Thanh Nguyen

Abdullah Yasin Nur

CSCI 4460

7 October 2022

Network Measurement and Analysis

In this lab, I analyzed a dataset that includes AS relationships between c2p and p2p relationships on the internet. I used Java code to read a text file and filtered out the dataset to search for 8 answers. The Internet is frequently referred to as a network of networks since it is a global telecommunications system connecting millions of smaller networks [1]. The internet is used to help people connect at a faster rate. It enables communication between computer users over distance and between computer platforms [1]. Anybody with a computer with the internet can connect to others on the internet. The Internet serves as a pipeline when sending electronic messages from one network to another [1]. A user can access the Internet by using an Internet service provider's (ISP) server [1]. Some examples of ISPs are Cox, AT&T, Spectrum, etc., which provide internet services to customers.

A group of routers known as an Autonomous System (AS) shares administrative authority over their prefixes and routing policies [2]. The AS controls the flow of Internet traffic by using route policies. The AS serves as a connected group of one or many IP prefixes (blocks of IP addresses) that have been issued to that business and offers a single routing policy to systems outside the AS [2].

Business relationships between ASes can be categorized as customer-to-provider (c2p) and peer-to-peer (p2p) [3]. In c2p arrangements, the provider is paid by the customer for the traffic transferred between the two ASes [3] When two ASes are peer-to-peer, they often do so without either ASes having to pay the other for access to their clients [3]. In the dataset, I had that contained p2p and p2c relationships. The data set was formatted as ASNumber1|ASNumber2|Relationship|Source. If the relationship was equal to -1 (c2p), then ASNumber1 was a provider and ASNumber2 was a customer. If the relationship was equal to 0 (p2p), then both ASNumber1 and ASNumber2 were peers. I didn’t use Source in my test runs, but Source represents Border Gateway Protocol (BGP) and Multilayer Perception (MLP). BGP can be viewed as the Internet’s Postal Service, [5] and MLP is a deep learning method that generates outputs from a set of inputs [6].

According to the conventional AS tier classification, tier-1 ASes are described as provider-free ASes that peer with every other provider-free AS to provide reachability to all destinations without using IP transit [4]. Peering is the practice of exchanging Internet traffic with another Tier 1 provider on a casual basis [5]. A Tier 1 ISP uses private settlement-free peering infrastructure to exchange Internet traffic [5]. Because they own their network infrastructure, Tier 1 ISPs can provide the best network throughput through the Internet backbone [5].

A Tier 1 ISP is an independent network that does not share peering information with other networks [2]. This means it does not have to pay the costs associated with using any other network for transit [2]. An Internet service provider (ISP) in the Tier 2 ISP category uses peering with other Tier 2 ISPs in addition to paid transit through Tier 1 ISPs to distribute Internet traffic to end consumers through Tier 3 ISPs [2]. They are typically farther from the Internet's backbone and frequently have slower connection speeds [2]. An ISP that only buys Internet transit is known as a Tier 3 provider [2]. They give end users local access to the Internet through DSL, cable, wireless access networks, or fiber [2]. A Tier 3 provider's main activity is providing end users with Internet access through their networks [2].

This report aims to learn about Degree Centrality's details in graph theory for network analysis. I used real-life data sets of the internet at the autonomous system level. We calculate the Degree of Centrality to see if it is important or not. Like in social media, we have Instagram. Let’s say a user has 10k followers and another has 1k followers. That means in terms of degree size, the user with 10k followers is more important than the user with 1k followers. We will discuss why degree size is not always the best thing to compare.

Another example would be the quality of followers compared to the quality of followers, so let’s say I have ten thousand followers, and you have only ten followers. Your ten followers can be President Biden, Elon Musk, Bill Gates, etc., but my 10k followers are students. So, this is a different metric and if I use degrees of centrality to compare the importance, then this would fail. An autonomous system has a similar idea as well. Therefore, we don’t use degrees of centrality to compare the importance, but it still provides us with extremely important information. We will see that larger autonomous systems will have larger connections and smaller autonomous systems will have smaller connections (like stub-AS; they will have one or two providers and that will be it). (Tier 1 will have 5 thousand customers and 1 thousand p2p, and so on. So, the degree will be high. This type of information.) Finally, the most important part

For the experimental part, there were 8 steps were made. The first step calculated how many unique ASNs are in the dataset. The second step calculated how many unique ASN links are in the data set. The third step calculated the numbers of unique p2p and p2c relations are in the data set. The fourth step calculated the top 10 largest ASN in terms of connection size (Node degree distribution). The fifth step calculated the top 10 largest Provider ASN in terms of connection size (Node degree distribution). The sixth step calculated the top 10 largest peer-to-peer ASN in terms of connection size (Node degree distribution). The seventh step calculated the top 10 largest tier-1 ASN in terms of connection size (Node degree distribution). The eighth step calculated how many stub-AS are in the dataset. To start most of my Java code, the code read in a txt file and parsed each line in the file to split the data needed into its category/column. Through steps 4 to 7, I added codes provided by Dr. Abdullah Yasin Nur that sorts a HashMap and another that prints the top 10 largest degrees.

The first step calculated how many unique ASNs are in the dataset. I only needed to take in ASNumber1 and ASNumber2. I created an array list to gather all the ASNumber1 and ASNumber2 then I took the data from the array list and put it into a HashSet. The HashSet was used to filter out the duplicates. Based on my code, I got an output of 74579 unique ASNs.

In step 2, I read the txt file with the Java code and calculated how many unique links were in the data set. Since the code had to take in the unique links in the dataset, I used a Set (instead of an array list) to create a list of the links in my dataset and put that in a HashSet. When storing the links from the dataset, it stores the link as ASNumber1 connected to ASNumber2 in that order. The code also had to store the reverse order, which in this case would be ASNumber2 connected to ASNumber1. Based on my code, we take the total number of links stored in the list and divide it by 2. The output came out to be 463141 unique links.

