BGPStream: a framework for historical analysis and real-time monitoring of BGP data

Chiara Orsini, Alistair King, Alberto Dainotti
alberto@caida.org

Center for Applied Internet Data Analysis
University of California, San Diego
Country-wide Internet outages in Iraq that the government ordered in conjunction with the ministerial preparatory exams - Jul 2015
Outage of AS11351 (Time Warner Cable LLC)
September 30, 2015

BGP EVENTS & DYNAMICS
IODA: Detection and Analysis of Internet Outages

www.caida.org/funding/ioda/
BGP EVENTS & DYNAMICS

Hijacks: detection of MITM BGP attacks

- **S**: source (poisoned)
- **D**: dest (hijacked prefix)
- **A**: attacker

normal path
hijacked path
normal path

[www.caida.org/funding/hijacks/](http://www.caida.org/funding/hijacks/)
IODA SYSTEM DIAGRAM
(toy diagram)
IODA SYSTEM DIAGRAM
(toy diagram)
The "prefix-monitor" plugin (distributed with source) monitors a set of IP ranges as they are seen from BGP monitors distributed worldwide:

- how many prefixes reachable
- how many origin ASes
- generates detailed logs

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)

```
from _pybgpstream import BGPStream, BGPRecord, BGPElem
from collections import defaultdict
from itertools import groupby
import networkx as nx

stream = BGPStream()
as_graph = nx.Graph()
rec = BGPRecord()

bgp_lens = defaultdict(lambda: defaultdict(lambda: defaultdict(int)))
stream.add_interval_filter((1438415400, 1438416600))
stream.start()

while(stream.get_next_record(rec):
    elem = rec.get_next_elem()
    while elem:
        monitor = str(elem.peer-as)
        hops = {h for h, g in groupby(elem.fields['as-path'].split(' ')) if len(g) == 1}
        orig = hops[-1]
        for i in range(0, len(hops)-1):
            as_graph.add_edge(hops[i], hops[i+1])
        bgp_lens[monitor][orig] +=
        min(1, bgp_lens[monitor][orig], len(hops)))
        elem = rec.get_next_elem()

for monitor in bgp_lens:
    nxlen = len(nx.shortest_path(as_graph, monitor, orig))
    print monitor, orig, bgp_lens[monitor][orig], nxlen
```

30 LINES OF PYTHON CODE
$ 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|
R|E|1445306401|routeviews|route-views.sfmix
U|A|1445306401|routeviews|route-views.sfmix|32354|2001:504:30::ba01:4061:1|2402:ef35::/32|
2001:504:30::ba01:4061:1|32354 6939 6453 4755 7633|7633|||
U|A|1445306401|routeviews|route-views.sfmix|14061|2014:300000|2a02:158:200::/39|
...
1. A web service (“BGPStream Broker”)  
   • enables SIMPLE access to LOTS of 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 LOTS of heterogeneous BGP sources

3. Command-line tools and APIs in C and Python
• 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
  - Easily extensible
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)

bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "rrc06");
bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "route-views.jinx");
bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_RECORD_TYPE, "updates");
bgpstream_add_interval_filter(bs, 1286705410, 1286709071);
```

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

---

Center for Applied Internet Data Analysis
University of California San Diego
libBGPStream keeps retrieving data as it becomes available

```python
stream.add_filter('record-type',
  'ribs')
stream.add_filter('collector',
  'route-views.sfmix')
stream.add_interval_filter(1445306400,
  -1)
bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "rrc06");
bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "route-views.jinx");
bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_RECORD_TYPE, "updates");
bgpstream_add_interval_filter(bs, 1286705410, BGPSTREAM_FOREVER);
```

$ bgpreader -c route-views.sfmix -t updates

$ bgpcorsaro -p ris
Access BMP-generated data from BGPStream

Data available with ~1 min latency

Developed in collaboration with Tim Evens @ Cisco and John Kemp @ Route Views

Experimental integration using OpenBMP to export MRT files (native BMP support planned for BGPStream)
BMP DATA SOURCES

Data Providers

- Current BMP feeds provided courtesy of Route Views, Cisco, and Randy Bush
BMP DATA SOURCES
don’t need to download a new BGPStream version

• Available to all existing BGPStream installs
  - Use filter to select data from provider “caida-bmp”
  - E.g. bgpreader -p caida-bmp -w 1453912260
• send us a bmp feed!
  - contact bgpstream-info@caida.org

olistair@gni:~$ bgpreader -p caida-bmp -w 1453912260 2>/dev/null | head -10
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV6_core1.sjc2.he.net/1693912001:470:0:1a::112a00:9380::/29/12001:470:0:1a::116939 12732112732111
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net/17018|12.0.1.631206, 208.95.0/24112.0.1.6317018 3356 4323 3728 19837 19837 19837 19837 198371983717018:5000 7018:3922011
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV6_core1.sjc2.he.net/1693912001:470:0:1a::112a00:9380::/29/12001:470:0:1a::116939 12732112732111
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net/17018|12.0.1.63179, 124.4.0/24112.0.1.6317018 3356 57344 601686016817018:5000 7018:3723211
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net/17018|12.0.1.63177, 136.84.0/23112.0.1.6317018 3356 3549 18881 263164 262485 264162 263132126313217018:5000 7018:3723211
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net/17018|12.0.1.63177, 136.80.0/24112.0.1.6317018 3356 3549 18881 263164 262485 264162 263132126313217018:5000 7018:3723211
UIA14540195021caida-bmp-router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net/17018|12.0.1.631206, 208.95.0/24112.0.1.6317018 3356 4323 3728 19837 19837 19837 198371983717018:5000 7018:3723211
THANKS

bgpstream.caida.org