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

Chiara Orsini, Alistair King, Alberto Dainotti

alberto@caida.org

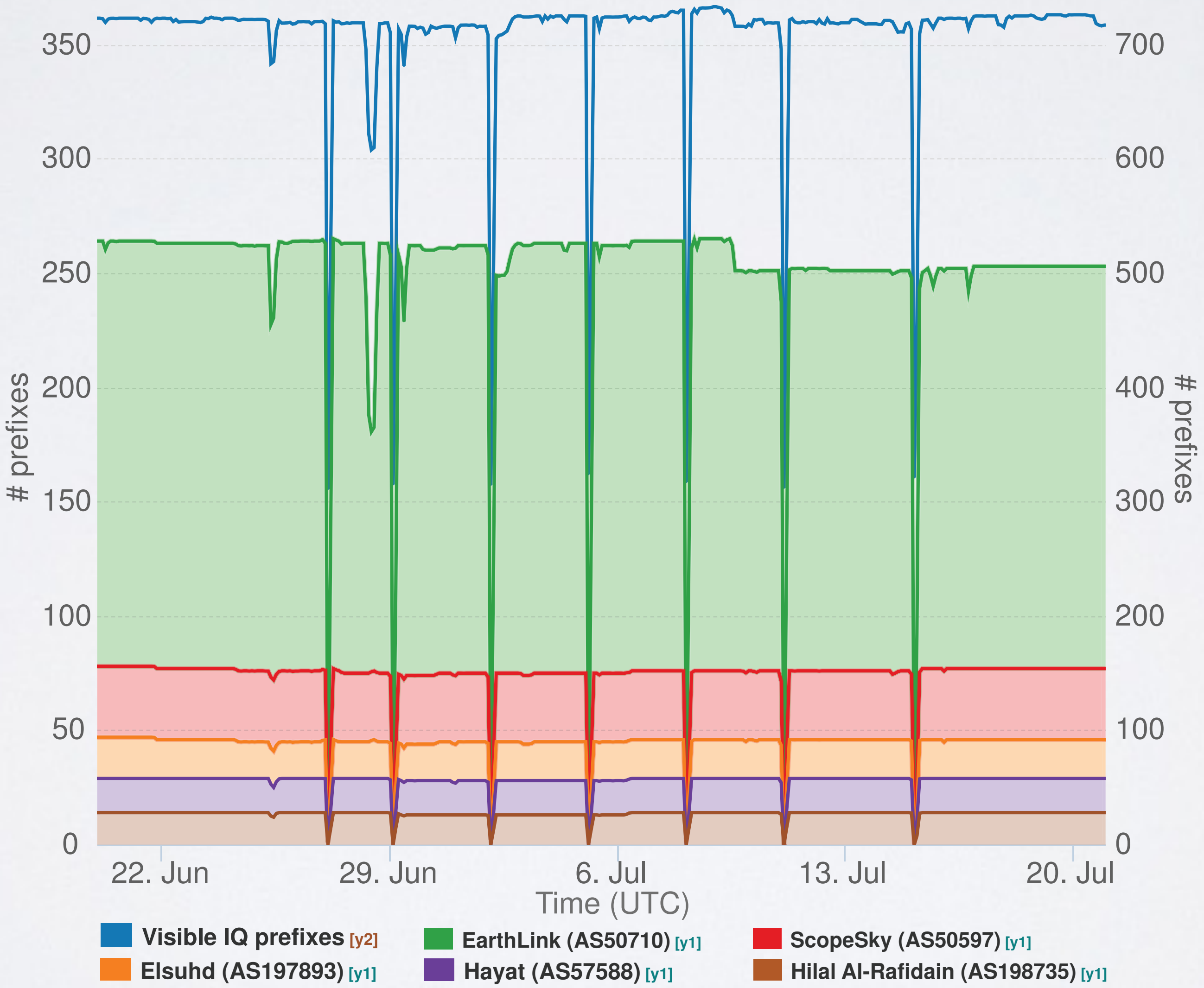
Center for Applied Internet Data Analysis

University of California, San Diego

BGP EVENTS & DYNAMICS

IODA: Detection and Analysis of Internet Outages

Country-wide Internet outages in Iraq that the government ordered in conjunction with the ministerial preparatory exams - Jul 2015



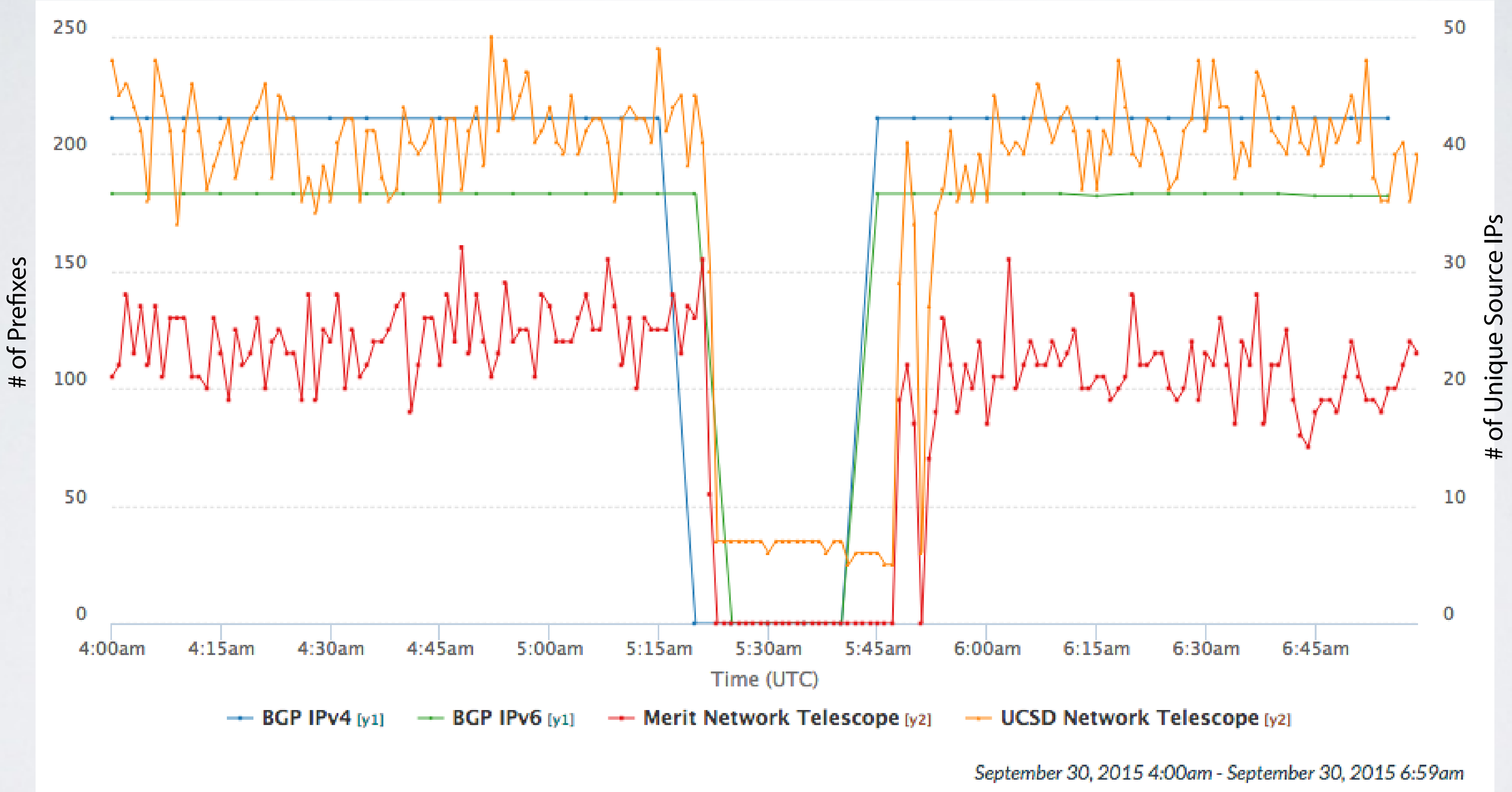
www.caida.org/funding/ioda/

COMCAST

BGP EVENTS & DYNAMICS

IODA: Detection and Analysis of Internet Outages

Outage of AS11351(Time Warner Cable LLC)
September 30, 2015

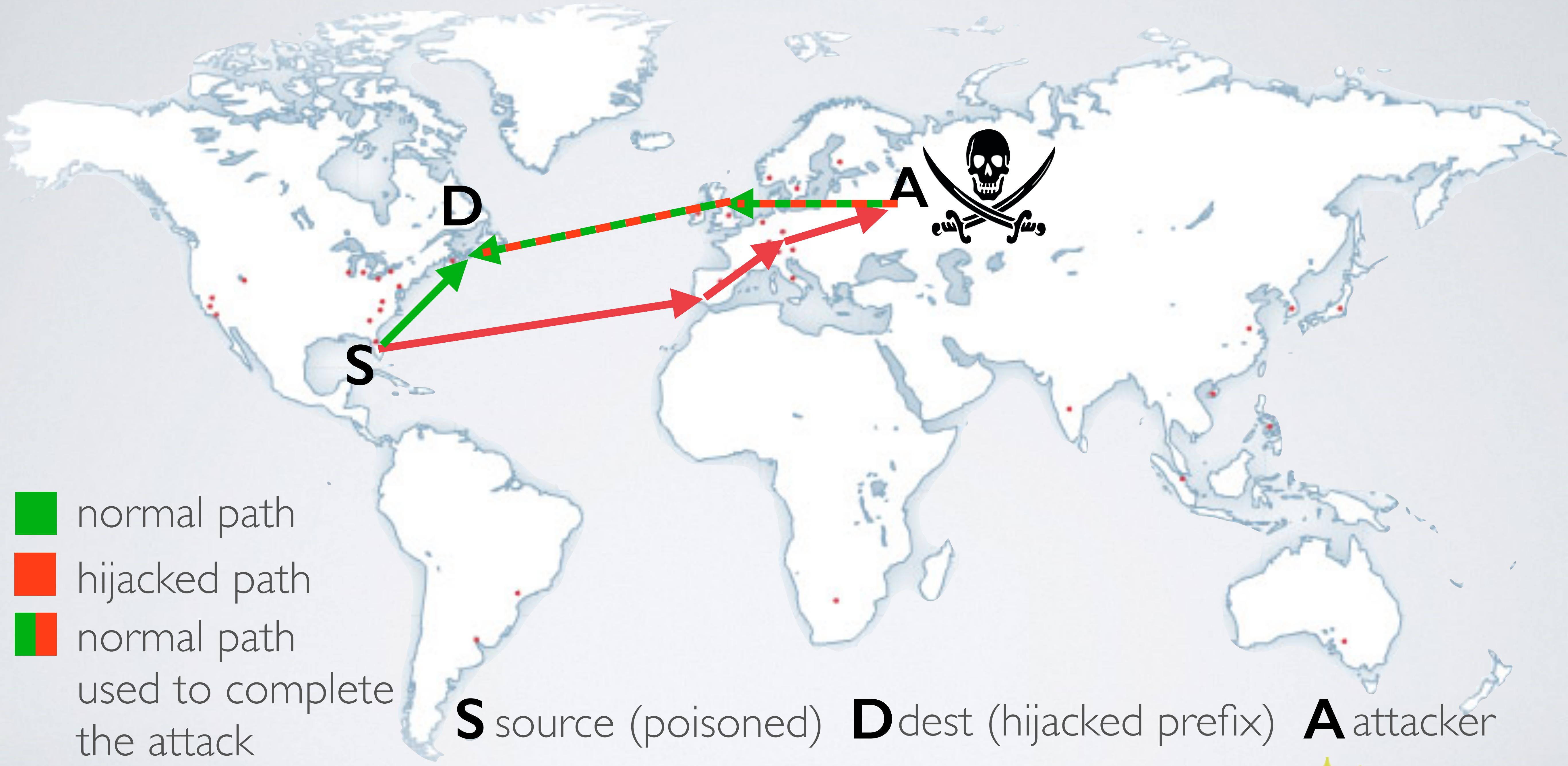


September 30, 2015 4:00am - September 30, 2015 6:59am

www.caida.org/funding/ioda/

BGP EVENTS & DYNAMICS

Hijacks: detection of MITM BGP attacks



🟩 normal path

🟥 hijacked path

🟩🟥 normal path used to complete the attack

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

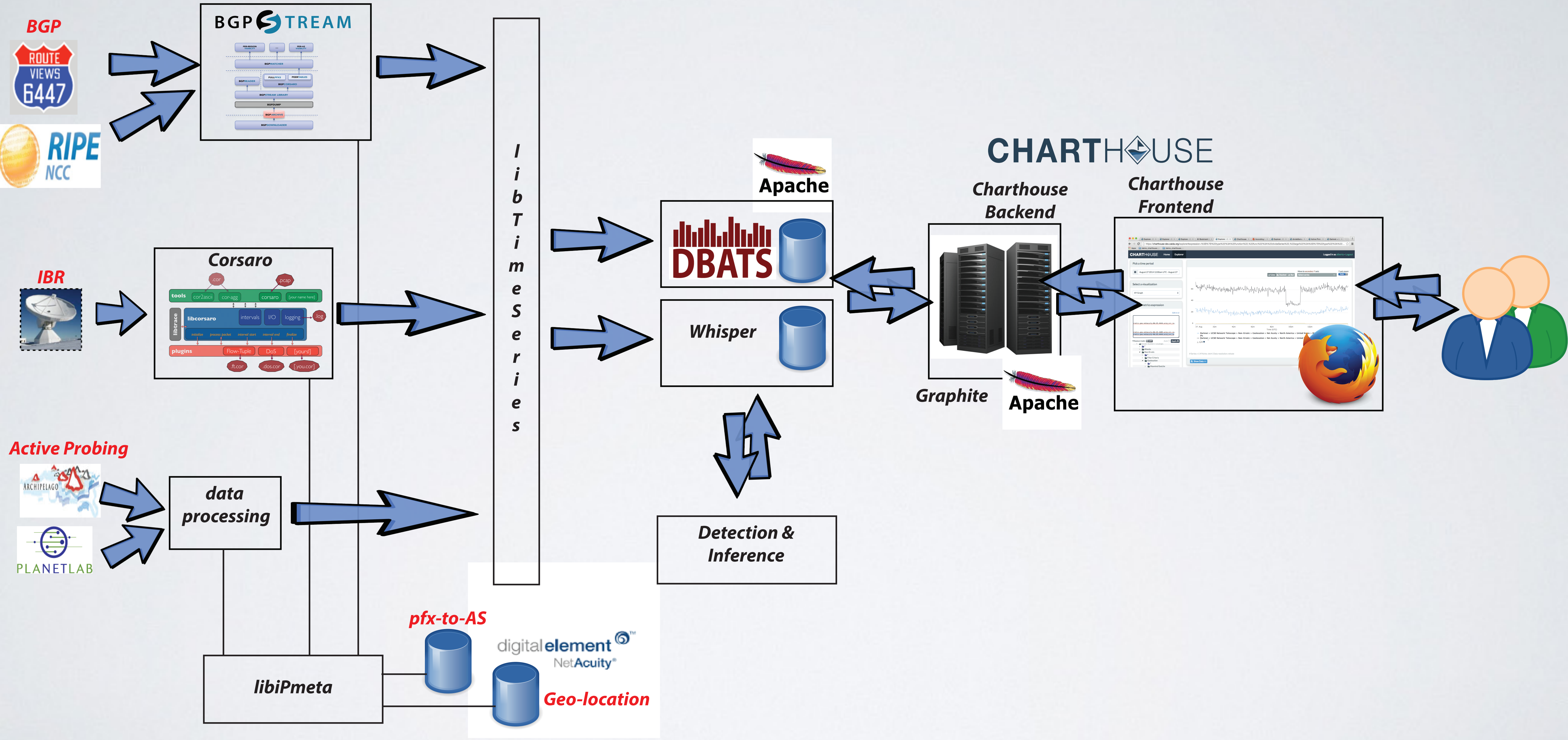
Center for Applied Internet Data Analysis
University of California San Diego

