New York University
Computer Science Department

G22.2620-001, Fall 07 **Problem Set 2** Solution by Hui Zhang

---

1 Understanding BGP with routing table dumps(10)

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]

   Answer: NYU's AS number is 12.

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?

   Answer: This Oregon router knows 45 different routes to reach NYU's network.
   The best next hop router to reach NYU is 216.140.8.59, and its AS number is 6395.
   The path contains 2 AS hops (6395→12).

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?

   Answer: NYU's ISPs are listed in Table 1. Duplicating NYU's AS number in a path discourages the use of the path by making it longer. AS 7018 is likely to be NYU's secondary (or backup) provider for the multi-homed NYU network. NYU prefers others to route traffic via its primary provider than the backup by making the backup route appear longer.

[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** $< ASnumber >$.

Table 1: List of NYU's Network Providers

AS number	ISP name
6395	Broadwing Communications Services, Inc.
6517	Yipes Communications. Inc.
3754	NYSERNet
7018	AT&T WorldNet Services

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?

   My traceroute output is as follows:

   1 WWHGWA-VL45.NET.NYU.EDU (128.122.81.2) 0.583 ms 0.448 ms 0.876 ms
   2 NYUGWA-GI2-1.NET.NYU.EDU (128.122.1.50) 1.617 ms 0.605 ms 0.481 ms
   3 EXTGWB-VLAN15.NYU.NET (192.76.177.89) 0.567 ms 0.477 ms 0.478 ms
   4 nyc-gsr-nyu.nysernet.net (199.109.4.21) 0.570 ms 0.475 ms 0.480 ms
   5 alb-7600-nyc-gsr.nysernet.net (199.109.7.97) 3.629 ms 3.667 ms 3.545 ms
   6 buf-7600-alb-7600.nysernet.net (199.109.7.9) 9.621 ms 9.656 ms 9.533 ms
   7 199.109.11.2 (199.109.11.2) 22.813 ms 22.195 ms 22.447 ms
   8 so-4-3-0.0.rtr.kans.net.internet2.edu (64.57.28.36) 309.163 ms 282.128 ms 269.706 ms
   9 64.57.28.57 (64.57.28.57) 333.994 ms 367.136 ms 335.333 ms
   10 so-3-0-0.0.rtr.losa.net.internet2.edu (64.57.28.44) 78.700 ms 78.920 ms 96.076 ms
   11 vl-101.xe-0-0-0.core0-gw.pdx.oregon-gigapop.net (198.32.165.65) 100.548 ms 100.550 ms 100.458 ms
   12 100.ge-5-0.core0.eug.oregon-gigapop.net (198.32.163.5) 102.817 ms 102.960 ms 102.792 ms
   13 vl-110.uonet8-gw.eug.oregon-gigapop.net (198.32.163.131) 102.937 ms 102.958 ms 102.907 ms
   14 ge-5-2.uonet1-gw.uoregon.edu (128.223.3.1) 103.162 ms 103.136 ms 103.060 ms
   15 ge-0-2.route-views.uoregon.edu (128.223.51.103) 103.474 ms * 103.510 ms

   By mapping the IP address in the above traceroute output to the IP prefix in the snapshot, I get the following ASes sequence from NYU to the Oregon router:

   ```
   12(NYU, 1-3 hop)
   3754(NYSERNet, 4-7 hop)
   11537(Internet2, 8-10 hop)
   4600(Oregon, 11-13 hop)
   3582(Oregon, 14-15 hop)
   ```

   This route is not the same as the best route from the Oregon router to NYU which is 3582→6239→12. The forward and reverse routes could be different because ASes choose routes using different local policies that are not simply the shortest AS path length.

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?

   The first CIDR-style address greater than 192.*.*.* is 192.0.32.0/20.

