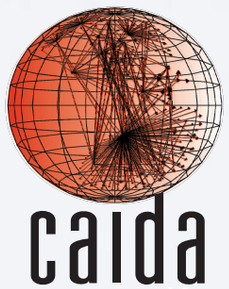
7th TMA PhD School on Traffic Monitoring and Analysis
Dublin, Ireland, 19th Jun 2017

BGP measurement and live data analysis

Alberto Dainotti
alberto@caida.org



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

BGP QUICK TOUR

BGP

intro

- What is BGP?
  - Border Gateway Protocol - RFC 4271
  - The routing protocol of the Internet, used to route traffic across the Internet
- What are ASes?
  - Each routing domain is known as an Autonomous System, or AS
  - Each AS has an AS number (ASN), assigned by RIRs
- Again, what is BGP?
  - BGP helps to choose a path through the Internet, usually by selecting a route that traverses the least number of autonomous systems: the shortest AS Path.
  - Each AS announces to the others, by means of BGP update messages, the routes (AS Paths made of ASN hops) to its local prefixes and the preferred routes learned from its neighbors. (Path Vector routing protocol)
  - It's used also internally to make multiple BGP routers within the same AS exchange routes (IBGP). But we're mostly interested in Inter-AS dynamics here (EBGP)

BGP PACKET FORMAT

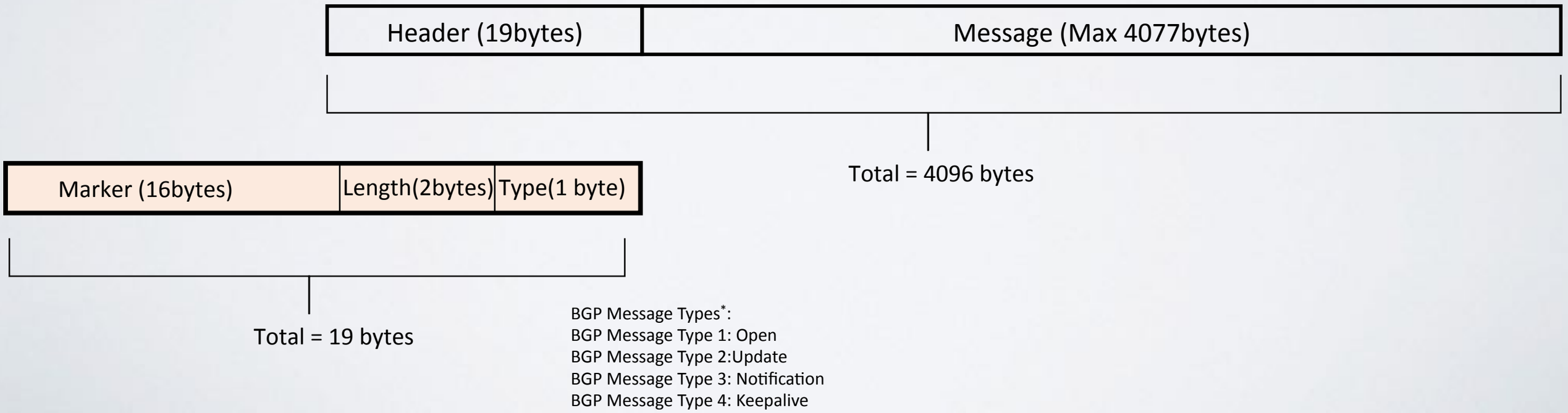
- BGP is a Layer 4 protocol that sits on top of TCP

Each BGP packet (or message) includes a Header*

Min size of a BGP packet: 19bytes (header only)

Max size of a BGP packet: 4096 (including header)

All fields network byte order (big endian, left to right)



BGP SESSIONS

- BGP neighbors (*peers*) are established by manual configuration between routers to create a BGP session on top of a TCP session on port 179.

- Supposed to stay up all time
  - keepalive message (e.g., every 30s). If no messages within *hold time* (e.g., 90s) the session is shut down
    - shutdown removes all prefixes received over the terminated session

- Open, Keepalive, Notification messages

ADVERTISE & WITHDRAW

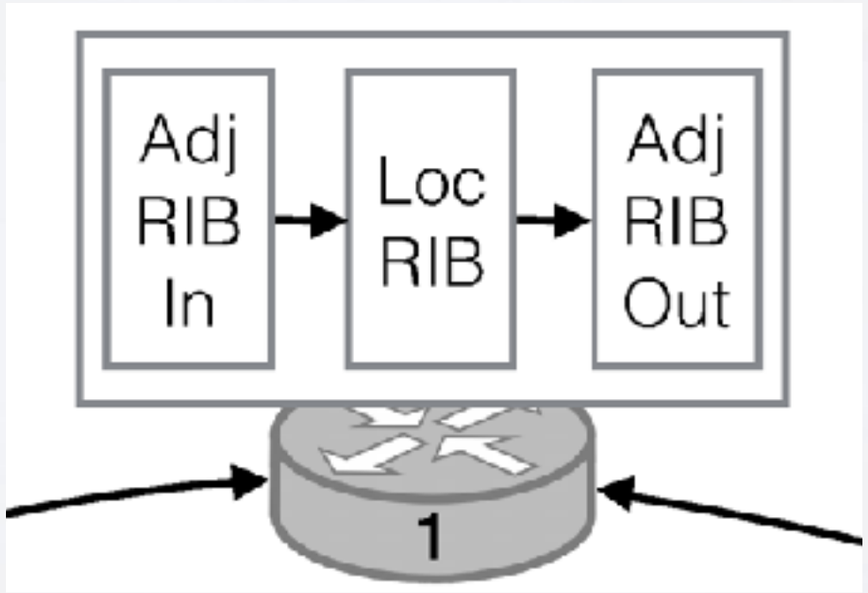
- Update messages are used to transfer routing information between BGP peers

- ***Advertisement***
- AS PATH
  - A router adds its AS number to a route's AS_PATH only when the route is sent to an EBGP neighbor.
  - Convention in writing an AS path: *[F, E, D, C, B, A:10.0.1.0/24]*
    - F adds ("prepends") its ASN before advertising the AS_PATH to its neighbors
- Loop avoidance

- ***Withdrawal***
  - *only prefix info (no path)*

RIBS TABLES AND FILTERS

- A BGP router maintains reachability information in the *Routing Information Base* (RIB), which is structured in three sets:
  - **Adj-RIBs-In**: routes learned from inbound update messages from its neighbors.
  - **Loc-RIB**: routes selected from Adj-RIBs-In by applying local policies (e.g., shortest path, peering relationships with neighbors); the router will install these routes in its routing table to establish where to forward packets.
  - **Adj-RIBs-Out**: routes selected from Loc-RIB, which the router will announce to its neighbors; for each neighbor the router creates a specific Adj-RIB-Out based on local policies (e.g., peering relationship).



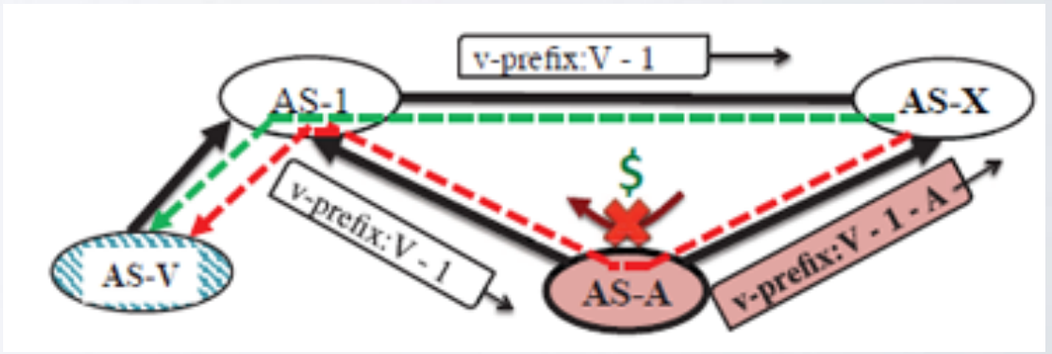
RELATIONSHIPS

- Stub vs Transit

- Economic relationships
  - Provider-to-customer *(p2c)*
  - Peer-to-peer *(p2p)*
  - Sibling-to-sibling *(s2s)*