In step 3, I read the txt file with the Java code and calculated how many unique p2p and p2c relations are in the data set. In the code, two lists were created to split the p2p and p2c relations. If the data being read had a p2p relationship it was put in a list and into HashSet. If the data had a p2c relationship, it was put in another list and into HashSet. Here I also stored the reverse order of each link in the list and divided the total of each list by 2 at the end. Based on the code, I got the output of 315049 unique p2p relations and 148092 unique p2c relations. This shows that 68% of unique links were p2p and the other 32% of unique links were p2c.

In step 4, the code read the txt file with the Java code and calculated the top-10 largest ASN in terms of connection size (Node degree distribution). The code to calculate step 4 uses a HashMap to store the ASN and the number of connections then sorts the HashMap by value and prints the top 10 largest ASN. The output here gave the ranking of the ASN# degree and the amount that was in the dataset.

1. ASN6939 degree is 9766
2. ASN174 degree is 6633
3. ASN3356 degree is 6423
4. ASN14840 degree is 4600
5. ASN24482 degree is 4550
6. ASN51185 degree is 4517
7. ASN35280 degree is 4314
8. ASN1828 degree is 4197
9. ASN61568 degree is 4001
10. ASN58511 degree is 3330

Output 1: These were the top 10 largest ASN in terms of connection size for step 4.

In step 5, the code took the same approach as step 4 but we only took p2c relations. The output here gave us the top 10 largest Provider ASN in terms of connection size (Node degree distribution).

1. ASN174 degree is 6542
2. ASN3356 degree is 6353
3. ASN7018 degree is 2419
4. ASN1299 degree is 2279
5. ASN6939 degree is 2065
6. ASN6461 degree is 2038
7. ASN3257 degree is 1940
8. ASN2914 degree is 1533
9. ASN701 degree is 1374
10. ASN209 degree is 1189

Output 2: These were the top 10 largest Provider ASN in terms of connection size for step 5.

In step 6, the code took the same approach as step 5 but only took p2p relations. The output here gave us the top 10 largest peer-to-peer ASN in terms of connection size (Node degree distribution).

1. ASN6939 degree is 7697
2. ASN51185 degree is 4512
3. ASN24482 degree is 4509
4. ASN14840 degree is 4330
5. ASN35280 degree is 4252
6. ASN1828 degree is 4126
7. ASN61568 degree is 3847
8. ASN58511 degree is 3277
9. ASN199524 degree is 3185
10. ASN37497 degree is 3160

Output 3: These were the top 10 largest peer-to-peer ASN in terms of connection size (Node degree distribution) for step 6.

In step 7, I had to find the top-10 largest tier-1 ASN in terms of connection size. // Tier-1 ASN: They do not have any providers, they have peers and customers. I used the same approach as step 6, which took in p2p relations. I added another code to also add p2c relations if ASNumber2 (a customer) wasn't on the list. I added another code that checked to see if p2c relations had any ASNumber1s (providers) that were equal to ASNumber2s (customers) and removed the ASNumber1 from the list. Then printed out the top 10 largest tier-1 ASN in terms of connection size.

1. ASN6939 degree is 9008
2. ASN174 degree is 6463
3. ASN3356 degree is 4967
4. ASN24482 degree is 4524
5. ASN51185 degree is 4512
6. ASN14840 degree is 4472
7. ASN35280 degree is 4280
8. ASN1828 degree is 4175
9. ASN61568 degree is 3905
10. ASN58511 degree is 3293

Output 4: **These were the top 10 largest tier-1 ASN in terms of connection size (Node degree distribution) for step 7.**

In step 8, I had to calculate how many stubs-AS were in the dataset. Stub-AS: they may have providers and peers, but they do not have customers. This is basically the opposite of step 7 which filtered out the providers, but step 8 filtered out the customers instead of the providers. The output for step 8 showed 28709 stub-AS were in the dataset. I believe I had human errors in my code because this indicates that 28709 stub-AS/74579 unique-ASNs = ~38% of the dataset was stub-AS. But ~90% of the dataset should've been stub-AS.

In conclusion, I spent most of my time trying to dissect a dataset and found many methods to read in a text file. I initially tried to do everything into one code but found that to be tedious. In the future, dividing the dataset up into more lists would be helpful to filter out the dataset more than having everything into one list. My algorithms for steps 7 and 8 could use improvement.

[1] *The Internet: Internet Basics*, https://fcit.usf.edu/internet/chap1/chap1.htm.

[2] Krzyzanowski, Paul. *Understanding Autonomous Systems: Routing and Peering*, Paul Krzyzanowski, 21 Mar. 2016, https://people.cs.rutgers.edu/~pxk/352/notes/autonomous_systems.html.

[3] Luckie, Matthew, et al. "AS relationships, customer cones, and validation." Proceedings of the 2013 conference on Internet measurement conference. 2013. https://www.caida.org/catalog/papers/2013_asrank/asrank.pdf

[4] Abdullah Yasin Nur, “Analysis of Autonomous System Level Internet Topology Graphs and Multigraphs”, IEEE International Symposium on Networks, Computers and Communications (ISNCC), 2021. https://ayasinnur.com/wp-content/uploads/ASLandASMLComp.pdf

[5] “Internet Service Provider 3-Tier Model.” *ThousandEyes*, https://www.thousandeyes.com/learning/techtorials/isp-tiers.

[6] https://www.cloudflare.com/learning/security/glossary/what-is-bgp/

[7] https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp