COSC 465: Computer Networking
Lab 5: BGP trace analysis
Due: March 28, 2012, before lab

1 Overview

The BGP (border gateway protocol) is responsible for exchanging inter-domain routing information in order for Internet routers to deliver packets from a source to a destination. Updates to BGP routes can disrupt the delivery of data traffic and consume bandwidth and CPU resources on routers. In this lab, you will analyze BGP update messages logged by RouteViews (`http://www.routeviews.org`) to analyze aspects of BGP behavior. The goal for this lab is for you to learn more about BGP behavior, and what information about global Internet routes is exposed through BGP updates.

RouteViews has BGP *sessions* with a variety of different ISPs, and logs the update messages sent over each of these sessions. (A session is basically just a connection to a router over which eBGP update messages are delivered.) What RouteViews logs is the collection of all the BGP updates from many routers, resulting in a (potentially) more complete view of global Internet routing activity. Each router generating a BGP update message is acting on locally received and processed data; what we see at RouteViews is the aggregate of a variety of these local views of the Internet.

I have already collected some data from `ftp://archive.routeviews.org` and used a tool called —bgpdump— to prepare some files for you to analyze. The data we will use includes a "full" snapshot of the routing table (Routing Information Base, or RIB) taken at midnight on March 1, 2012. The data also includes a 10 hour trace of BGP updates, starting at midnight on March 1, 2012. You should be aware that these files are gzip'ed text files, and contain quite a bit of data. The RIB file has 4,382,208 lines, and the update file contains 1,871,123 lines (the structure of these files is described below)[1]. (Note that if you want to decompress these files, you can just type `gzip -d filename.gz`.)

The deliverable for this lab is a short writeup that answers the questions below, as well as any code you wrote to process the BGP data files. You are welcome to write your analysis code in *any* language (C, C++, Python, shell scripts, or even (yuck) Java).

You are welcome to work in groups of up to 3 persons for this lab; only one submission needs to be made for the group. While there are no template files for this lab, there are some code snippets below and linked on Moodle to help with producing various types of plots on Moodle.

2 Data set details

There are two data files for this lab: a full RIB (routing information base) dump and a BGP update trace. The RIB dump provides the state of routing tables from the perspective of several ISPs. The BGP update trace provides a series of route announcements and withdrawals, which likely cause changes to the routing table. Note that there are some lines in the RIB and update file that indicate IPv6 addresses. You can ignore any IPv6 prefixes in your analysis.

The RIB file contains a series of lines with the following structure:

Output version | Timestamp | A | Peer IP address | Peer AS | Prefix | AS Path | Origin

Some example lines are shown below:

```
TABLE_DUMP_V2|02/26/12 08:40:08|A|206.223.115.26|16559|1.0.16.0/22|16559 6461 2914 2519|IGP
TABLE_DUMP_V2|02/26/12 07:20:19|A|206.223.115.12|2914|1.0.16.0/22|2914 2519|IGP
TABLE_DUMP_V2|02/14/12 10:01:44|A|206.223.115.24|11666|1.0.16.0/22|11666 3257 2914 2519|IGP
TABLE_DUMP_V2|02/18/12 06:57:54|A|206.223.115.47|19151|1.0.16.0/22|19151 2516 2519|IGP
TABLE_DUMP_V2|01/26/12 19:50:36|A|206.223.115.10|4589|1.0.16.0/22|4589 2516 2519|IGP
```

[1] If you want to use a different set of files (you're welcome to do so), you can obtain `bgpdump` from the following link: `http://www.ris.ripe.net/source/libbgpdump-latest.tgz`, and you can get additional/different RouteViews data from `ftp://archive.routeviews.org/route-views.eqix/bgpdata/2012.03/RIBS/` and `ftp://archive.routeviews.org/route-views.eqix/bgpdata/2012.03/UPDATES/`. You'll need to use `bgpdump` to transform the binary RIB and UPDATE files into text files for your analysis.

Each line shows the AS path between one AS and a particular prefix. For example, the first line shows that AS16559 can reach the prefix 1.0.16.0/22 through the AS path 16559, 6461, 2914, and 2519.

Each BGP update file contains a series of lines with the following structure:

BGP protocol | Timestamp | Withdraw or Announce | Peer IP address | Peer AS | Prefix

A "withdraw" message means that the advertising AS is indicating that the given prefix is no longer reachable through it. An "announce" message indicates that a prefix is reachable through the advertising AS along the specified AS path.