www.caida.org/funding/hijacks/

COMCAST

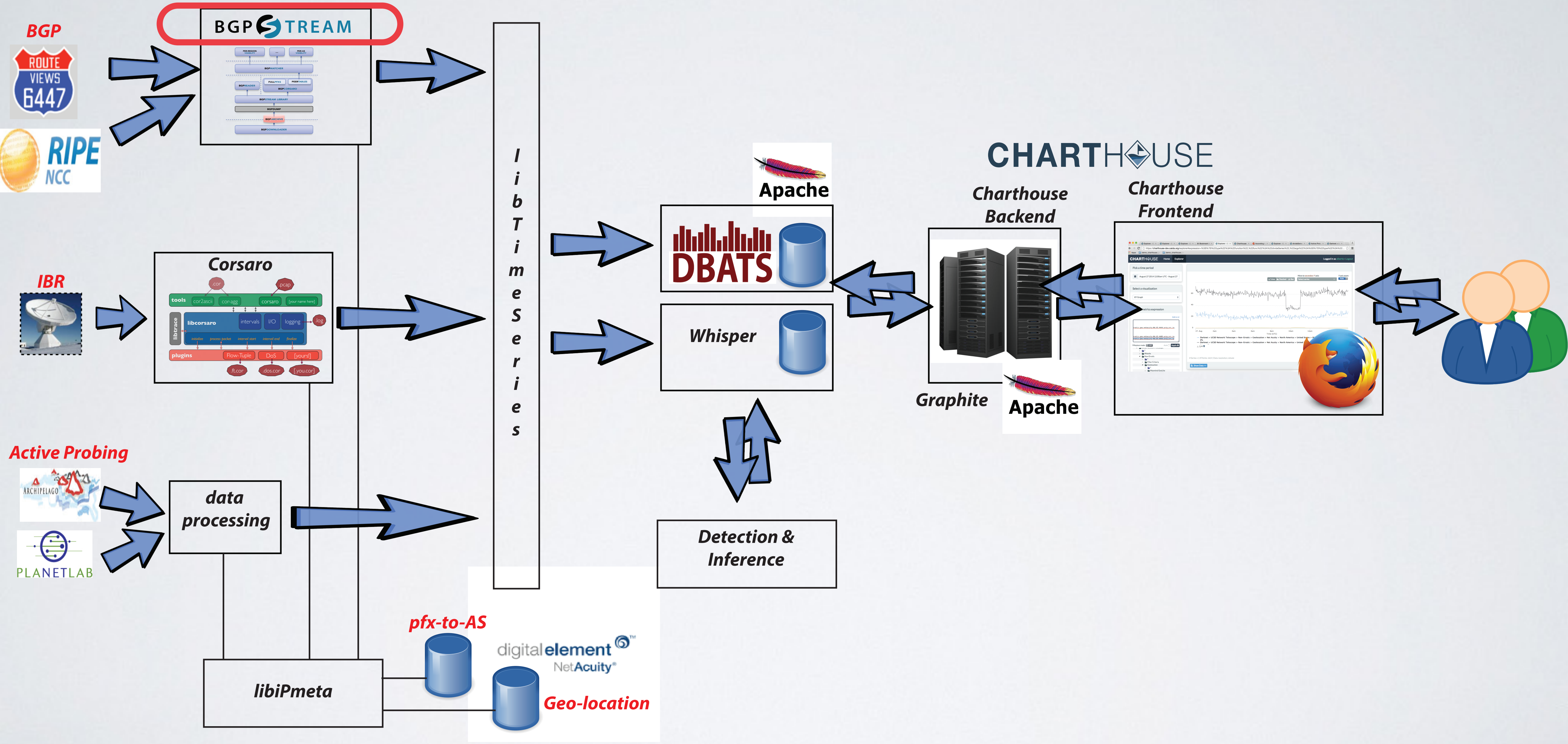
IODA SYSTEM DIAGRAM

(toy diagram)



IODA SYSTEM DIAGRAM

(toy diagram)