This CIDR address can potentially aggregate 16 traditional 24-bit class C prefixes.
However, this amount of aggregation is not guaranteed if some subset of the addresses have multi-home with different providers to advertise their own routes. For example, the routing table snapshot contains 192.0.36.0/24 which is a sub-prefix of 192.0.32.0/20. Routers will forward packets addressed to 192.0.36.0/24 according the routing entries of 192.0.36.0/24 instead of 192.0.32.0/20 because of the longest matching prefix rule.

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

   Answer: With this information, we get a complete list of destination networks and masks. This time series data could tell us several things:

   - The growth of classless versus classful prefixes as CIDR became more prevalent from 1997 onward.
   - How much of IP space has actually been allocated and is in use; if there are blocks of the 32 bit address space that "come online" (because, for example, they simply weren't represented at a previous point but later, a routing entry appears for them), we can infer growth of the Internet.
   - This information can also tell us something about deaggregation: if an address block was previously represented by one routing entry and is now split into several entries with longer prefixes, then we can infer that addresses have been resold, or new companies or providers have come online, etc.

   (b) Only the lines marked *>. Answer: The best routes with corresponding next-hops and AS path can tell us several things:

   - We could get some of the same information as in (a), because the sheer quantity of lines we have been given tells us something about the size of the Internet and the number of distinct networks (since there is one of these lines per routing entry).
   - We could estimate the total number of unique AS's and growth of AS usage.
   - We could also measure average, maximum and minimum AS path lengths as a function of time.
   - This is a list of all of the "best" routes, that is, the routes that the Oregon router actually would have used to forward traffic. Since the lines we have been given are associated to next hop IP addresses, and since we can map IP addresses to AS, we can determine which ASes have grown in connectivity and which are likely to carry traffic and over which part of their IP space.

   (c) Only the paths, with the best next hops marked.

   Answer: Under this assumption, we are given no information about the prefixes, but we can still determine a number of things:

   - First, note that this is actually more information than we use in question 2, to calculate degrees and provider-customer, sibling-sibling relationships. So we can do everything we did/will do in question 2

- We can also do much of what we did in parts (a) and (b) in this question (since we again can tell how many best paths there are, which tells us something about the number of routing entries, which tells us something about the growth of the Internet)

2 Inferring inter-AS relationships(10)

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. Table 1

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.)

   Answer: Figure 1 shows the CCDF plot of AS dgree. Table 2 lists the top 10 ASes and their degrees in my calculation. The AS degrees distribution in CCDF plot implies that only a small number of ASes have large degrees. Majority of the ASes have fairly small degrees, e.g. less than 10.

   The degrees distribution in this CCDF plot does not reflect the actual degrees. Because it only contains the routes seen by a specific router route-views.oregon-ix.net with a limited number of vantage points. There may be other AS connections that this router is not aware of.

