Tools and Techniques for the Analysis of Large Scale BGP Datasets

Manish Karir, Larry Blunk (Merit)
Dion Blazakis, John Baras (UMd)
The Problem

• Large amounts of data are now, or soon will be available:
  – RouteViews, RIPE Archives, PREDICT, etc
• The problem is no longer access to raw data but how to extract useful information from the raw data
• Need tools that can:
  – Scale to large input datasets
  – Provide useful data summarizations
  – Are easy to use
  – Provide useful information
• BGP::Inspect
  – Goal is to attempt to make it easier to use raw data from archives such as RouteViews, by pre-processing, reformatting and indexing the data
Outline

• BGP::Inspect and BGPdb
  – Architecture, Techniques, Algorithms

• BGP::Inspect Interface
  – Basic queries, Global Summarizations
  – Detailed specific queries, AS/Prefix

• Case Study 1 – The AS9121 Incident

• Case Study 2 – Prefix Hijacking Example

• Conclusions, Future Work and Discussion
BGP::Inspect

• Analyzing MRT Data:
  – Large volumes of data ~RV-66G compressed
  – Extracting useful information requires writing custom parsers even for basic information
  – Lots and lots of redundancy

• Approach:
  – Preprocess RouteViews data
  – Remove redundancy as much as possible
  – Use data compression to the extent possible
  – Build efficient indices to help queries
  – Pre-compute and store commonly used statistics at data load time not at query time
  – Build easy to use interface
BGPdb

- BGPdb is the core of the BGP::Inspect system
- BGPdb represents the pre-processed database, which is queried by the BGP::Inspect interface
- Provides some useful techniques that maybe applied to processing other large datasets not just BGP datasets
BGPdb – Techniques and Algorithms

• Removing redundancy from BGP datasets
  – ASPATH, COMMUNITY, UPDATE Msgs are repeated over and over, only time changes

• Compressed-Chunked Files
  – Compromise between size and usability

• B+ Tree indices
  – Indexing based on time, this enables fast time-range queries

• Caching while processing input datasets
  – Messages are repetitive, so keep cache of previous processing for speedup
BGPdb – System Architecture
BGP::Inspect

BGP::Inspect – Beta v0.2
http://weasel.merit.edu:8080

Dataset: Jan1- March31 2005

• Example queries (per peer, 1,7,30 days):
  • Most active AS’s
  • Most active prefixes
  • Prefixes with most OriginAS changes

• Raw Data Analysis (per peer)
  • Prefix/AS, Time Range
  • Uniques prefixes by AS
  • OriginAS changes for a prefix
  • Time to run query
  • More specific prefixes announced
BGP::Inspect Interface
Global Queries – Most Active ASes
Global Queries: Most OriginAS Changes
Raw Data Analysis – AS Query
Raw Data Analysis – Prefix query

### BGP::Inspect

#### Global Summary Queries:
- **Query Type:** (Select a RouteViews Peer, Query Type, and Time Interval)
  - 12.0.153.41
  - 46.183.128.1-4096
  - 144.238.241.81
- **Query Time Range:** (Select a RouteViews Peer, Query Type, AS, Prefix, and the Query Time Range)
- **Start Date:**
- **End Date:**

#### Query Summary Statistics

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Time Range</td>
<td>Fri Mar 25 00:00:00 2005</td>
</tr>
<tr>
<td>Total Advertisement</td>
<td>35</td>
</tr>
<tr>
<td>Total Update</td>
<td>33</td>
</tr>
<tr>
<td>Avg Path Length</td>
<td>2.94</td>
</tr>
<tr>
<td>Average AS Path</td>
<td>20</td>
</tr>
<tr>
<td>Time to run query</td>
<td>0.06</td>
</tr>
</tbody>
</table>

#### Prefix Announcements:

<table>
<thead>
<tr>
<th>Time</th>
<th>Type</th>
<th>AS Paths</th>
<th>Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fri Mar 25</td>
<td>03:37:31 2395</td>
<td>1239 2914 17675 23918</td>
<td>1239:2512 1239:1000 1239:3012</td>
</tr>
<tr>
<td>Fri Mar 25</td>
<td>03:58:24 2395</td>
<td>1239 2388 17675 23918</td>
<td>1239:2512 1239:1000 1239:3012</td>
</tr>
<tr>
<td>Fri Mar 25</td>
<td>09:02:35 2395</td>
<td>1239 2388 17675 23918</td>
<td>1239:2512 1239:1000 1239:3012</td>
</tr>
<tr>
<td>Fri Mar 25</td>
<td>16:19:29 2395</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Fri Mar 25</td>
<td>22:01:43 2395</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Fri Mar 25</td>
<td>02:12:23 2395</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Sat Mar 26</td>
<td>05:57:52 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Sun Mar 27</td>
<td>09:10:22 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Mon Mar 28</td>
<td>01:28:45 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Mon Mar 28</td>
<td>03:32:00 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Mon Mar 28</td>
<td>05:17:20 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Mon Mar 28</td>
<td>09:03:30 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
<tr>
<td>Tue Mar 29</td>
<td>02:29:40 2005</td>
<td>1239 29327</td>
<td>1239:123 1239:5000 1239:5140</td>
</tr>
</tbody>
</table>
Case Study 1 – AS9121 Incident

