CS 425: Computer Networking

IIT Kanpur

Project 4 : Internet Measurements

0. Due Date

This project is due on **November 14, 2016 at 11:55 pm via moodle**. The late submission policy and penalty discussed in the first class applies.

1. Introduction

In this project, you will analyze publicly-available measurement data to understand important properties of the Internet. For the project submission, submit (via moodle) a single PDF file containing (i) the answers to the questions below and (ii) appendices containing the source code for programs you wrote (in whatever language you prefer) to analyze the data. You will also want to choose software for plotting graphs (e.g., Matlab, gnuplot, Excel, R), and have some reusable code for generating a probability-distribution plot from a list of numbers (using either linear or logarithmic scales). (E.g., the statistics toolbox in Matlab, and the various features in R, are quite useful, but you can also write your own short programs to put data in the right format for plotting distributions in Excel or gnuplot.) You can run your programs and analyze the result on any machine you wish, though you may find it useful to use a Linux machine, a Mac, or Cygwin in Windows so you can use the UNIX commands.

Researchers often summarize a large collection of measurement data using distribution functions. Imagine you have a list of Web pages with different sizes, in terms of number of bytes. The *cumulative* distribution function (CDF) of page sizes would have a y-axis of "the fraction of Web pages that are less than or equal to x bytes", and an x-axis of the number of bytes. The graph would start at y=0, since no Web pages have less than or equal to 0 bytes, and reach y=1 when x reaches the size of the largest page. Sometimes researchers plot the *complementary* cumulative distribution function (CCDF), which is "the fraction of Web pages that are greater than x bytes". The graph would start at y=1, since all Web pages have more than 0 bytes, and gradually decrease toward y=0 upon reaching the x-axis value for the largest page. Researchers sometimes plot one or both axes of the CCDF on a logarithmic scale to see more of the detail in some region of the curve. In the questions below, you will plot CCDF distributions, on either linear or logarithmic scales.

2. Traffic Measurement

Many networks collect Netflow measurements directly from the routers. For more information, read the Wikipedia and Cisco pages on Netflow. In this part of the assignment, you'll analyze a five-minute trace of Netflow records captured from a router in the Internet2 backbone that connects the major research universities in the United States. The flow record file in .gz format is available (**flow.2001_09_29.csv.gz**) along with this project description on moodle. Note that the Netflow data from Internet2 anonymizes the last 11 bits of the source and destination IP addresses, to protect user privacy. The records have been parsed into CSV (comma-separated variable) format, with the names of the fields listed in the first row of the file. Internet2 collects Netflow measurements with ***1/100 packet sampling***, so the data reflects 1% of the traffic at the router.

The important fields in the Netflow data are: *dpkts* and *doctets* (the number of packets and bytes in the flow, respectively), *first* and *last* (the timestamps of the first and last packets in the flow, respectively), *srcaddr* and *dstaddr* (the source and destination IP addresses, respectively), *srcport* and *dstport* (the source and destination transport port numbers, respectively), *prot* (the transport protocol, e.g., TCP, UDP), *src_mask* and *dst_mask* (the length of the longest matching IP prefix for the source and destination IP addresses, respectively), and *src_as* and *dst_as* (the AS that originated the IP prefixes matching the source and destination IP addresses, respectively). For example, looking at the first two lines of the file

```
#:unix_secs,unix_nsecs,sysuptime,exaddr,dpkts,doctets,first,last,engine_type,engine_id,srcaddr,dstaddr,nexthop,input,output,srcport,dstport,prot,tos,tcp_flags,src_mask,dst_mask,src_as,dst_as
1285804501,0,2442636503,127.0.0.1,1,40,2442590868,2442590868,0,0,128.103.176.0,24.8.80.0,64.57.28.75,213,225,80,51979,6,0,17,16,0,1742,0
```

you have a flow with one 40-byte packet that arrived at time 2442590868. The packet was sent by source 128.103.176.0 to destination 24.8.80.0, though the last 11 bits are set to 0 due to the *anonymization* of the data. The source port is 80 (i.e., HTTP) and the destination port is 51979 (i.e., an ephemeral port), suggesting this is traffic from a Web server to a Web client. The protocol is 6 (i.e., TCP). The tcp_flags of 17 suggests that the ACK and FIN bits were set to 1, suggesting this is a FIN-ACK packet; the other packets of the Web transfer were presumably not included in the flow record due to packet sampling. The source and destination masks were 16 and 0, respectively, meaning that the source prefix 128.103.0.0/16 and the destination prefix was either unknown or 0.0.0.0/0. The source AS was 1742 (Harvard, according to "whois -h whois.arin.net 1742"), and for whatever reason the destination AS was not known.

A hint: You may find various UNIX commands like cut, sort, uniq, and grep useful in parsing and analyzing the data. For example, if you are processing the file foo.gz you can do:

gzcat foo.gz | cut -d "," -f6 | sort | uniq -c | sort -nr

to extract the sixth comma-separated field (i.e., number of bytes in the flow), count the number of occurrences of each value, and list the frequency counts from most-popular value to least-popular. Including small awk/perl/ruby/python scripts in the pipeline can be helpful for computing sums, averages, and so on. (While testing your code, you may want to test with smaller inputs by piping the data through "head -40" to see just the first 40 lines of the file. You may also find "tail +2" useful for skipping the first line of the file, which consists of the names of the data fields.) ***Answer the following questions:***

