy all the time?



We need 200 Kbits per RTT to keep the link busy, or 100 Kbits per sender. That means 12.5 KB

Assume that the link buffer overflows whenever $cwnd_1 + cwnd_2 \geq 36$ KB and that at time 0, $cwnd_1 = 12$ KB and $cwnd_2 = 24$ KB. Approximately, what are the values of $cwnd_1$ and $cwnd_2$ one RTT later? Assume that all packet losses are detected by a triple duplicate ack.

6 KB and 12 KB

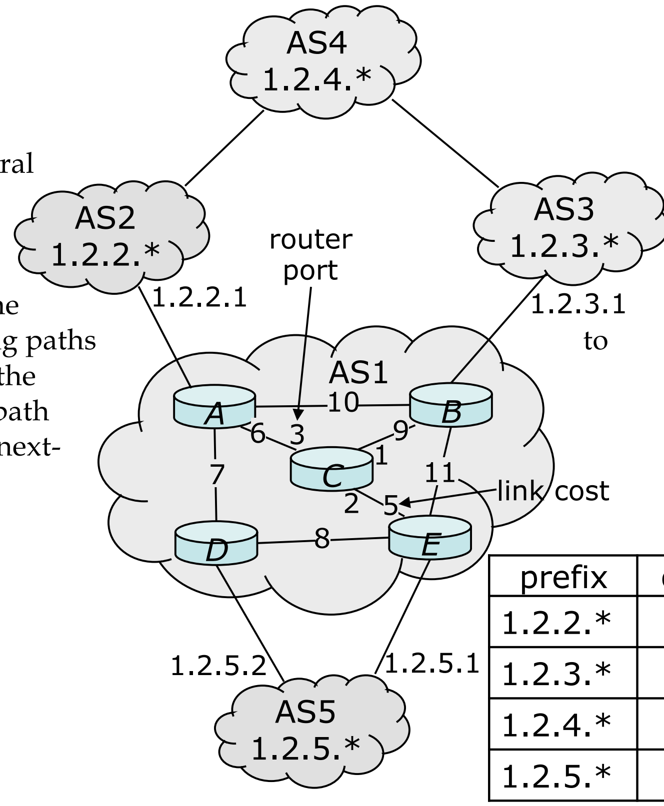
How many RTTs pass before $cwnd_1 + cwnd_2 = 36$ again? What are the values of $cwnd_1$ and $cwnd_2$ at this point?

After 9 more RTTs, we have $cwnd_1 = 15$ KB and $cwnd_2 = 21$ KB

Approximately, how many RTTs pass (in total) before $cwnd_2 - cwnd_1 < 2$ KB?

After 11 RTTs, the difference is 3 KB and it remains 3 KB for another 9 RTTs when the buffer fills again, at which point it becomes 1.5 KB. So approximately 21 RTTs pass before the difference in the congestion windows drops below 2 KB.

5. (10 points). The diagram below shows several autonomous systems and for each one, a prefix that it advertises to other ASs. Assuming that router *C* has IBGP sessions to all four gateway routers in AS1, list all the path vectors that it would receive describing paths to other ASs. For each path vector identify the gateway router from which *C* receives the path vector, the vector itself, and the associated next-hop address.



prefix	output
1.2.2.*	3
1.2.3.*	1
1.2.4.*	3
1.2.5.*	2

A AS2 1.2.2.1

A AS2, AS4 1.2.2.1

A AS2, AS4, AS3 1.2.2.1

B AS3 1.2.3.1

B AS3, AS4 1.2.3.1

B AS3, AS4, AS2 1.2.3.1

D AS5 1.2.5.2

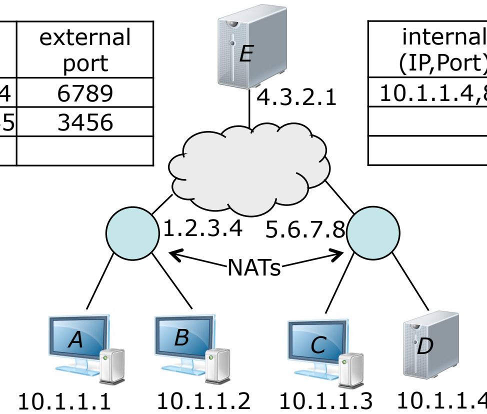
E AS5 1.2.5.1

The table at the bottom right of the figure represents a portion of the routing table at *C*. Complete the table, as it would be completed, assuming that AS1 uses OSPF in addition to BGP. For this part assume there are no policy constraints.