BGPSTREAM

bgpstream.caida.org

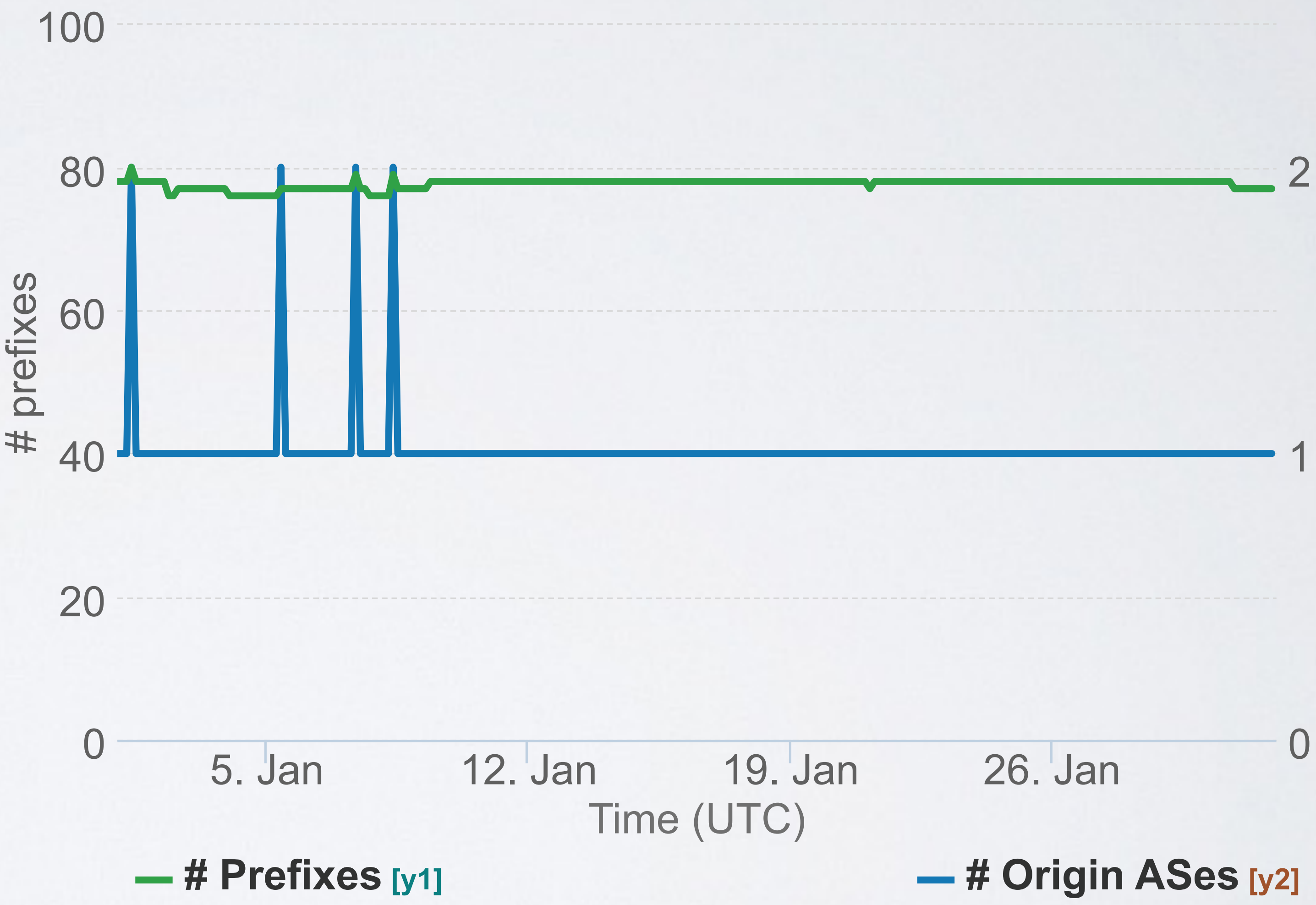
BGPCORSARO

Example: monitor your own address space on BGP

*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*

Hijacking of AS137 (GARR) - Jan 2015*

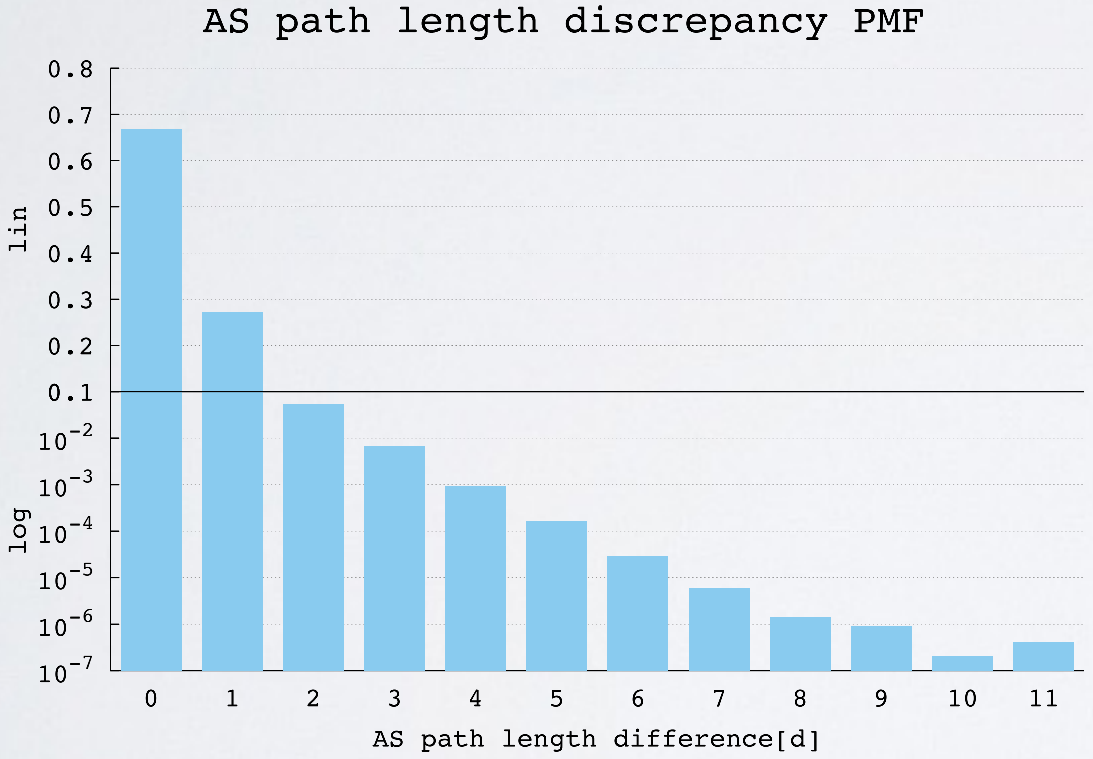


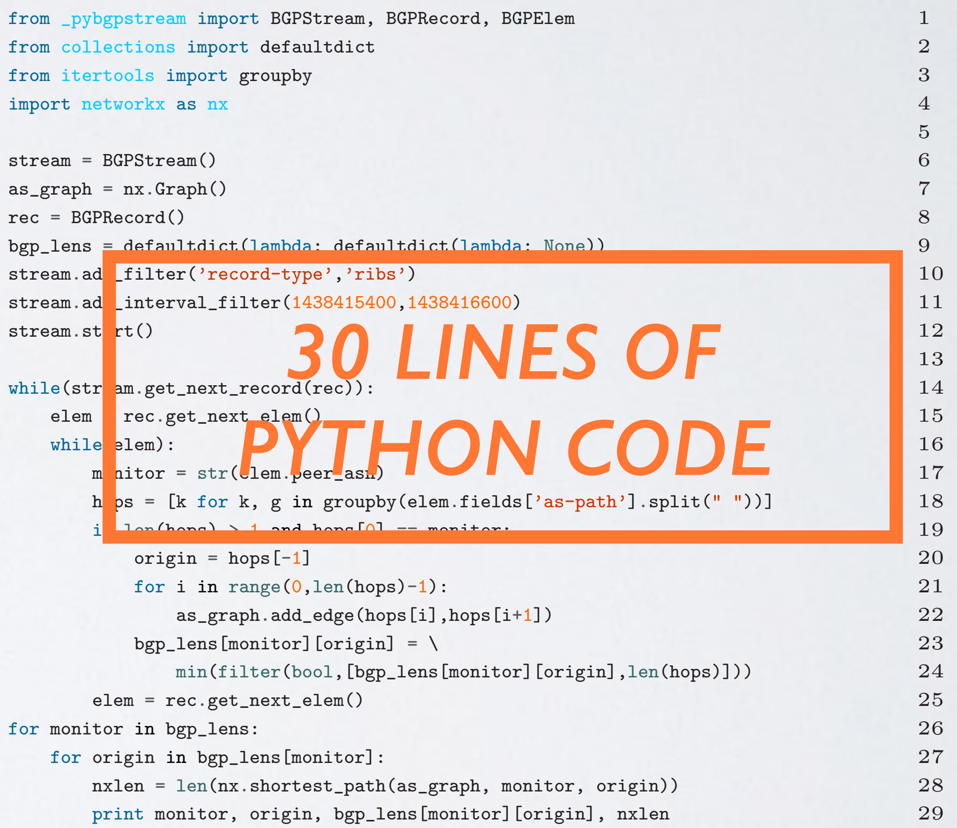
**Originally discovered by Dyn: http://research.dyn.com/2015/01/vast-world-of-fraudulent-routing/*