- **Q1.1:** What is the average packet size, across all traffic in the trace? Describe how you computed this number.
- **Q1.2:** Plot the Complementary Cumulative Probability Distribution (CCDF) of flow durations (i.e., the finish time minus the start time) and of flow sizes (i.e., number of bytes, and number of packets). First plot each graph with a *linear* scale on each axis, and then a second time with a *logarithmic* scale on each axis. What are the main features of the graphs? What artifacts of Netflow and of network protocols could be responsible for these features? Why is it useful to plot on a logarithmic scale?
- **Q1.3:** Summarize the traffic by which TCP/UDP port numbers are used. Create two tables, listing the top-ten port numbers by *sender* traffic volume (i.e., by source port number) and by *receiver* traffic volume (i.e., by destination port number), including the percentage of traffic (by bytes) they contribute. Where possible, explain what applications are likely responsible for this traffic. (See the IANA port numbers reference for details.) Explain any significant differences between the results for sender vs. receiver port numbers.
- **Q1.4:** Aggregate the traffic volumes based on the source IP prefix. What fraction of the total traffic comes from the most popular (by number of bytes) 0.1% of source IP prefixes? The most popular 1% of source IP prefixes? The most popular 10% of source IP prefixes? Some flows will have a source mask length of 0. Report the fraction of traffic (by bytes) that has a source mask of 0, and then exclude this traffic from the rest of the analysis. That is, report the top 0.1%, 1%, and 10% of source prefixes that have positive mask lengths.
- **Q1.5:** Princeton has the *128.112.0.0/16* address block. What fraction of the traffic (by bytes and by packets) in the trace is sent by Princeton? To Princeton?

3.BGP Measurement

BGP routing changes disrupt the delivery of data traffic and consume bandwidth and CPU resources on the routers. In this part of the assignment, you will analyze BGP update messages logged by RouteViews to analyze BGP (in)stability and convergence behavior. RouteViews has BGP sessions with a variety of different ISPs, and logs the update messages sent on each of these sessions. For this assignment, you will examine BGP sessions from Sydney, Australia available from RouteViews on January 3, 2014, February 3, 2014, and March 3, 2014 between 12pm and 2pm. The update files log BGP updates for each 15-minute interval, and the RIB files have the periodic routing-table (Routing Information Base) dumps. Download the update files

from moodle (provided with this assignment:**updates-asst4.tar.gz**) and the RIB files from moodle (provided with this assignment: **rib-asst4.tar.gz**). These files are in the MRT format specified in RFC6396.

Here is a sample line from an RIB table in TABLE_DUMP_V2 MRT format:
`TABLE_DUMP2|1388750400|B|202.167.228.107|7631|0.0.0.0/0|7631 4826|IGP|202.167.228.107|0|0|7631:50|NAG||`
The source IP is in the fourth column, the source AS in the fifth, the prefix in the sixth, and in the seventh column is the AS path with the origin AS as last.
The update files are in BGP4MP MRT format. Here is a sample line from an update file:
`BGP4MP|1391431556|A|202.167.228.37|38809|66.240.183.0/24|38809 2914 174 23136|IGP|202.167.228.37|0|0|2914:420 2914:1005 2914:2000 2914:3000 65504:174|NAG||`
The prefix affected by update is in the sixth column.
Be aware that the number of prefixes and (especially) the number of BGP update messages is fairly large; so, you will need to take care that your analysis programs make efficient use of memory. ***It is highly recommended that you do not try to complete this assignment on the VM since you will get limited memory available warnings.***

BGP is an incremental protocol, sending an update message only when the route for a destination prefix changes. So, most analysis of BGP updates must start with a snapshot of the RIB, to know the initial route for each destination prefix. Use the RIB snapshot to identify the initial set of destination prefixes, and then analyze the update messages to count the number of update messages for each prefix on a single BGP session (i.e., from one BGP speaker to the RouteViews server).

- **Q2.1:** How many BGP updates per minute does the session handle, on average? Count each announcement or withdrawal for a single prefix as a BGP update, even if multiple prefixes appear in a single update message or a single prefix has multiple announcement or withdrawal messages in a short period of time. Describe how you computed this result.
- **Q2.2:** What fraction of IP prefixes experience *no* update messages? (Count each prefix equally, independently of what fraction of address space they cover or whether one prefix is contained inside another.)
- **Q2.3:** What prefix (or prefixes) experiences the most updates, and how frequent are they?
- **Q2.4:** What fraction of all update messages come from the most unstable 0.1% of prefixes? The most unstable 1% of prefixes? The most unstable 10% of prefixes? By "unstable" we mean the prefixes that have the most route changes.
- **Q2.5:** Briefly summarize your results and what you learned about BGP stability from them.

4. Frequently Asked Questions

What does repeating analysis for several different BGP sessions imply?

You should repeat your analysis for all parts of question 2.1-2.4 across each sessions (each 12-2pm time period). Reporting the results for each session (rather than averaging) is fine. You should get a sense of how similar or different various trends are across different sessions.

When determining prefixes in Question 2, should we consider updates for IP addresses 175.29.0.0/16 and 175.29.135.0/24 to be updates for the same IP prefix, 175.29.0.0?

As said above, "Count each prefix equally, independently of [...] whether one prefix is contained inside another." These should be counted separately.

Can I ignore IPv6 traffic?

Yes.

For Q2.2, we are asked to find the fraction of prefix which is not updated. But the messages in the file are all about the updated prefix. How can we know the prefixes which experience no update message?

You have to consider the RIB data (the full contents of the routing table) as well as the update data. Some of the prefixes in the table are never updated, so you will only know about them if you look at the table dumps.

5. Submission

You should submit your answers to the questions as a single pdf document with appendix with clean formatted listing of code for data analysis.