Meeting with Cisco Systems
San Jose, CA, 5th Dec 2017

BGPStream and OpenBMP

Alberto Dainotti, Alistair King
alberto@caida.org, alistair@caida.org

Center for Applied Internet Data Analysis
University of California, San Diego
AGENDA

20 min + Q/A

- BGPStream
- Collab w/ Cisco OpenBMP group
- V2 & Applications
- Future Work
Game-changing technology to enable BGP analytics

- Open Source Software **APIs** for historical and *live* BGP data analysis
  - Python, C, Command Line tools, …
- Accelerates existing and enables fundamentally new analytic capabilities

- Design goals:
  - Efficiently deal with large amounts of distributed BGP data
  - Offer a time-ordered data stream of data from heterogeneous sources
  - Support near-realtime data processing
  - Target a broad range of applications and users
  - Scalable (e.g., *use Apache Spark to crunch billions of updates*)
  - Easily extensible
  - Simple API
  - Facilitates reproducibility and repeatability

bgpstream.caida.org
• **RIPE 70** and tech report - May 2015
• Version 1 and **IETF 94** Tech Plenary - Nov 2015
• Version 1.1 and **BGP Hackathon** - Feb 2016
• **NANOG 66** - Feb 2016
• **ACM IMC** paper - Nov 2016
• **IRTF ANRP** award - Jan 2017
• **IETF 98** - Mar 2017
• Version 2.0 beta 1 - Today!
PEOPLE USE IT

hackathons, papers, net admins, …

• Various hackathons: NANOG, RIPE, …
• github.com/caida/bgpstream
  • some significant pull requests from 3rd parties

• Selected papers:
  - **Counter-RAPTOR: Safeguarding Tor Against Active Routing Attacks** [SP’17] - Sun et al.
  - **I-Seismograph: Observing, Measuring, and Analyzing Internet Earthquakes** [ToN’17] - Zhang et al.
  - **Sibyl: A Practical Internet Route Oracle.** [NSDI’16] - Cunha et al.
  - **PathCache: A Path Prediction Toolkit.** [SIGCOMM’16] - Singh et al.
State of the Art?

wget http://archive.org/xyz/abc/file.mrt
bgpdump -m file.mrt | my_parser.py
The BGPStream Framework

- Metadata Broker
- Metadata crawler
- BGPStream
- User Libraries
- MRT data (via HTTP)
- User Code
- Python API
- libBGPStream

Public HTTP Data Archives
1. A web service (“BGPStream Broker”)  
   • enables SIMPLE **access** to many heterogeneous BGP sources  
2. **LibBGPStream**:  
   • Acquires the data and provides to upper layers a realtime stream of BGP data  
   • makes it SIMPLE to **process** data from many heterogeneous BGP sources  
3. Command-line tools and APIs in C and Python
libBGPStream talks to the broker and gets the data.

```python
stream.add_filter('record-type', 'ribs')
stream.add_filter('collector', 'route-views.sfmix')
stream.add_interval_filter(1445306400, 1445306402)
```

```bash
$ bgpreader -w 1445306400,1445306402 -c route-views.sfmix -t updates
```

Experiments can be easily reproduced: a script defines the (public) data used.

Center for Applied Internet Data Analysis
University of California San Diego
BGPREADER

command-line tool for ASCII output w/ filters

$ bgpreader -w 1445306400,1445306402 -c route-views.sfmix
R|B|1445306400|routeviews|route-views.sfmix
R|R|1445306400|routeviews|route-views.sfmix|32354|206.197.187.5|1.0.0.0/24|206.197.187.5|32354 15169|15169|||
...
R|R|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|2c0f:ffd8::/32|
2001:504:30:ba01:4061:1|14061 1299 33762|33762|1299:30000|||
R|R|1445306401|routeviews|route-views.sfmix|32354|2001:504:30:ba03:2354:1|2c0f:ffd8::/32|
2001:504:30:ba00:6939:1|33762|33762|6939:37105|37105:33762:33762|||
R|E|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|2402:ef35::/32|
U|A|1445306401|routeviews|route-views.sfmix|32354|2001:504:30:ba03:2354:1|2402:ef35::/32|
2001:504:30:ba03:2354:1|33762|6453 4755 7633:76333|||
U|A|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|2a02:158:200::/39|
...
Example: studying AS path inflation

How many AS paths are longer than the shortest path between two ASes due to routing policies? (directly correlates to the increase in BGP convergence time)
PYBGPSTREAM

Example: synchronizing with active measurements

- We monitor **community-based black-holing**
- Victim of DoS attack announces prefix with special community attribute to request that neighbors drop traffic
- We trigger traceroutes to characterize the black-holing event (using 50-100 probes per event)
- probed 253 victims (90-95% of black-holing events) while black-holing in effect
- **Combined passive control-plane and active data-plane measurements to capture and investigate transient routing policies**

Figure 4: Two metrics showing a pronounced difference in the fraction of probes reaching origin AS. 

1. Fraction of probes reaching origin AS.
2. Traceroutes during RTBH
3. Traceroutes after RTBH

Example: synchronizing with active measurements.
There are a few observations in this experiment. There are a few observations in this experiment. We identify MOAS (Multi Origin AS) prefixes. We then calculated the number of transit ASes (ASes appearing in the middle of an AS path) observed for different collectors. Besides the slow growth in observable MOAS sets over the years, we find a fast rate: the IPv6 graph is growing fast while its edge community diversity as observed by VPs represents a higher concentration of points from different ASes.

To obtain a better view into overall (top blue line) and per-collector (other lines), this graph highlights that to obtain a better view into overall (top blue line) and per-collector (other lines), this graph highlights that to obtain a better view into overall (top blue line) and per-collector (other lines).

CODE AT

There are a few observations in this experiment:

- RIB-outs are numerous and they significantly skew the distribution; only 710 VPs out of 2,296 are within 20 percentage points of the maximum at each time bin.
- We identified MOAS (Multi Origin AS) prefixes.
- Since around 2012, such decay has slowed down considerably, while the fraction of transit ASes is much larger than the maximum number identified as many collectors as are available: the number of MOAS prefixes, it is important to analyze data from at least two collectors (Route Views and RIPE RIS repositories).

This graph highlights that to obtain a better view of MOAS per collector and aggregate, we find it convenient to normalize data in the other experiments.

In this experiment, we also compute, at each level of aggregation (VP, collector, overall), the number of unique prefixes and ASes observed, which we use to form our analyses with data from the 15th day of the month (thus we perennially miss RIB dumps (34 per year on average) that both the Route Views and RIPE RIS repositories do not have a single full-feed peer, thus may not provide enough information for most experiments; we then calculated the number of transit ASes (ASes appearing in the middle of an AS path) observed for IPv4, despite the nearly-linear growth in the number of ASNs (dashed lines) and percentage of those ASNs which has kept growing at a constant pace.

Besides the slow growth in observable MOAS sets over time, this graph highlights that to obtain a better view of MOAS per collector and aggregate, we find it convenient to normalize data in the other experiments.

For IPv4, despite the nearly-linear growth in the number of ASNs (dashed lines) and percentage of those ASNs which has kept growing at a constant pace, the total number of IPv6 ASes has kept growing at a growing faster than transit. However, since around 2012, such decay has slowed down considerably, while the fraction of transit ASes is much larger than the maximum number identified as many collectors as are available: the number of MOAS prefixes, it is important to analyze data from at least two collectors (Route Views and RIPE RIS repositories).

The graph in Figure 5c, shows that the number of unique sets of ASes (#MOASs sets) per collector and aggregate is always significantly larger than the maximum number identified as many collectors as are available: the number of MOAS prefixes, it is important to analyze data from at least two collectors (Route Views and RIPE RIS repositories).

The graph in Figure 5b, shows many problems ([34]) including the detection of BGP hijacking activity ([20]). The graph in Figure 5c, shows that the number of unique sets of ASes (#MOASs sets) per collector and aggregate is always significantly larger than the maximum number identified as many collectors as are available: the number of MOAS prefixes, it is important to analyze data from at least two collectors (Route Views and RIPE RIS repositories).

The graph in Figure 5a, shows that the number of unique sets of ASes (#MOASs sets) per collector and aggregate is always significantly larger than the maximum number identified as many collectors as are available: the number of MOAS prefixes, it is important to analyze data from at least two collectors (Route Views and RIPE RIS repositories).
There are a few observations in this experiment. Results of historical analysis using PyBGPStream and Apache Spark. Transient and permanent ASes are numerous and they significantly skew the distribution. Only 710 VPs out of 2,296 are within 20% of the maximum at each time bin. The graph in Figure 5b shows the number of unique MOAS sets identified in the Adj-RIB-out of VPs; warmer colors represent a higher concentration of points. There is a significant number of points from different VPs. The heatmap of data from 2,296 VPs (warmer colors representing a higher concentration of points). Aggregated data (collector and project) is depicted as grey circles. The percentage of points of the maximum at each time bin is significant, and there is a skewed distribution. Only 710 VPs out of 2,296 are within 20% of the maximum at each time bin.

Figure 5: Transit ASNs

- Transit ASNs % (IPv4)
- Transit ASNs % (IPv6)
- # ASNs (IPv4)
- # ASNs (IPv6)

COLLAB W/ CISCO
COLLAB W/ CISCO

Tasks

• 1: Native OpenBMP Support in BGPStream
  - Rearchitect BGPStream
  - Add support for Kafka Encapsulation
  - Deserialize BMP data into BGPStream

• 2: Distribute BMP data
  - Run a public collector
  - Coordinate w/ RouteViews
  - Tutorial on how to use BGPStream with your BMP router/collector

• 3: Cooperate w/ OpenBMP group
  - Create LibParseBGP
  - New OpenBMP features: NAT/PAT, Router connection rate limiting
  - Bi-dir feedback and testing
  - Dissemination of results and work in progress
TASK 1

Native OpenBMP Support in BGPStream

• Pre-existing conditions: BGPStream v1.1
  - No BMP support
  - entirely MRT-based
  - RouteViews, RIPE RIS, … ➔ ~20 min delay! We want real live streaming
**Task 1.1 - Rearrchitect BGPStream**
- LibBGPStream had a monolithic architecture. We turned it into modular (object-oriented C implementation)

**BEFORE**
- Data Interfaces:
  - Broker; Single file; CSV
- Single format
  - MRT
- Single transport
  - Dump files (http/FS)

**NOW**
- Data Interfaces:
  - Broker; Single file; CSV; **Kafka**
- Formats:
  - MRT; **BMP**; ...
- Transports:
  - Dump files; **Kafka**
**TASK 1**

Native OpenBMP Support in BGPStream

- Task 1.2 - Add support for Kafka encapsulation
  - Based on Librdkafka (https://github.com/edenhill/librdkafka)

---

**BEFORE**

- Data Interfaces:
  - Broker; Single file; CSV
- Single format
  - MRT
- Single transport
  - Dump files (http/FS)

**NOW**

- Data Interfaces:
  - Broker; Single file; CSV; **Kafka**
- Formats:
  - MRT; **BMP**; ...
- Transports:
  - Dump files; **Kafka**
TASK 1

Native OpenBMP Support in BGPStream

• Task 1.3 - Deserialize BMP data into BGPStream
  - Deserialization
    - Wrote C code inspired by OpenBMP’s C++
    - Created a standalone library libParseBGP (Task 3.1) https://github.com/CAIDA/libparsebgp
      - Good engineering practice (cleanness, modularity, …)
      - Provide the community with a BMP/MRT/BGP parsing library

  - New format for encapsulation of BMP data
    - OpenBMP currently uses an ascii encapsulation
    - Needed timestamps, info about router and collector, …
    - binary/ascii
• Parses BGP, BMP, MRT from a buffer into a C structure
• Parsed data is “close to the RFC”.
  - library doesn’t assume anything about how you will use it -- e.g., addresses are left as 4, 16 byte network-byte ordered values
• Data is parsed into a reusable structure
  - dynamic memory inside the structure is reused between parses -- avoids free/mallocs and drastically improves performance
• Many path attributes supported
  - https://github.com/CAIDA/libparsebgp/blob/master/lib/bgp/parsebgp_bgp_update.h#L225
• Support for selectively parsing features
  - clients who only need specific features, e.g., AS Path, Community attributes, don’t need to parse the entire message
 TASK 2

Distribute BMP data

• **Task 2.1 - Run a public collector**
  • Nobody had experience in operating a public BMP collector
  • CAIDA’s Public BGPStream OpenBMP Collector
    • [bmp.bgpstream.caida.org:9092](bmp.bgpstream.caida.org:9092)
  • Already providing feeds
    - 1 Cisco router, 1 Cisco peer (AS11017 - CSN)