- At ~09:19 UTC on Dec 24, 2004, AS9121 began re-originating a large number of globally routed prefixes

- Forensics:
  - What happened?
  - Who did it?
  - Could there have been some early detection?
  - How widespread was it?
Step 1: What…

RouteViews Peer: 12.0.1.63

Most Active ASes, Last 30 Days:

1. 21617 NARA National Archives and Records Administration 537856
2. 27155 HARRIS-61 Harrisonville Telephone Company 268882
3. 7018 ATT WAT WorldNetwork Services 891313
4. 16501 TIDT-I-3 The Titan Corporation 644659
5. 19458 CARGIL-Cargill Incorporated 564255
6. 2736 ADACS-1 AT&T Data Communications Services 55540
7. 12062 DECIMA-34 Decision One 46797
8. 9416 BATEC-UB 186538
9. 16509 AES-C&G. Edwards & Sons, Inc. 24173
10. 271 ONIC DDI Network Information Center 22463

BGP::Inspect

Global Summary Queries:

Raw Data Analysis:
Step 1.5: Hmm...interesting...
Step 2: Was I affected?/Should I care?
Step 3: Where...

Level 3 - No

GLBX - No

AOL - Yes

Sprint - Yes
Step 4: How widespread...
Step 4: How widespread...
Step 5: How long...

<table>
<thead>
<tr>
<th>Time</th>
<th>Unique Prefixes Announced by 9121 as seen by Sprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>07-08</td>
<td>0</td>
</tr>
<tr>
<td>08-09</td>
<td>0</td>
</tr>
<tr>
<td>09-10</td>
<td><strong>4604</strong></td>
</tr>
<tr>
<td>10-11</td>
<td>56</td>
</tr>
<tr>
<td>11-12</td>
<td>804</td>
</tr>
<tr>
<td>12-13</td>
<td>56</td>
</tr>
<tr>
<td>13-14</td>
<td>196</td>
</tr>
<tr>
<td>14-15</td>
<td>159</td>
</tr>
<tr>
<td>15-16</td>
<td>34</td>
</tr>
<tr>
<td>16-17</td>
<td>92</td>
</tr>
<tr>
<td>17-18</td>
<td>54</td>
</tr>
<tr>
<td>18-19</td>
<td>172</td>
</tr>
<tr>
<td>19-20</td>
<td><strong>4496</strong></td>
</tr>
<tr>
<td>20-21</td>
<td>229</td>
</tr>
<tr>
<td>21-22</td>
<td>15</td>
</tr>
<tr>
<td>22-23</td>
<td>0</td>
</tr>
</tbody>
</table>
Case Study 2 – Prefix Hijack Incident

- **Incident:** On Feb 10th, AS2586, announces 207.75.135.0/24, which is part of Merit’s CIDR block 207.72.0.0/14
- **Trouble ticket filed,** bogus announcement withdrawn by AS2586 by Feb 10th, 19:22hrs
- **How do we find out** what happened?
- **Could there have been** automated detection?
- **What was the impact,** how widespread was it?
Step 1 – Finding out what happened...
Step 2 – Who, why…
Step 3 – where…
Conclusions and Future Work

• There is a need to build efficient tools that help extract useful information from large BGP datasets.
• BGP::Inspect is currently available to the network operator and research communities and feedback is appreciated.
• Aside from BGP::Inspect we have presented some basic techniques such as chunked-compressed files, B+ Tree indexing, data redundancy elimination, and caching that can be applied by other data mining tools to help analyze other large datasets as well.
• The goal is not just to provide access to the data, but to try to provide useful data summaries as well, that can help researchers and network operators quickly identify potentially “interesting” events. Top20 lists are a good way to bring potentially interesting things to the attention of people.
• Tools need to be useful before they can be used, and in order to be useful, feedback from potential users is critical.
• BGP data analysis need not be hard/painful/tedious, that’s what tools are for!
• Where do we go from here, so we have basic capabilities what about:
  – Automated anomaly detection, notification, same tool?, different tool?
  – More scalability,? What are the limits?
  – What are more useful queries? What book-keeping do we need to track those?