For a withdraw message, the structure is exactly as above. For an announce message, two additional items on each line are included, giving a structure exactly like the lines in the RIB file:

BGP protocol | Timestamp | Withdraw or Announce | Peer IP address | Peer AS | Prefix | AS Path | Origin

Some example lines:

```
BGP4MP|03/01/12 00:00:41|A|206.223.115.47|19151|186.193.16.0/20|19151 3549 27724
 53078|IGP
BGP4MP|03/01/12 00:00:41|A|206.223.115.120|41095|122.161.0.0/16|41095 9498 24560
 24560 24560|INCOMPLETE
BGP4MP|03/01/12 00:00:41|A|2001:504:0:2::4436:2|4436|2001:7fb:fe0e::/48|4436 286
 12859 12654|IGP
BGP4MP|03/01/12 00:00:41|A|206.223.115.25|6079|205.104.240.0/20|6079 3356 209 72
1 27064 3475|IGP
BGP4MP|03/01/12 00:00:41|A|2001:504::2:0:1:9151:1|19151|2001:7fb:fe0e::/48|19151
 12654|IGP
BGP4MP|03/01/12 00:00:41|A|2001:504:0:2::2914:1|2914|2001:7fb:fe0e::/48|2914 128
59 12654|IGP
BGP4MP|03/01/12 00:00:41|W|206.223.115.181|8781|176.15.127.0/24
BGP4MP|03/01/12 00:00:41|W|206.223.115.181|8781|199.58.99.0/24
BGP4MP|03/01/12 00:00:41|W|206.223.115.181|8781|212.33.118.0/23
BGP4MP|03/01/12 00:00:41|W|206.223.115.181|8781|46.252.32.0/24
BGP4MP|03/01/12 00:00:41|W|206.223.115.181|8781|46.252.33.0/24
BGP4MP|03/01/12 00:00:41|W|206.223.115.181|8781|46.252.37.0/24
```

(Note that there are three lines above that refer to IPv6 addresses; any IPv6 advertisements and updates can be ignored in your code.)

3 Problems

3.1 Colgate and BGP

First, some initial exercises to learn a bit about Colgate's connection to the Internet.

(a) The RIB file contains evidence of a number of ASes announcing reachability to Colgate's IPv4 prefix, 149.43.0.0/16. (You should find 10.) What is Colgate's AS number? (You should be able to infer Colgate's AS number from the AS paths you find.)

(b) Each of the AS paths showing reachability to Colgate's IPv4 prefix shows something of a peculiarity in that Colgate's AS number is repeated exactly 4 times in each AS path. Formulate an explanation for why you think this is so. (You may wish to search the interwebs for "AS path prepending" or "AS path padding" to learn more about this behavior.)

(c) Colgate buys Internet service from Time Warner. What is Time Warner's AS number? (Note that while Colgate is technically a multi-homed AS—we have "commodity" Internet service through Time Warner, and Internet 2 service, also through Time Warner—the AS paths evident in the RIB file only show our commodity (commercial) link.)

3.2 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 in order to know the initial route for each destination prefix. Use the RIB snapshot and analyze the routing table from a *single* BGP session. You should build a data structure to hold the initial set of prefixes (along with any other information you think is relevant), then analyze the update file, which includes 10 hours of BGP update messages. (For identifying the RIB and update messages for a single session, you can simply process all

lines of the files that have the same peer AS number.) Your overall goal for this part of the lab is to count the number of update messages for each prefix.

(a) 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.)

(b) What prefix experiences the most updates, and how frequent are they (in updates per minute)?

(c) Create a plot of the number of updates per prefix. At minimum, you should create a histogram of the number of updates per prefix. You may wish to plot the histogram on log scale to highlight features of the distribution better, or to plot a cumulative distribution function. (Some code for Python's matplotlib library is shown below to help with this.)

(d) Analyze the plot; briefly summarize your results and what you can infer about BGP stability from the plot and your results.

3.3 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.

(a) Create a program to plot the distribution of update-message inter-arrival times for a given BGP session. You can choose how to plot these times. You can use a simple histogram at minimum, and you can consider whether to use a log axis, or a cumulative distribution plot. You can experiment with the plot to highlight interesting features of the distribution.

(b) What is a reasonable value for the threshold T? Briefly explain.

(c) What do you think this threshold suggests about the nature of BGP events and BGP route convergence time?

4 Plotting example with matplotlib

For plotting, matplotlib is a great library that provides matlab-like plotting capabilities. It should be available on all Linux lab machines, and can be easily installed on Ubuntu Linux using apt-get[2]. The main page for matplotlib is available at: `http://matplotlib.sourceforge.net/`. Even if you don't write your analysis code in Python, you could simply create data files that could be read in by the plotting code to make the figure.

A short snippet for creating a histogram plot is shown below. The `makehisto` function takes a list of numbers from which to create a histogram. Specific documentation on the matplotlib `hist` function can be found at: `http://matplotlib.sourceforge.net/api/pyplot_api.html#matplotlib.pyplot.hist`.

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import random

def plothisto(hlist):
    fig = plt.figure()
    ax = fig.add_subplot('111')
    ax.hist(hlist, histtype='bar')
    plt.xlabel('Number of BGP update messages')
    plt.ylabel('Count of BGP update messages')
    plt.title('Here\'s my fancy histogram!')
    plt.savefig('myhistogram.pdf', bbox_inches='tight', pad_inches=0.1)
    plt.close()

# make some random data to plot
data = [ random.randint(1,100) for _ in xrange(100) ]
plothisto(data)
```

[2] `apt-get install python-matplotlib`.