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

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

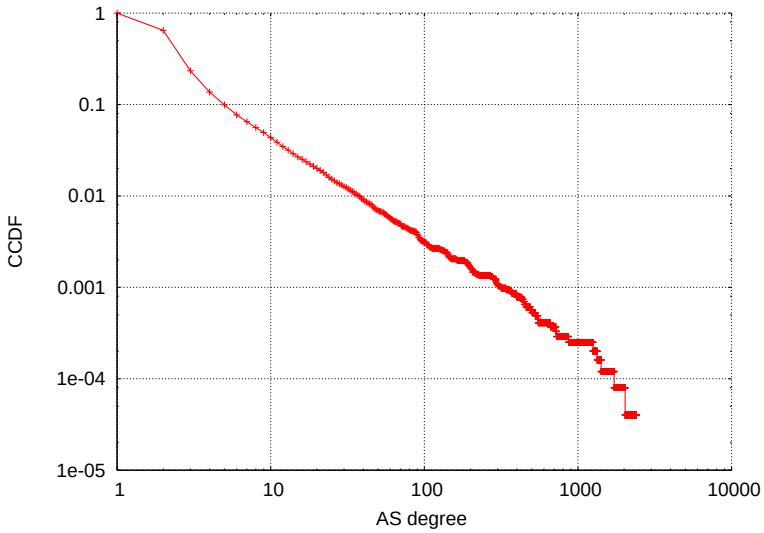

Figure 1: CCDF of AS degrees

Table 2: Top 10 ASes in term of degrees

AS number	Degree	ISP name
701	2396	MCI Communications Services, Inc. d/b/a Verizon Business
7018	2028	AT&T WorldNet Services
1239	1719	Sprint
174	1414	Cogent Communications
3356	1333	Level 3 Communications, LLC
209	1252	Qwest
3549	868	Global Crossing
4323	726	Time Warner Telecom, Inc.
6461	718	Abovenet Communications, Inc
7132	661	AT&T Internet Services

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
- 2493 3602 1239 3356 32473

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

Answer: `AS:degree  3333:143  3356:1333 6517:184  12:4`
`Relationship 3333 ---> 3356 <--- 6517 <--- 12`

```
AS:degree   3277:100  3216:183  3549:868  6517:184  32473:2
Relationship 3277 ---> 3216 ---> 3549 <--- 6517 <--- 32473

AS:degree   2493:1    3602:97   1239:1719 3356:1333 32473:2
Relationship 2493 ---> 3602 ---> 1239 <--> 3356 <--- 32473
```

Note that even though AS 1239 has a bigger degree than AS 3356, we categorize their relationship as siblings because there exists another route whose "top-of-the-hill" AS (say X) has a bigger degree than AS 1239 and the route contains a path segment $X...3356\ 1239$, resulting in the inference that $1239 \rightarrow 3356$.

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

   Gao's heuristic is wrong when the top provider isn't the AS with the highest degrees. There can be secnarios where some ASes connect with many other ASes despite the fact that they are not top-level AS. Consider the company Internap, which is a customer that tries to get its customers better performance by buying many transit providers themselves. Similarly, a small network may peer with a lot of other small networks, but this connectivity does not imply anything about position in the AS hierarchy.

3 Setting router buffers for TCP(10)

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 $min(W/RTT, \mu)$, where W is the current TCP congestion window

   Answer:

   The maximum number of packets (in terms of bytes) a connection can keep outstanding in the network is equal to bandwidth-RTT product plus the bottleneck router's buffer size. Therefore, the maximum congestion window size of the single TCP connection $W_{max} = \mu \cdot RTT + B$, where B is the buffer size.

   According to the Hint, the smallest possible TCP flow throughput is W_{min}/RTT. Hence if $W_{min}/RTT = \mu$, the TCP flow will keep the link fully utilized all the time. Therefore, we obtain $W_{min} = \mu \cdot RTT$.

   Because $W_{max} = 2 * W_{min}$, $B = 2 * W_{min} - \mu \cdot RTT = \mu \cdot RTT$.

   The amount of buffering in this case is $B = \mu \cdot RTT = 50Gbps \times 100ms = 512MB$.

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.

   Answer: If Alice allocates a very small amount of beffering, then $W_{max} \approx \mu \cdot RTT$. Since $W_{min} = W_{max}/2 = \mu \cdot RTT/2$, the minimum link utilization would be 50%. The maximum link utilization is 100%.

   Since congestion window size increases linearly with time from W_{min} to W_{max}, the router's average link utilization (U) can be calculated as $U = (U_{min} + U_{max})/2 = (1 + 0.5)/2 = 75\%$.

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.