PYBGPSTREAM

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: None))
stream.add_filter('record-type','ribs')
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_asn)
        hops = [k for k, g in groupby(elem.fields['as-path'].split(" "))]
        if len(hops) > 1 and hops[0] == monitor:
            origin = hops[-1]
            for i in range(0,len(hops)-1):
                as_graph.add_edge(hops[i],hops[i+1])
            bgp_lens[monitor][origin] = \
                min(filter(bool,[bgp_lens[monitor][origin],len(hops)]))
        elem = rec.get_next_elem()
for monitor in bgp_lens:
    for origin in bgp_lens[monitor]:
        nxlen = len(nx.shortest_path(as_graph, monitor, origin))
        print monitor, origin, bgp_lens[monitor][origin], nxlen
```

30 LINES OF PYTHON CODE

BGPREADER

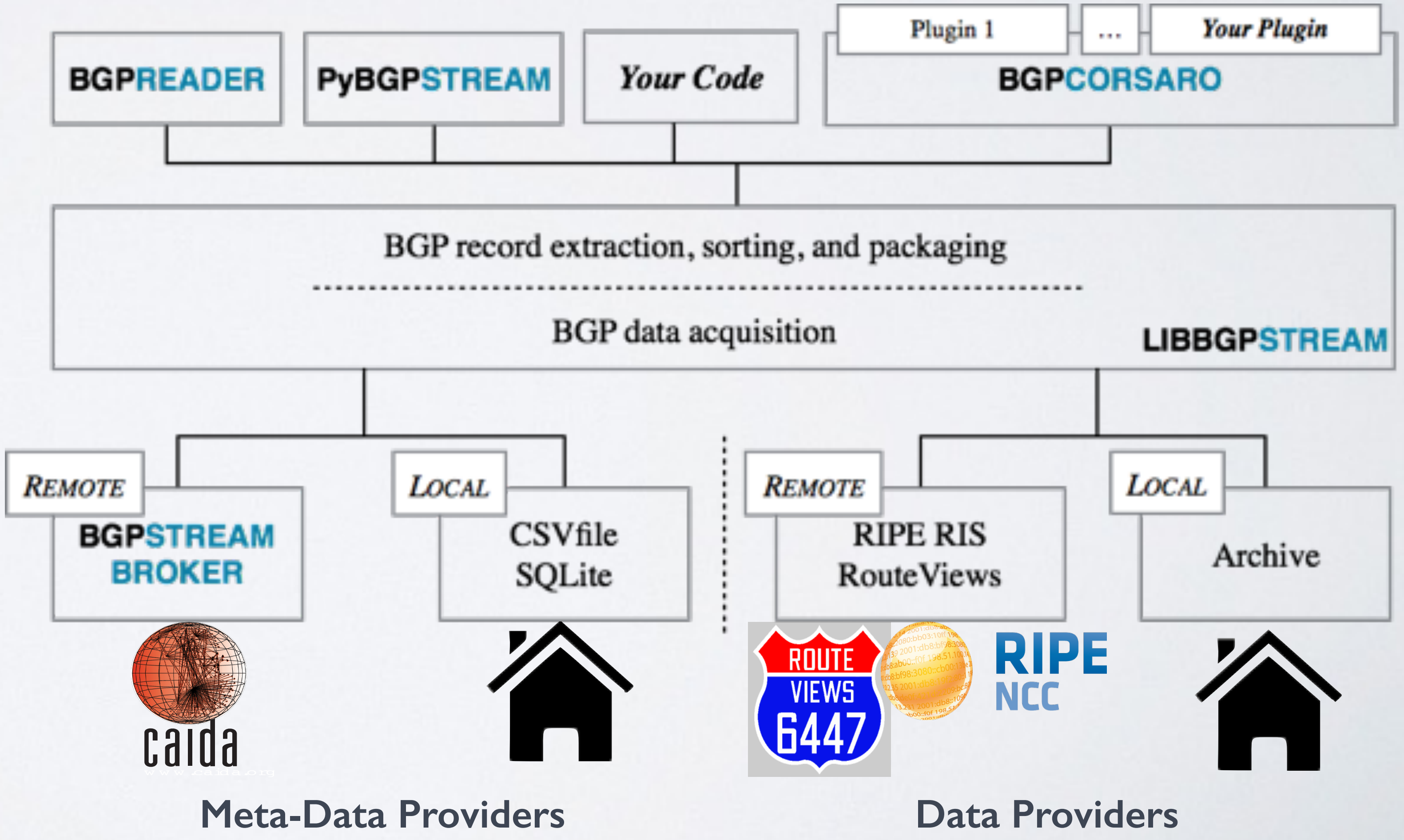
command-line tool for ASCII output w/ filters

```
$ 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|32354 6939 37105 33762|33762|||
R|R|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|3803:b600::/32|
2001:504:30::ba01:4061:1|14061 2914 3549 27751|27751|2914:420 2914:1008 2914:2000 2914:3000||
R|E|1445306401|routeviews|route-views.sfmix
U|A|1445306401|routeviews|route-views.sfmix|32354|2001:504:30::ba03:2354:1|2402:ef35::/32|
2001:504:30::ba03:2354:1|32354 6939 6453 4755 7633|7633|||
U|A|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|2a02:158:200::/39|
2001:504:30::ba01:4061:1|14061 2914 44946|44946|2914:410 2914:1201 2914:2202 2914:3200||
...
```

BGPSTREAM

bgpstream.caida.org

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*



Center for Applied Internet Data Analysis
University of California San Diego

BGPSTREAM

bgpstream.caida.org

- 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

BGPStream: a software framework for live and historical BGP data analysis

Chiara Orsini, Alistair King, Alberto Dainotti
CAIDA, UC San Diego

ABSTRACT

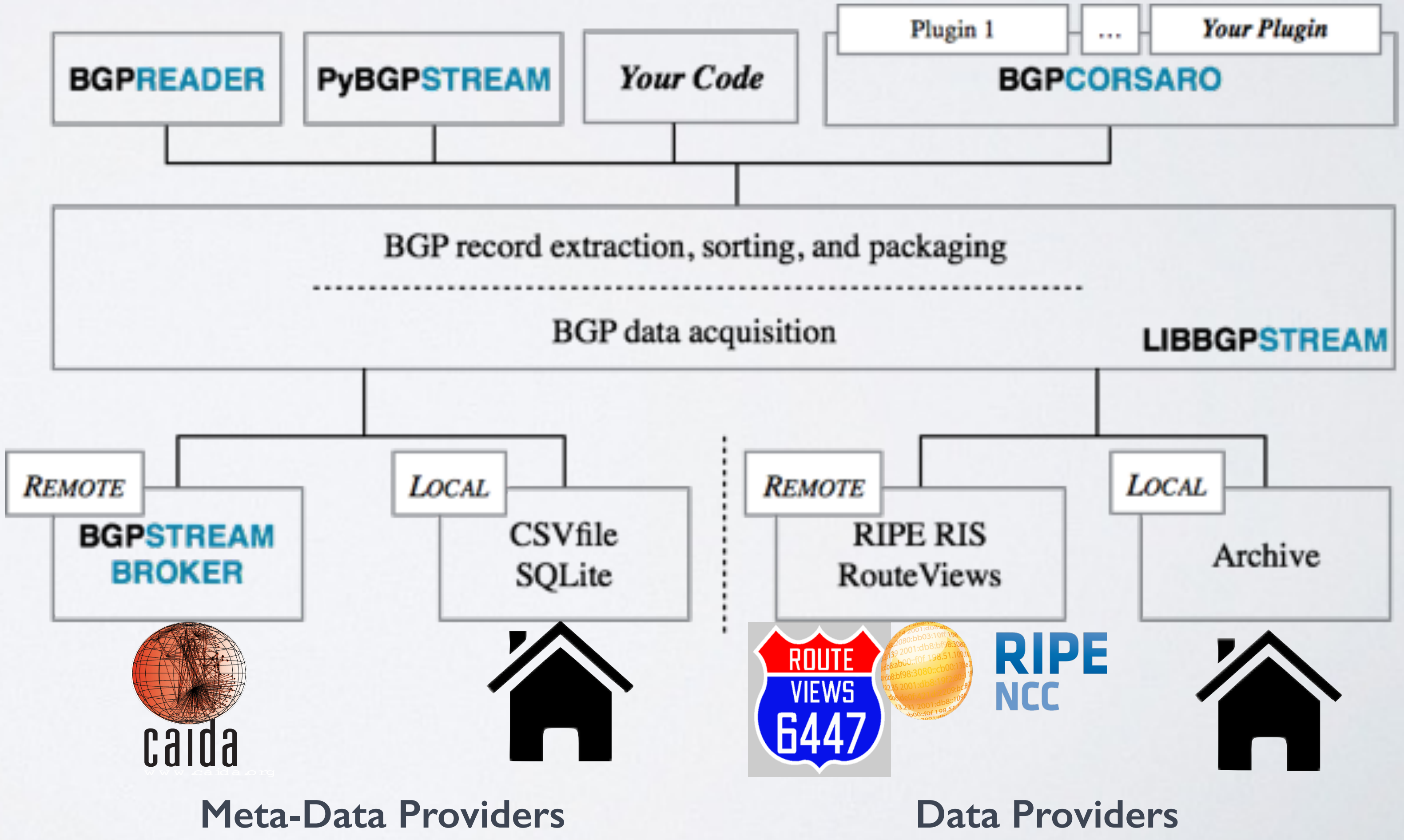
[illegible]

1. INTRODUCTION

[illegible]

2. BACKGROUND

[illegible]



NO MANUAL DOWNLOADS

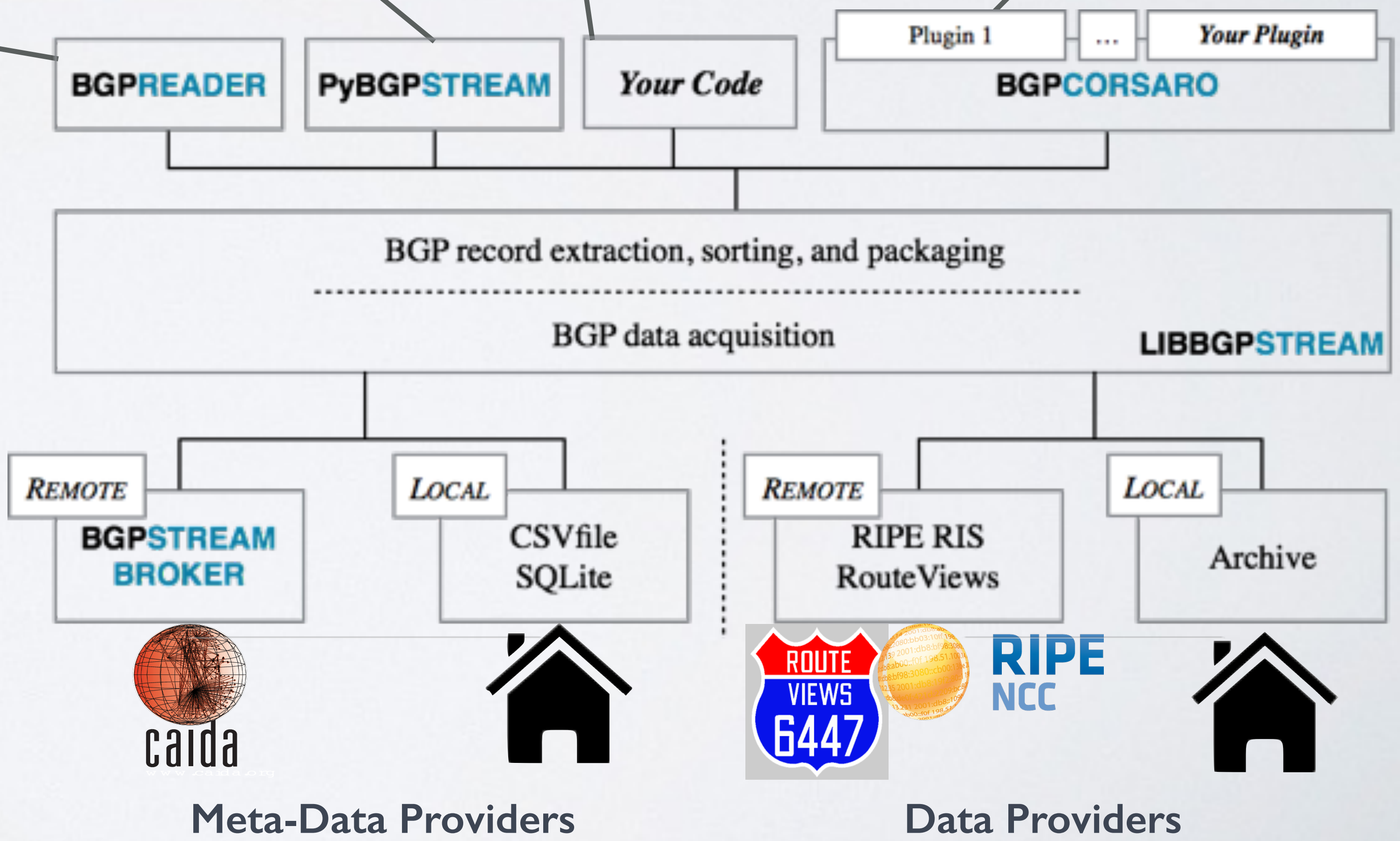
libBGPStream talks to the broker and gets the data

