COS 561 Assignment #3: Internet Measurement

In this assignment, you will analyze publicly-available measurement data to understand important properties of the Internet. For the assignment, submit 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), and have some reusable approach for generating a probability-distribution plot from a list of numbers, since several of the questions involve plotting probability distributions.

1 Traffic Measurement

Many networks collect Netflow measurements directly from the routers. For more information about NetFlow, see

`http://en.wikipedia.org/wiki/Netflow`

and

`http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html`

In this part of the assignment, you'll analyze a five-minute trace of Netflow records captured from a router in the Internet2 backbone—the network that connects the major research universities in the United States. Download the flow records from

`http://www.cs.princeton.edu/courses/archive/fall10/cos561/assignments/flow.2001-09-29.csv.gz`

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.

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. 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 probability distribution 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). What are the main features of the graphs? What artifacts of Netflow and of network protocols could be responsible for these features?
- 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 `http://www.iana.org/assignments/port-numbers` for details.)
- Q1.4: Plot the distribution of the fraction of bytes and the fraction of packets by source IP prefix. Choose whether to plot a Cumulative Distribution Function or Complementary CDF, and either log or linear axes, to highlight the main interesting features of the distribution. Describe how you computed the distribution, and the key features of the graphs.
- Q1.5: Princeton owns 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?

2 BGP Measurements

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 (`http://www.routeviews.org/`) 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. To access the data, go to

`ftp://archive.routeviews.org/`

and pick one of the directories starting with "route-views" (e.g., "route-views.eqix"), and find update data from a particular month, e.g., the directory

`ftp://archive.routeviews.org/route-views.eqix/bgpdata/2010.08/UPDATES/`

has files logging the BGP updates for each 15-minute interval, and the directory

`ftp://archive.routeviews.org/route-views.eqix/bgpdata/2010.08/RIBS/`

has the periodic routing-table (Routing Information Base) dumps. These files are in a compressed, binary format (e.g., gzip or bzip2). You will need tools to "uncompress" and parse the data, as discussed in

`http://www.routeviews.org/tools.html`

The bgpdump tool is probably the best choice for parsing the update messages (running "`bgpdump -m`" is especially useful to produce easily-parsable output). The latest version of the bgpdump tool is available from:

`http://www.ris.ripe.net/source/libbgpdump-latest.tgz`

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.

2.1 BGP Stability Across Prefixes

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 next several hours of 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), repeating your analysis for several different BGP sessions.

- Q2.1: 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.)
- Q2.2: What prefix experiences the most updates, and how frequent are they?
- Q2.3: Plot the probability distribution of the number (or fraction) of updates by prefix. Choose whether to plot a Cumulative Distribution Function or Complementary CDF, and either log or linear axes, to highlight the main interesting features of the distribution.
- Q2.4: Briefly summarize your results and what you learned about BGP stability from them.

2.2 BGP Convergence

BGP routing changes involve a path-exploration process, where a router may explore several alternate routes for a prefix before settling on a final decision. Then, a future network event, like a failure or recovery, may lead to another flurry of update messages that are logically distinct from the first set of update messages. As such, a common step in analyzing BGP convergence is to group BGP update messages into "events"—update messages for the same prefix that are sent close together in time. This requires a way to identify a threshold T for update-message interarrival times—consecutive updates sent *less* than T seconds part are part of the *same* event, whereas updates sent T or more seconds apart belong to different events.

- Q2.5: Plot the distribution of update-message inter-arrival times, combining data across all <prefix, session> pairs. Choose whether to plot a Cumulative Distribution Function or Complementary CDF, and either log or linear axes, to highlight the main interesting features of the distribution.
- Q2.6: What is a reasonable value for the threshold T?

- Q2.7: Using the threshold T, group BGP updates (for each <prefix, session> pair) into events, where each event has a start time (the time of the first update message), finish time (the timestamp of the last update message), and the total number of update messages. Plot the probability distributions of the (i) the number of updates per event and (ii) the time duration of an event.
- Q2.8: Based on the probability distributions, summarize your understanding of the main classes of convergence events.