15-744 Computer Networks (Fall 2010)
Homework 2

Due: Oct. 11th, 2010, 3:00PM (in class)

Name:
Andrew ID:

October 11, 2010

1. Route servers (e.g. those available at `http://www.traceroute.org/#Route%20Servers`) are BGP speaking routers with a publicly accessible interface. In other words, you can telnet to these routers and access their full BGP tables.

   `route-views.oregon-ix.net` is one such route server hosted at the University of Oregon. One use of this route server is that you can potentially get the route(s) from any AS X to any AS Y at an AS path level. You can also get the routes that *almost* any AS X would take to reach a given address prefix P.

   For this exercise, download the following routing table entries from the RouteViews server at:

   `http://www.cs.cmu.edu/~dga/15-744/S07/ps2/oix-full-snapshot-2007-02-14-1800.dat.bz2`

   Warning: this is 13 MB! You can use the `bzcat` command-line program to read it without decompressing it all. Although you only need a small part of this table to answer the following questions, you should to learn how to navigate large datasets like this efficiently. (Hint: write some code)

   RouteViews has archived their BGP tables since 1997. You can examine more of them at:

   `http://archive.routeviews.org`

   (a) CMU owns the address block `128.2.0.0/16`. Using this information, can you figure out the ISP CMU uses (the AS number of the ISP)? Using the `whois` service at `http://www.arin.net/whois/` or the `whois` command-line program, determine who this AS number actually corresponds to (the name of the ISP). Note: some address blocks allocated pre-CIDR appear without the netmask in the table; i.e., CMU's address block appears as `128.2.0.0`. Three, two, and one trailing 0 octets imply a class A (/8), class B (/16), or class C (/24) network, respectively.

   (b) Print the best AS route from the route server to CMU.

   (c) What is the AS number of MIT? List all providers of `mit.edu` that you can infer from the table. (Hint: MIT is one of the few class A networks. You can use `nslookup` to get the IP address for a host at MIT)

   (d) Some of the routes to MIT repeat its AS number multiple times. Why would they do that? What does this tell you about the upstream provider in those paths?

   (e) Find the first "Class C" CIDR address in the table (address prefix $\geq$ 192.0.0.0). How many class C networks does this address correspond to? What is the maximum number of routing table entries that this single CIDR address saves? Why is it that we can only infer the maximum, and not the actual, number of addresses that this CIDR address saves?

   You can get more information if you log into this route server by executing:

   `telnet route-views.oregon-ix.net`

Run `sh ip bgp` at the prompt and you get the entire BGP table, shown one screen after another (much like when you execute `more`). In general you can type `sh ip BGP ?` for help on the possible extensions to the `sh ip bgp` command. For example, you can use the help to figure out that `sh ip bgp 12.0.0.0` will give you all the routes from `oregon-ix` to `12.0.0.0/8`.

Solution:

(a) **6 points.** The AS number of CMU is 9. The AS number of the ISP CMU uses is 5050. This corresponds to PSC.

(b) **6 points.** The AS path is: 1239 5050 9.

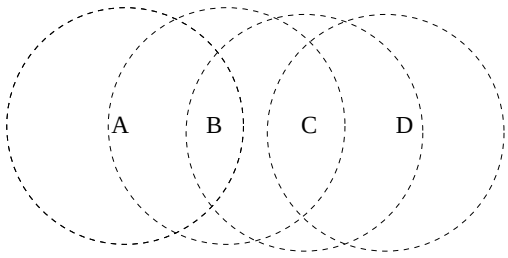
(c) **6 points.** The AS number of MIT is 3. The providers are AS 10578 (Harvard University) (this could be a sibling relationship or a mutual backup), AS 3356 (Level 3 Communications), AS 174 (Cogent), and AS 1239 (Sprint).

(d) **6 points.** Several of the routes through Cogent (174) have AS 3 duplicated. This is one way to make AS paths longer so that they are less likely to be chosen via BGP's shortest path preference. This is most likely because MIT's link to Cogent is a backup link.

(e) **6 points.** 192.0.32.0/20 is the first CIDR address in the "Class C" range of addresses. A class C network is a /24, so this corresponds to $2^4 = 16$ class C networks. This saves up to 15 routing table entries compared with when we had classful networks, since we only need one entry for up to 16 different class C networks. (In the CIDR world, there could be up to $2^{12} = 4096$ networks in here.)

We can't infer the actual number of addresses this CIDR address saves because we don't know how many of those class C networks are actually in use.