```
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);
```

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

```
$ bgpcorsaro -w 1445306400,1445306402 -p ris
```

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



GET A LIVE STREAM

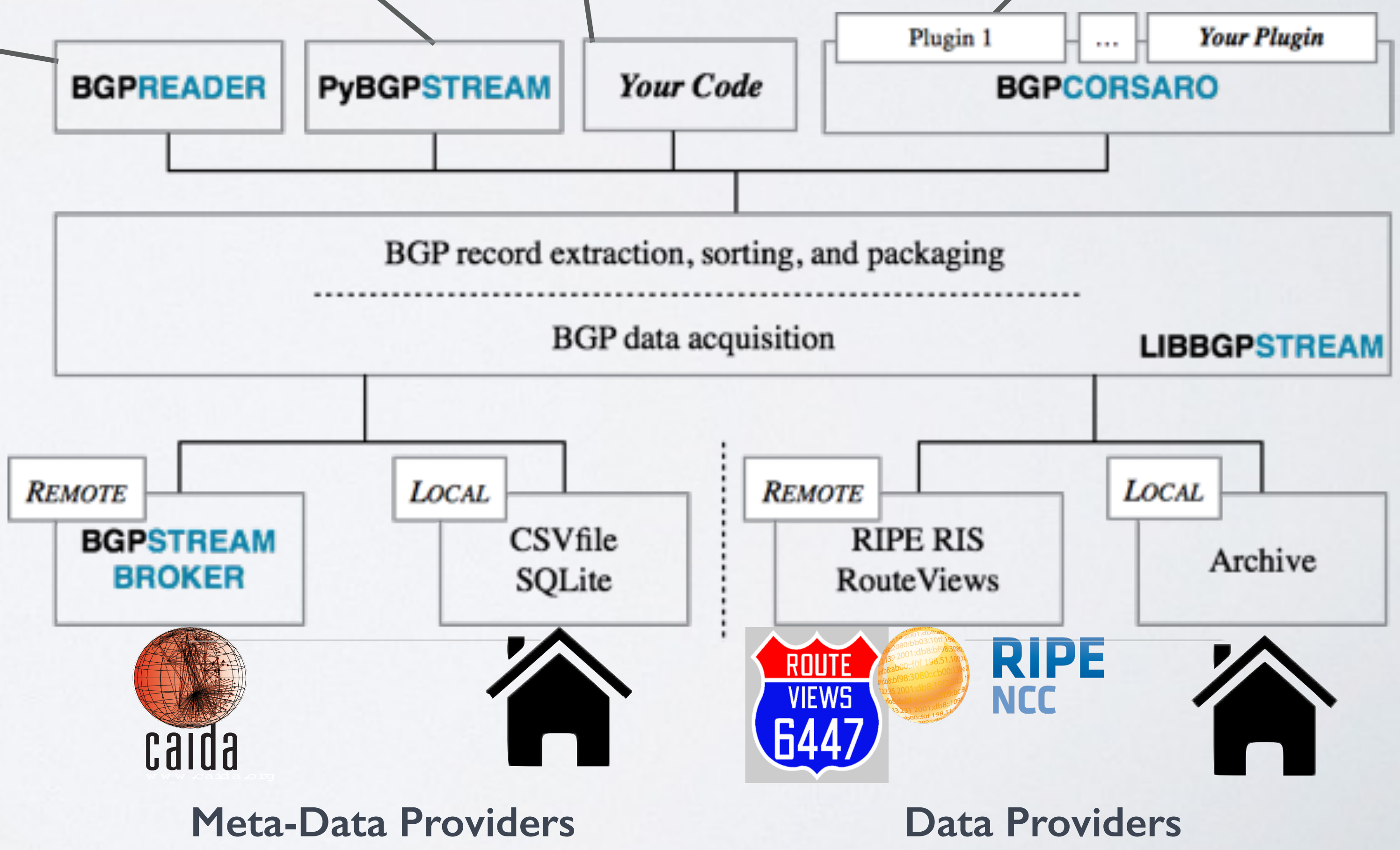
libBGPStream keeps retrieving data as it becomes available

