New York University
Computer Science Department

G22.2620-001, Fall 07 (Li) **Problem Set 2** Due: Oct 15, 2007

---

1 Understanding BGP with routing table dumps

In this problem, we examine the BGP table snapshot taken at a measurement Cisco router belonging to the Oregon Route view project (**http://www.routeviews.org/**). This router peers with multiple ASs at multiple locations to obtain a large list of advertised routing entries.

Download the compressed snapshot at **http://www.news.cs.nyu.edu/classes/fa07/snapshot-1-22-07.bz2**. This snapshot is taken at **route-views.oregon-ix.net** whose list of peering ASs can be found here (**http://www.routeviews.org/peers/**).

You can directly examine the contents of a ***.bz2** file using command **bzcat** without saving the uncompressed file.

1. Find the routing table entry for NYU's campus network. What is NYU's Autonomous System (AS) number? [1]

2. How many different routes does this Oregon router know to reach NYU's network? What is the AS number of the next hop router to reach NYU network as specified by the best route? How long (in terms of number of ASs) is this path?

3. Use the immediate AS numbers preceding NYU's network to infer the complete list of ISPs that are NYU's network providers. (Hint: you need to use the **whois** command to obtain ISP names from AS numbers.) In the paths where AS 7018 appears as the upstream AS to NYU, why does the path end with duplicate NYU AS numbers? What is the likely relationship between this AS and NYU?

4. Use the **traceroute** command to examine the route from some machine at NYU to the Oregon router **route-views.oregon-ix.net** where the snapshot is taken. Attach your traceroute output. What is the sequence of ASes from NYU to the Oregon router? Is this route the same as the chosen best route from the Oregon router to NYU? Why might the forward and reverse routes be different?

5. Some routing entries in the dump has CIDR style addresses (w.x.y.z./m). Find the first CIDR network address belonging to the traditional class C addresses (i.e. addresses greater than 192.0.0.0) How many traditional class C addresses can this CIDR address potentially aggregate and hence save in terms of the amount routing entries? Is this amount of aggregation guaranteed?

[1] You can confirm you answer using the **whois** command. IP address and AS numbers are managed by 3 organizations, RIPE (Reseaux IP Europeans), ARIN (American Registry for Internet Numbers), APNIC (Asian Pacific Network Information Center). To query an AS number belonging to a US organization, do **whois -h whois.arin.net** < *ASnumber* >.

6. Suppose you have a series of routing table dumps from 1997 onwards. If you are only given each of the following information in the snapshot, what can you infer about the evolution of the Internet? Try to come up with as many interesting ways to quantify your conclusion as possible.

   (a) Only the destination network address and mask
   (b) Only the lines marked `*>`.
   (c) Only the paths, with the best next hops marked.

2 Inferring inter-AS relationships

We know that the commercial agreements between different administrative domains are reflected in BGP's operations. In general, a network will re-advertise its customer routes to its peers and providers, but will not readvertise routes obtained from a peer to other peers or providers. Knowing these rules and a view of a default-free routing table, you will deduce relationships between AS pairs in this problem.

In the paper "On inferring automonous system relationships in the Internet"[2], Gao points out that AS paths in BGP routing tables are typically "valley free", i.e. they observe the following patterns:

- A series of customer-provider links (an *uphill* path)
- A series of provider-customer links (a *downhill* path)
- An uphill path followed by a downhill path.
- An uphill path followed by a peering link
- An peering link followed by a downhill path
- An uphill path followed by a peerig link, followed by a downhill path

However, it is clear from an AS path where lies the "top of the hill". Gao uses the intuition that a provider tends to be larger than its customers and the bigger an AS, the larger its degree in the AS graph. Hence, a reasonable heuristic for picking out the top provider in an AS path is to find the one with the highest degree.

1. Produce the complementary cumulative distribution function (CCDF) of AS degree based on the route view data from Problem 1. In your CCDF plot, the y axis is the fraction of AS's with degree $\geq x$, on a log-log scale. Attach the CCDF plot as well as a table of the top 10 AS's and their degrees. Does the degree distribution in your CCDF reflect the actual degrees of each AS?

   (Do not count any link that goes from an AS to itself. You should also consider *all* AS paths given in the table and not just the best path for each prefix.)

2. Infer the transit relationship between pairs of ASes. This is a two step process. First, you need to scan all AS paths. For each AS path (e.g. $ABCD$), pick out the AS with highest degree (e.g. C) as "top of the hill", and note down the transit relationships implied by this path ($A \rightarrow B$, $B \rightarrow C$, $D \rightarrow C$). Second, designate certain pairs of ASes (e.g. AB) as having sibling relationships if they transit for each other (e.g. $A \rightarrow B$ and $B \rightarrow A$) as witnessed from different AS paths.

   List the transit or sibling relationships between consecutive pairs of ASes in the following paths.

   - 3333 3356 6517 12
   - 3277 3216 3549 6517 32473

[2] A copy of the paper is available at **http://www-unix.ecs.umass.edu/˜lgao/ton.ps**

- 2493 3602 1239 3356 32473

(Note that both the transit or sibling relationships that you have categorized *might* actually be a peering relationship.)

3. Gao's algorithm for picking the top provider is only a heuristic. Why might this heuristic be wrong sometimes?

3 Setting router buffers for TCP

Alice works for a large ISP that has recently installed a router with $\mu = 40$ gigabits/sec link speed. She comes you for help on how to configure the router's buffer size correctly to ensure high link utilization.

1. You decide to start off with some simplifying assumptions. Assume there is exactly one TCP connection with RTT 100 ms traversing your router. Assume this TCP connection is long running and always in its AIMD congestion-avoidance phase (i.e. you are going to ignore the effects of slow-start and timeouts).

   Show that if Alice sets the router's buffer size to be the product of bandwdith and roundtrip delay, the single TCP connection can always drive the router's link to 100% utilization. How much memory does this amount of buffering take?

   Hint: You can observe that at every point, the throughput of the single TCP flow is equal to min(W/RTT, μ), where W is the current TCP congestion window

2. Alice is shocked by your calculation and asks what link utilization will result if she only allocates a very small amount of buffering? Explain your result.

3. Generalize your calculation for 1) and 2) to show how the average link utilization (U) varies as a function of r, the ratio of the amount of buffering, B, to the bandwidth-RTT product. Plot the function $U(r)$.

   Hint: You might want to break each AIMD epoch into two stages. In the first stage, the congestion window W increases linearly from W_{min} to the point that TCP achieves 100% link utilization. In the second stage, the link utilization simply remains at 100% (unaffected by the increasing W). If you can figure out the fraction of time the TCP connection remains in the first stage vs. the second stage, you can do a weighted average to figure out the average link utilization during each AIMD epoch.

4. Your solution in (3) is derived with the assumption of a single TCP. What do you think are the best and worst case link utilizations for a given buffer size when there are a large number of TCP connections going through your router?