2. Srini, excited to setup a new adhoc wireless network on campus, uses the following topology with nodes *A*, *B*, *C*, and *D*. The dotted lines indicate the range of the wireless transmission from each node. For example, *C* is within transmission range of nodes *B* and *D*. In the following question, you will determine the outcome of transmissions that Srini has scheduled to take place in order to test his new network. Also assume that packets are never rescheduled for transmission, i.e., if a packet did not succeed the first time the node does not bother to retransmit it.



(a) In the first set of scheduled transmissions, carrier sense is enabled, but Srini has disabled RTS/CTS. Each transmission has a source, destination, associated relative start time, and total transmission time. For each transmission, briefly state the result and relate it to other transmission numbers if needed to justify your answer. **In this section, transmissions are never rescheduled.** A result would be "success" or "prevented by ...something..." (8 points)

#	Src	Dst	Start Time	Duration	Type	Result
1	A	B	0	15	DATA	
2	C	B	5	10	DATA	
3	C	D	20	20	DATA	
4	B	C	22	8	DATA	
5	B	A	30	10	DATA	

Solution:

#	Src	Dst	Time	Length	Type	Result
1	A	B	0	15	DATA	Collision with 2 (hidden terminal)
2	C	B	5	10	DATA	Collision with 1 (hidden terminal)
3	C	D	20	20	DATA	Successful
4	B	C	22	8	DATA	Prevented by carrier sense from 3
5	B	A	30	10	DATA	Prevented by carrier sense from 3 (exposed t.)

(b) After the first series of transmissions, Srini enables RTS/CTS on each node and reschedules the same transmissions from *part a* with the same start and transmission times. For each scheduled transmission give the outcome **and** any additional transmissions that may result from them. Make sure to specify the type of packet using DATA, RTS, or CTS. Assign new transmission numbers, but consider the times of RTS/CTS packets to be negligible (do not assign times). You may not need all of the blank lines.

(Fill in the table on the next page) (7 points)

#	Src	Dst	Time	Length	Type	Result
1	A	B			RTS	Success
2	B	A			CTS	Success
3	A	B	0	15	DATA	Success
4	C	B	5	10	DATA	Prevented: heard CTS#2

Solution:

#	Src	Dst	Time	Length	Type	Result
1	A	B			RTS	Success
2	B	A			CTS	Success
3	A	B	0	15	DATA	Success
4	C	B	5	10	DATA	Prevented: heard CTS (#2)
5	C	D			RTS	Success
6	C	D			CTS	Success
7	C	D	20	20	DATA	Success
8	B	C	22	8	DATA	Prevented: heard RTS (#5)
9	B	A	30	10	DATA	Prevented

3. Xi sets up a wireless AP (access point) at home, and finds the performance to be much poorer than he expected. In this question, you are going to help him figure out possible reasons.

   (a) First, Xi finds that since the access point is open, some neighbor has also associated with the AP. While Xi is close to the AP and can receive data at 11Mbps, the neighbor has a much lower signal strength and thus can only receive data at 1Mbps. Assume the AP always has full-size (1500 bytes) packets to send to both Xi and his neighbor, and it uses round-robin to schedule between the two transmissions. If we ignore all overheads (no headers or preambles, no packet loss, no acknowledgements), what is the throughput Xi will get? What if Xi can receive data at 54Mbps?

   Solution:
   At 11Mbps, the throughput is $\frac{1}{1+\frac{1}{11}} = 0.917$Mbps.
   At 54Mbps, the throughput is $\frac{1}{1+\frac{1}{54}} = 0.982$Mbps.

   (b) After finding this out, Xi has used several techniques (enable security, enable MAC address filtering) to prevent his neighbor from using his access point. In the next two questions, we will look at the difference between physical data rate and the TCP throughput.

   Here is how the 802.11 MAC protocol works with only one sender and receiver in the network: the sender would sense the channel before transmitting, and if the channel is clear and has been clear for DIFS (Distributed Coordinate Function Interframe Space) time, the sender would send out the frame. After receiving the frame, the receiver would wait for SIFS (Short Interframe Space) time and send out a link-layer acknowledgement. 802.11 protocol also requires a preamble to let the receiver to synchronize with the sender and to be notified that the data is on its way, and the preamble is transmitted at the lowest data rate.

   Assuming Xi is using 802.11b wireless devices, here are the parameters for 802.11b: preamble take $192\mu s$, DIFS = $50\mu s$, SIFS = $10\mu s$, payload size = 1500 bytes, header size = 30 bytes, TCP ACK frame size (including all headers) = 70 bytes, link layer ACK size = 14 bytes. So what is the TCP throughput if Xi can receive at 11Mbps physical data rate (i.e., the payload, headers, and link layer ACKs are transmitted at 11Mbps)? In this problem, you can assume the wireless link is the bottleneck, and ignore TCP slow start, congestion control, round-trip time, delayed ACK.

   Solution: The total time for a TCP exchange is: $(50 + 192 + 30 * 8/11 + 1500 * 8/11) + (10 + 192 + 14 * 8/11) + (50 + 192 + 70 * 8/11) + (10 + 192 + 14 * 8/11) = 2072\mu s$. Useful data size is 1500 bytes. Thus the effective data rate is $1500 * 8/2072 = 5.8$Mbps.

   (c) Xi is now using 802.11g devices, with the following parameters: preamble take $20\mu s$, DIFS = $34\mu s$, SIFS = $16\mu s$, payload size = 1500 bytes, header size = 30 bytes, TCP ACK frame size (including all headers) = 70 bytes, link layer ACK size = 14 bytes. So what is the TCP throughput if Xi can receive at 54Mbps?

Solution: The total time for a TCP exchange is: $(34 + 20 + 30 * 8/54 + 1500 * 8/54) + (16 + 20 + 14 * 8/54) + (34 + 20 + 70 * 8/54) + (16 + 20 + 14 * 8/54) = 421.2\mu s$. Useful data size is 1500 bytes. Thus the effective data rate is $1500 * 8/421.2 = 28.5$Mbps.

4. Harry Bovik is given the responsibility of configuring the packet queuing component of a new router. The link speed of the router is 100 Mbit/s and he expects the average Internet round-trip time of connections through the router to be 80ms. Harry realizes that he needs to size the buffers appropriately.

   You should assume the following:

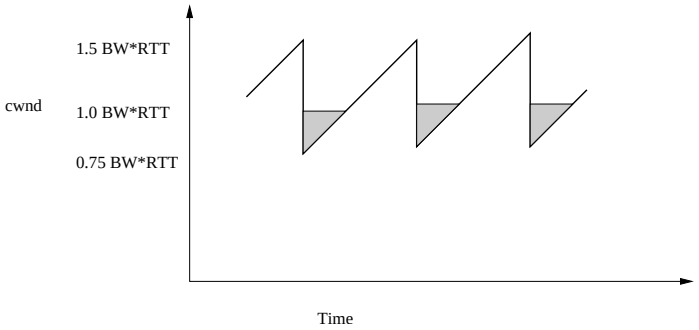
   - You're dealing with exactly one TCP connection.
   - The source is a long-running TCP connection implementing additive-increase (increase window size by 1 packet after an entire window has been transmitted) and multiplicative-decrease (factor-of-two window reduction on congestion).
   - The advertised window is always much larger than the congestion window.
   - The loss recovery is perfect and has no impact on performance.
   - The overhead due to headers can be ignored.

     Harry argues that because the average RTT is 80ms, the average one-way delay is 40ms. Therefore, the amount of buffering he needs for high link utilization is 100 Mbit/s * 40 ms or 500 KBytes.

   Question: approximately what bandwidth will TCP achieve with this buffering?

Solution: 10 points.

The buffering is not enough for TCP to fill the link. The window size with this buffering will go between $1.5BW * RTT and 0.75BW * RTT$.



The above graph shows the rough behavior of the congestion window over time. The overly simplified analysis:

During $\frac{1}{3}$ of the connection, the window will be too small, using an average of $(0.75 + 1)/2 = \frac{7}{8}$ of the capacity. During the other two thirds of the time, it will fill the link. The total utilization will be $\frac{23}{24}$ of the capacity, or about 95.8 Mbit/sec.

This analysis is overly simple, because it doesn't take into account the fact that the RTT increases by one packet's worth each RTT when we're sending a larger window than the buffer can handle. Exact answer:

In going from a window of 750 packets to a window of 1500 packets, the source sends:

$$\sum_{i=750}^{1500} i$$

packets. During this same amount of wall-clock time, an optimal scheme could have transmitted

$$\sum_{i=750}^{1500} i + \sum_{i=1}^{249} i$$

packets. (The missing packets from the underutilized periods). Or:

$$\begin{aligned}
\text{throughput} &= \frac{\sum_{i=750}^{1500}}{\sum_{i=750}^{1500} + \sum_{i=1}^{249}} \\
&= \frac{1}{1 + \frac{\sum_{i=1}^{249}}{\sum_{i=750}^{1500}}} \\
&= \frac{1}{1 + \frac{\frac{249*250}{2}}{\frac{1500*1501-749*750}{2}}} \\
&= \frac{1}{1 + \frac{31125}{844875}} = 0.9645 = 96.45 \text{ Mbit/sec}
\end{aligned}$$