Suppose that AS1 is not intended to carry transit traffic. Explain how AS1 can use BGP to enable it to receive traffic from other ASs, while not allowing them to send traffic through it.

It advertises its own prefixes, but does not advertise prefixes belonging to other ASs.

internal (IP,Port)	external port
10.1.1.1,7654	6789
10.1.1.2, 2345	3456



internal (IP,Port)	external port
10.1.1.4,80	80

6. (10 points). The figure at right shows two residential networks with routers that implement NAT. Suppose host *A* is connected to the web server at host *E*.

   In the left-hand NAT table, add an entry that would allow *A* to communicate with *E*. You may choose any port numbers you like.

   Show the values of the address and port fields in the diagram below, for a typical packet sent by host *A*.

src adr	dest adr	src port	dest port
10.1.1.1	4.3.2.1	7654	80

Show the fields in the packet as it might appear when it reaches *E*.

src adr	dest adr	src port	dest port
1.2.3.4	4.3.2.1	6789	80

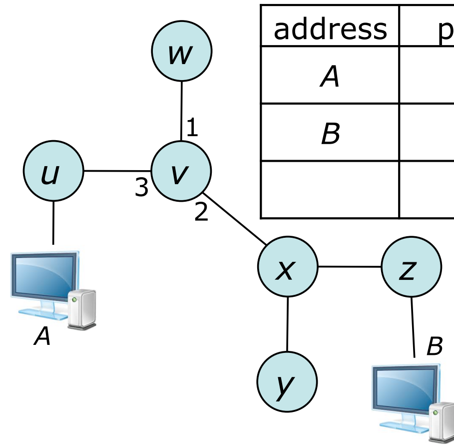
Suppose the user in the right-hand network wants to run a web server on host *D*. Add an entry to the right-hand table that would allow remote connections to the web server. Assume host *B* connects to the web server at *D*. Add an entry to the left-hand NAT table for this connection. Show the address and port field for a typical packet leaving host *B*, the fields in the same packet as it passes through the public internet, and the packet that is delivered to *D*.

src adr	dest adr	src port	dest port
10.1.1.2	5.6.7.8	2345	80

src adr	dest adr	src port	dest port
1.2.3.4	5.6.7.8	3456	80

src adr	dest adr	src port	dest port
1.2.3.4	10.1.1.4	3456	80

7. (10 points) The diagram at right shows a switched Ethernet LAN. Assume that the routing tables are empty initially. Show the contents of the routing table at switch v, after host A sends a packet to host B and host B sends a response. You may assume that A and B denote the MAC addresses of the corresponding hosts



address	port
A	3
B	2

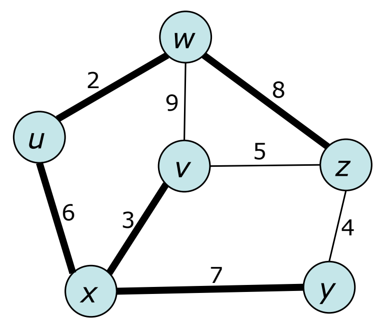
Suppose that at this point, a host at switch w sends a packet to some host other than B at switch z. Which switches in the network receive a copy of this packet?

All of them: u, v, w, x, y, z

If a host at switch y sends a packet to host A, which switches receive a copy?

u, v, x, y

The figure at right represents a switched Ethernet network, where the letters represent the switch identifiers and the numbers on the links represent costs used by the spanning tree algorithm. Assume that the switch identifiers are ordered alphabetically (so x comes before y, for example) and that all switches begin execution of the spanning tree algorithm at the same time and that all of them send an initial message to their neighbors, then receive the messages sent by their neighbors.



At this point, each switch has a tentative candidate for the spanning tree root. Complete the table below to show the candidate root for each switch.

u	v	w	x	y	z
u	v	u	u	x	v

In the diagram, highlight the edges that are included in the spanning tree when the algorithm completes (make the tree edges "heavier" by drawing a thicker line).

Suppose that all the links in the network operate at 1 Gb/s. Suppose we have a TCP connection between hosts on switches v and w, another between hosts on switches y and z and a third between hosts on switches u and x. If all hosts are trying to send data as fast as they can, what throughput would you expect each to achieve?

They would each get 333 Mb/s, since all three share the link from u to x.

What throughput could they achieve if the routing was not limited to the spanning tree?

They could each get 1 Gb/s if they used the directly connecting links.