   Answer: In this general case, $W_{max} = \mu \cdot RTT + B = (1+r)\mu \cdot RTT$ and $W_{min} = (1+r)\mu \cdot RTT/2$. In the first stage, the congestion window size will increase from W_{min} to $\mu \cdot RTT$. The link

utilization grows linearly in this stage, hence the average utilization can be calculated as the average of the link utilization at the start of the stage and the utilization at the end:

$$U_{first} = \frac{\frac{W_{min}/RTT}{\mu} + \frac{\mu \cdot RTT/RTT}{\mu}}{2} = \frac{r+3}{4}$$

In the second stage, the windows size will increase from $\mu \cdot RTT$ to W_{max}, the link utilization remains as 100%.

$$U_{second} = 1$$

The ratio of the amount of time the TCP flow spends at each of the two stage is:

$$T_{first} : T_{second} = (\mu \cdot RTT - W_{min}) : (W_{max} - \mu \cdot RTT) = (1-r)/2 : r$$

The above calcuation makes the approximation that the TCP flow increases its congestion window size linearly with time during the second stage, i.e. the amount of time lapsed till TCP increases its congestion window by one packet remains constant. This is not true in reality: when TCP congestion window size becomes bigger than $\mu \cdot RTT$, the amount of time it takes to receive a full congestion window of acknowledgements (thus increasing the congestion window by one) also increases since queue starts to build up at the bottleneck router. We thank Dejan for pointing this out.

Given the above approximation, we can calculate the overall average link utilization $U(r)$ as:

$$U(r) = U_{first} \cdot \frac{T_{first}}{T_{first} + T_{second}} + U_{second} \cdot \frac{T_{second}}{T_{first} + T_{second}} = \frac{-r^2 + 6r + 3}{4(4+1)}$$

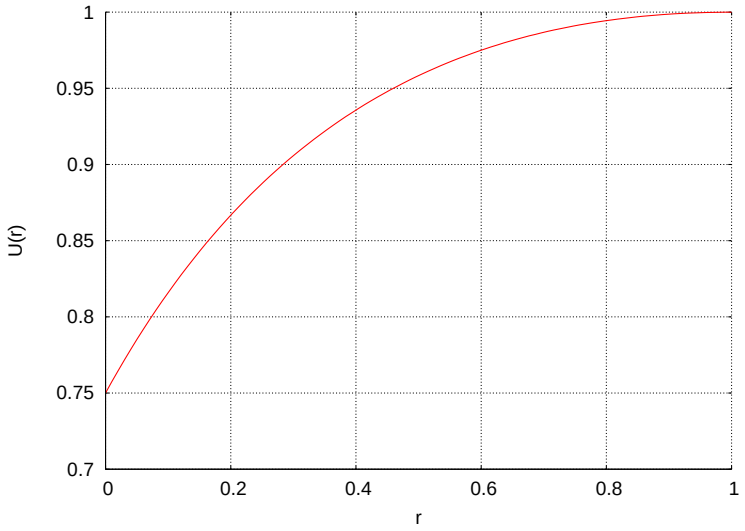

Figure 2: Plot of U(r)

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?

   The worst case happens when the congestion windows of all the TCP connections increase and halve synchronously. When the sum of all congestion window sizes exceeds $\mu \cdot RTT + B$, all the connections lose packets and halve their windows. Then they all grow linearly to the aggregate maximum ($\mu \cdot RTT + B$) again. This is very much like the single TCP connection case. Therefore the relationship between link utilization and buffer size remains the same as for the single TCP flow case.

   The second (best) case happens when TCP flows' congestion window adjustments are completely out of sync. Thus, the aggregate TCP congestion window size could vary according to a normal distribution. In other words, when some of the flows remain in AI phase while others suffer from MD, the aggregate TCP congestion window size varies across a much narrorwer range than linearly from one-half to the maximum size. In such a scenario, a much smaller buffer size is enough to keep the link utilization close to 100%.

   It is still being actively debated as for which of the two cases is closer to the truth in real life. Many people believe that the second case is more likely to happen at backbone routers with tens of thousands of active TCP connections while the first case happens at edges routers with much fewer active TCP flows.

Summary

A Statistics of the Students' Grades (Full 30)

Grades	Student Num
>26	4
24∼26	4
21∼23	4
18∼20	3
<18	2