• **Task 2.2 - Coordinate with RouteViews**
  • Operated in collaboration with RouteViews
    - 1 RV router, 3 RV peers (operational routers from Level3, HE, AT&T)
    - Work in progress, slowly will add more and more
    - Lesson learned: non negligible load on router due to BMP
• Task 2.3 - Tutorial to use BGPStream w/ your router/collector
  • Leverages the OpenBMP docker container
    • [https://bgpstream.caida.org/v2-beta](https://bgpstream.caida.org/v2-beta)
    • live demo now

• Shows how to analyze your private router’s BMP feed from pyBGPStream
TASK 3
Cooperate w/ OpenBMP group

• Great teamwork — Thank you Serpil+Tim!
• libParseBGP (see previous slides)
• Contributions to OpenBMP
  - NAT/PAT support
  - Router connection rate limiting
  - New (optional?) OpenBMP encapsulation format
  - Minor bug fixes
• Trained 2 UC San Diego master students

Ojas Gupta

Induja Sreekantan
TASK 3

Cooperate w/ OpenBMP group

- Dissemination of results & work in progress
- IRTF and IETF ’98 - (irtf-open and rtgwg)
- TMA PhD School, Dublin, Ireland (Lecture + Lab ~40 PhD students)

DATA COLLECTION
BGP Monitoring Protocol (BMP) - RFC 7854

- BMP encapsulates BGP messages a router receives from one or more BGP peers into a single TCP stream to one or more collectors
- Efficient, real-time, low memory/CPU on router; little to no service impact with peering
- Simplified configuration (one-time setup) with granular controls per peer
- All address families supported

DATA COLLECTION
OpenBMP/SNAS - www.openbmp.org

- Open-source collector that implements BMP to store and maintain data in both real-time and point-in-time (historical)
- The collector is a highly scalable producer to Apache Kafka. Both RAW BMP messages and parsed messages are produced for Kafka consumer consumption.

• Tim's Keynote at SIGCOMM BigData Workshop (BIG-DAMA)
• v2 ml-announcements and presentations (NANOG, IETF, ACM conferences, …) todo
V2-BETA1 AND APPS
Public BMP feed \texttt{bmp.bgpstream.caida.org:9092}  
BGPStream apps can read BMP  
Projects ready to use it  
\begin{itemize}
  \item \textbf{IODA} [\textit{IMC’16 and others}] - 24/7 Internet Outage detection  
    \hspace{1em} - speeding up BGP detection — dashboards at \texttt{ioda.caida.org}
  \item \textbf{ARTEMIS} [\textit{wip ToN}] - Self-operated BGP prefix hijacking detection and mitigation  
    \hspace{1em} - open source framework
  \item \textbf{SWIFT} [\textit{SIGCOMM’17}] - Fast rerouting upon remote outages \texttt{swift.ethz.ch}
    \hspace{1em} - downloadable demo
  \item \textbf{SuperSWIFT} [\textit{wip SIGCOMM}] - P4 version of Swift. Collab with Internet2  
    \hspace{1em} - analysis of data vs control plane
\end{itemize}
FUTURE WORK
V2 RELEASE

New features in addition to BMP

• BGPStream v2.0 expected to be released in Feb 2018
• New license: BSD
• v2 features:
  - RIPE RIS streaming support todo
  - RPKI validation (RTRlib) todo
  - Broker support for public BGPStream BMP feed todo
  - Local (optional) caching of dump files todo
  - New high-level Python API
  - New filter interface with a "BPF-like" syntax (hackathon contribution)
  - Performance improvements (new MRT parser, better resource management, …) todo
  - Bugfixes
BGP ANALYTICS
Prefix hijacking detection and more

- Leverages BGPStream
- Combines control-plane and data-plane measurements
  - detects interesting BGP events (e.g., MOAS, new edges in the topology, …)
  - and triggers traceroute measurements from Ark/RIPE probes
- Classifies events and generates alerts
- Visualization dashboard to analyze the events
- Based on NSF funding ending soon
THANKS

bgpstream.caida.org

bgpstream.caida.org/v2-beta

alberto@caida.org