```
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);
```

```
stream.add_filter(‘record-type’, ‘ribs’)
stream.add_filter(’collector’, ‘route-views.sfmix’)
stream.add_interval_filter(1445306400,-1)
```

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

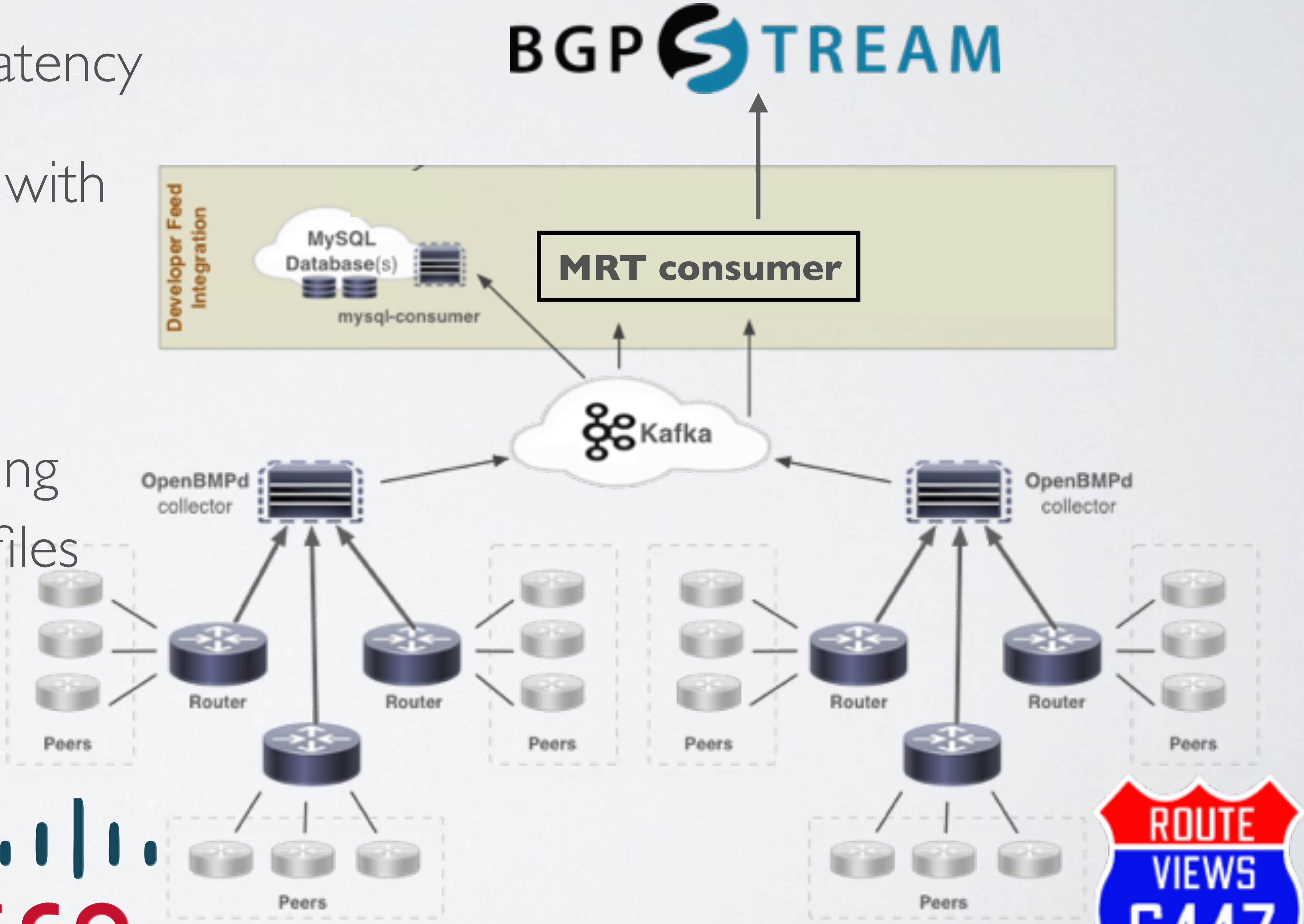
```
$ bgpcorsaro -p ris
```



BMP DATA SOURCES

(experimental)

- Access BMP-generated data from BGPStream
- Data available with ~1min 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)

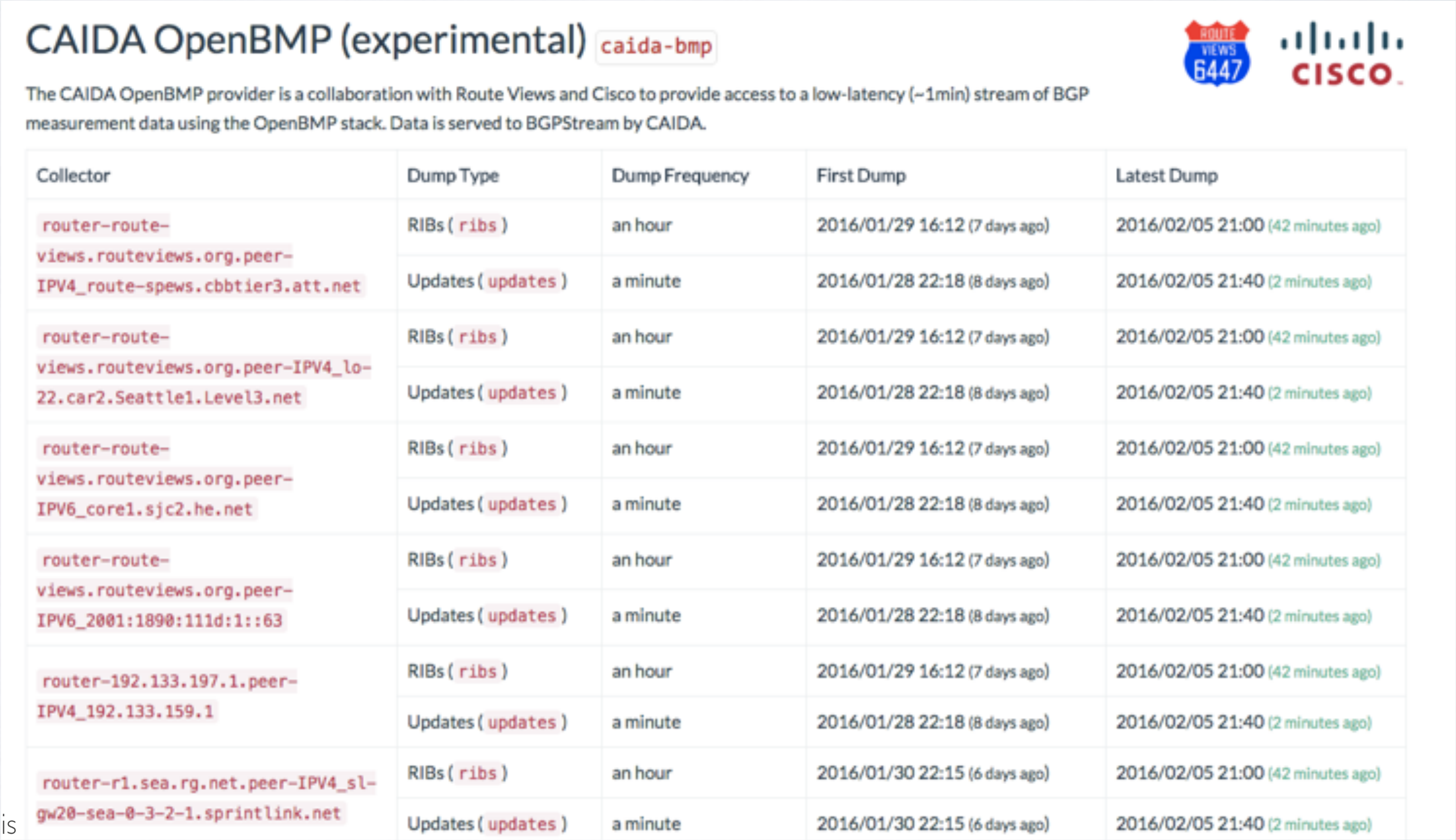


Center for Applied Internet Data Analysis
University of California San Diego

BMP DATA SOURCES

Data Providers

- Current BMP feeds provided courtesy of **Route Views**, **Cisco**, and **Randy Bush**



CAIDA OpenBMP (experimental) `caida-bmp`

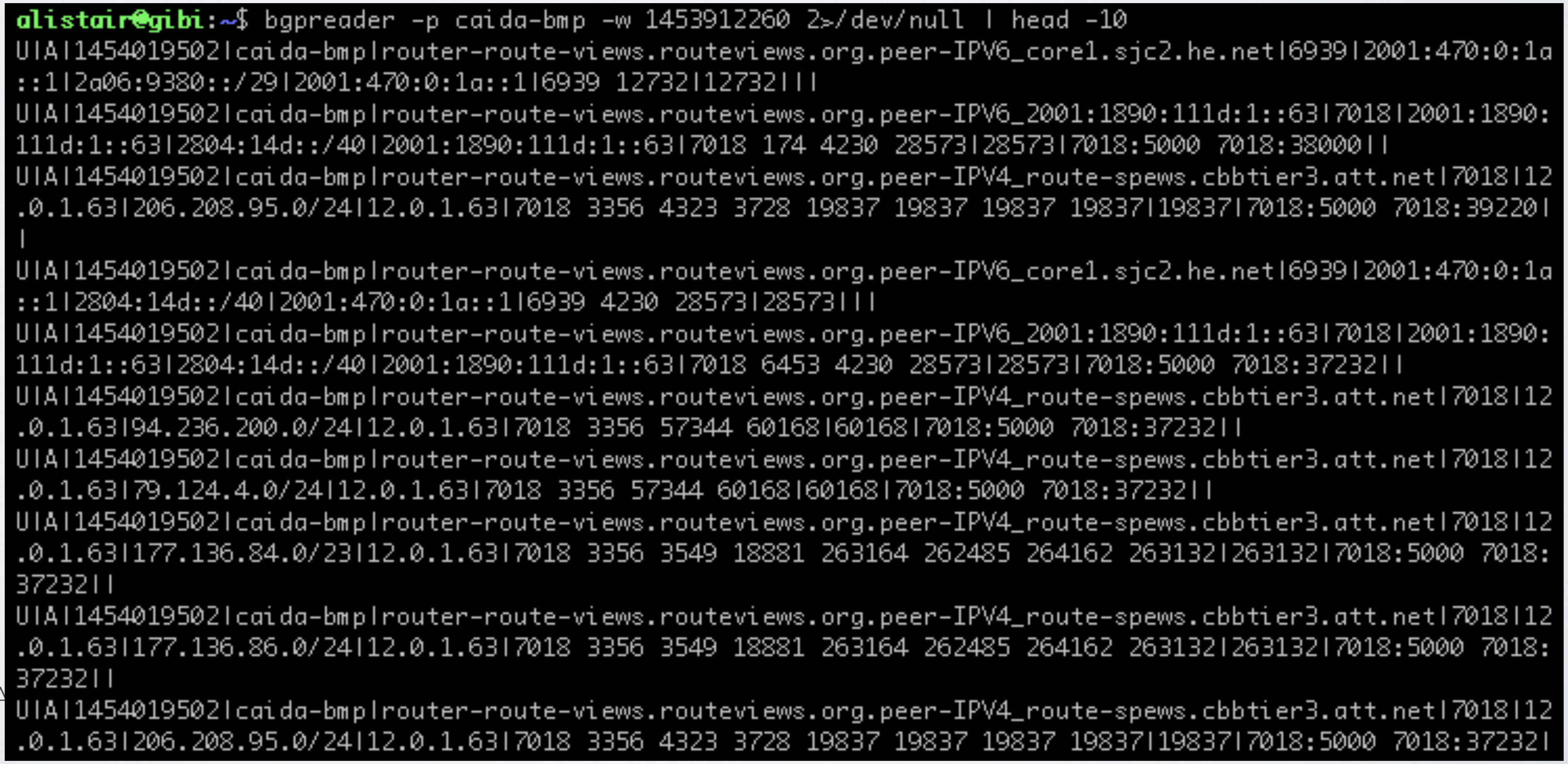
The CAIDA OpenBMP provider is a collaboration with Route Views and Cisco to provide access to a low-latency (~1min) stream of BGP measurement data using the OpenBMP stack. Data is served to BGPStream by CAIDA.

Collector	Dump Type	Dump Frequency	First Dump	Latest Dump
`router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net`	RIBs (`ribs`)	an hour	2016/01/29 16:12 (7 days ago)	2016/02/05 21:00 (42 minutes ago)
	Updates (`updates`)	a minute	2016/01/28 22:18 (8 days ago)	2016/02/05 21:40 (2 minutes ago)
`router-route-views.routeviews.org.peer-IPV4_lo-22.car2.Seattle1.Level3.net`	RIBs (`ribs`)	an hour	2016/01/29 16:12 (7 days ago)	2016/02/05 21:00 (42 minutes ago)
	Updates (`updates`)	a minute	2016/01/28 22:18 (8 days ago)	2016/02/05 21:40 (2 minutes ago)
`router-route-views.routeviews.org.peer-IPV6_core1.sjc2.he.net`	RIBs (`ribs`)	an hour	2016/01/29 16:12 (7 days ago)	2016/02/05 21:00 (42 minutes ago)
	Updates (`updates`)	a minute	2016/01/28 22:18 (8 days ago)	2016/02/05 21:40 (2 minutes ago)
`router-route-views.routeviews.org.peer-IPV6_2001:1890:111d:1::63`	RIBs (`ribs`)	an hour	2016/01/29 16:12 (7 days ago)	2016/02/05 21:00 (42 minutes ago)
	Updates (`updates`)	a minute	2016/01/28 22:18 (8 days ago)	2016/02/05 21:40 (2 minutes ago)
`router-192.133.197.1.peer-IPV4_192.133.159.1`	RIBs (`ribs`)	an hour	2016/01/29 16:12 (7 days ago)	2016/02/05 21:00 (42 minutes ago)
	Updates (`updates`)	a minute	2016/01/28 22:18 (8 days ago)	2016/02/05 21:40 (2 minutes ago)
`router-r1.sea.rg.net.peer-IPV4_sl-gw20-sea-0-3-2-1.sprintlink.net`	RIBs (`ribs`)	an hour	2016/01/30 22:15 (6 days ago)	2016/02/05 21:00 (42 minutes ago)
	Updates (`updates`)	a minute	2016/01/30 22:15 (6 days ago)	2016/02/05 21:40 (2 minutes ago)

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*



```
alistair@gibi:~$ bgpreader -p caida-bmp -w 1453912260 2>/dev/null | head -10
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV6_core1.sjc2.he.net|6939|2001:470:0:1a
::1|2a06:9380::/29|2001:470:0:1a::1|6939 12732|12732|||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV6_2001:1890:111d:1::63|7018|2001:1890:
111d:1::63|2804:14d::/40|2001:1890:111d:1::63|7018 174 4230 28573|28573|7018:5000 7018:38000||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net|7018|12
.0.1.63|206.208.95.0/24|12.0.1.63|7018 3356 4323 3728 19837 19837 19837 19837|19837|7018:5000 7018:39220|
|
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV6_core1.sjc2.he.net|6939|2001:470:0:1a
::1|2804:14d::/40|2001:470:0:1a::1|6939 4230 28573|28573|||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV6_2001:1890:111d:1::63|7018|2001:1890:
111d:1::63|2804:14d::/40|2001:1890:111d:1::63|7018 6453 4230 28573|28573|7018:5000 7018:37232||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net|7018|12
.0.1.63|94.236.200.0/24|12.0.1.63|7018 3356 57344 60168|60168|7018:5000 7018:37232||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net|7018|12
.0.1.63|79.124.4.0/24|12.0.1.63|7018 3356 57344 60168|60168|7018:5000 7018:37232||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net|7018|12
.0.1.63|177.136.84.0/23|12.0.1.63|7018 3356 3549 18881 263164 262485 264162 263132|263132|7018:5000 7018:
37232||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net|7018|12
.0.1.63|177.136.86.0/24|12.0.1.63|7018 3356 3549 18881 263164 262485 264162 263132|263132|7018:5000 7018:
37232||
U|A|1454019502|caida-bmp|router-route-views.routeviews.org.peer-IPV4_route-spews.cbbtier3.att.net|7018|12
.0.1.63|206.208.95.0/24|12.0.1.63|7018 3356 4323 3728 19837 19837 19837 19837|19837|7018:5000 7018:37232|
```

THANKS

bgpstream.caida.org