- Relationships between neighbors determine preferences and import/export policies (*e.g., prefer a customer over a provider*)

- Gao-Rexford Model
  - Valley-free assumption: an AS does not transit traffic at a revenue loss
  - ***L. Gao, J. Rexford, "Stable Internet routing without global coordination", SIGMETRICS 2000***



"IT'S COMPLICATED"

- More complex relationships
  - ***Giotsas et al. "Inferring Complex AS Relationships", IMC 2014***
  - ***Anwar et al. "Investigating Interdomain Routing Policies in the Wild ", IMC 2015***
- MOAS - Multi Origin-AS conflicts
  - ***Zhao et al. "An analysis of BGP multiple origin AS (MOAS) conflicts" IMW 2001***
  - ***Jacquemart et al. "A Longitudinal Study of BGP MOAS Prefixes", TMA 2014***
- LIES!
  - BGP Hijacking, misconfiguration, …
- AS_SETS
  - born to deal with aggregation. Used to play tricks too.
- Path prepending
  - e.g., used to set up backup links
- BGP Communities attribute [RFC1997]
- Address family: ability to distribute synch messages for:
  - v4, v6, VPNs, flowspec, …

WHY CARE

MEASURING BGP

Why?

BGP is the central nervous system of the Internet

BGP's design is known to contribute to issues in:

- **Availability**
  - -Labovitz et al. "*Delayed Internet Routing Convergence*", IEEE/ACM Trans. Netw., 2001.
  - -Varadhan et al. "*Persistent Route Oscillations in Inter-domain Routing*". Computer Networks, 2000.
  - -Katz-Bassett et al. "*LIFEGUARD: Practical Repair of Persistent Route Failures*", SIGCOMM, 2012.
- **Performance**
  - -Spring et al. "*The Causes of Path Inflation*". SIGCOMM, 2003.
- **Security**
  - -Zheng et al. "*A Light-Weight Distributed Scheme for Detecting IP Prefix Hijacks in Realtime*". SIGCOMM, 2007.

Need to *engineer* protocol evolution!

MEASURING BGP

Why?

Defining problems and make ***protocol engineering*** decisions through realistic evaluations is difficult also because **we know little about the <u>structure</u> and <u>dynamics</u> of the BGP ecosystem!**

- AS-level topology
  -Gregori et al. *"On the incompleteness of the AS-level graph: a novel methodology for BGP route collector placement"*, IMC 2012
- AS relationships
  - Giotsas et al. *"Inferring Complex AS Relationships"*, IMC 2014
- AS interactions: driven by relationships, policies, network conditions, operator updates
  - Anwar et al. *"Investigating Interdomain Routing Policies in the Wild "*, IMC 2015
  - Lychev et al. *"BGP Security in Partial Deployment: Is the Juice Worth the Squeeze?"*, SIGCOMM

TOOLS OF THE TRADE

MEASURING BGP

data cycle

Attempts to generate more info
***(not much traction in the past)*:**

- `RFC 4384 BGP Communities for Data Collection`
- `draft-ymbk-grow-bgp-collector-communities`

SOFT ROUTERS

Testbeds, Route Servers, Route Reflectors, …

- Quagga
  - A routing software suite providing implementations of OSPFv2, OSPFv3, RIP v1 and v2, RIPng and BGP-4 for Unix platforms
- Bird
  - The BIRD project aims to develop a fully functional dynamic IP routing daemon primarily targeted on (but not limited to) Linux, FreeBSD
- GoBGP
  - an open source BGP implementation designed from scratch for modern environment and implemented in a modern programming language, the Go Programming Language
    - https://github.com/osrg/gobgp

ROUTE SERVERS

RFC7947

- "Multilateral interconnection is a method of exchanging routing information among three or more External BGP (EBGP) speakers using a single intermediate **broker** system, referred to as a route server. Route servers are typically used on shared access media networks, such as IXPs, to facilitate simplified interconnection among multiple Internet routers."

- "Although a route server uses BGP to exchange reachability information with each of its clients, it does not forward traffic itself and is therefore not a router."

https://ripe72.ripe.net/presentations/97-RIPE72_05-16.pdf

MEASURING BGP

Data Collection

- **Looking Glasses**
- **Route Collectors**
- **BMP**

DATA COLLECTION

Looking Glasses

- A telnet or Web interface to routers or route servers
  - e.g., telnet to *route-views.oregon-ix.net* allows a subset of "show ip bgp commands"
  - *http://lg.pch.net*
- BGP looking glasses give users limited (e.g., read-only) access to a command line interface of a router, or allow them to download the ASCII output of the current state of the router RIB.
- Several allow traceroute/ping!
- *More useful for interactive exploration (e.g., troubleshooting) rather than systematic and continuous data acquisition.*

http://www.traceroute.org/
Looking Glass WIKI at http://www.bgp4.net/

Ref. https://www.nanog.org/meetings/nanog33/presentations/gibbard.pdf

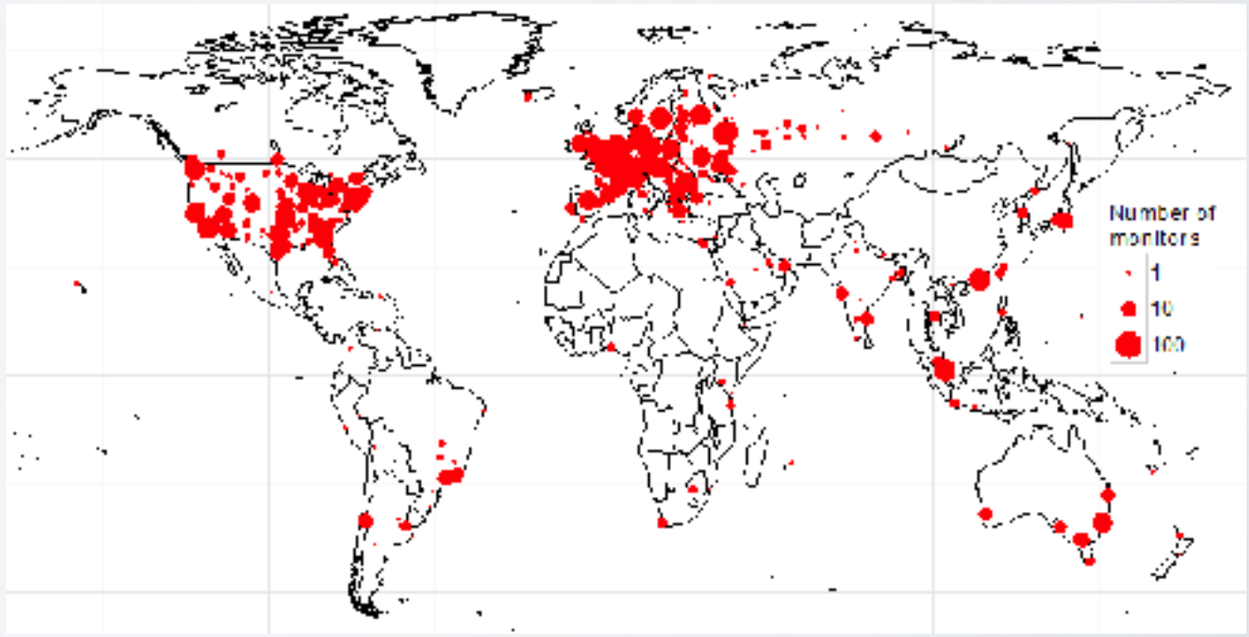
DATA COLLECTION

Looking Glasses - Periscope

- LGs are among the few public measurement tools that provide direct interfaces to routers and control+data plane access.
- Lack of standardization and consistency
- No centralized index of LGs, their locations and their capabilities
- Periscope: a unified API to LGs
  - implements a common querying scheme, indexing and data persistence features

Giotsas et al., "Periscope: Unifying Looking Glass Querying", PAM 2016

http://www.caida.org/tools/utilities/looking-glass-api

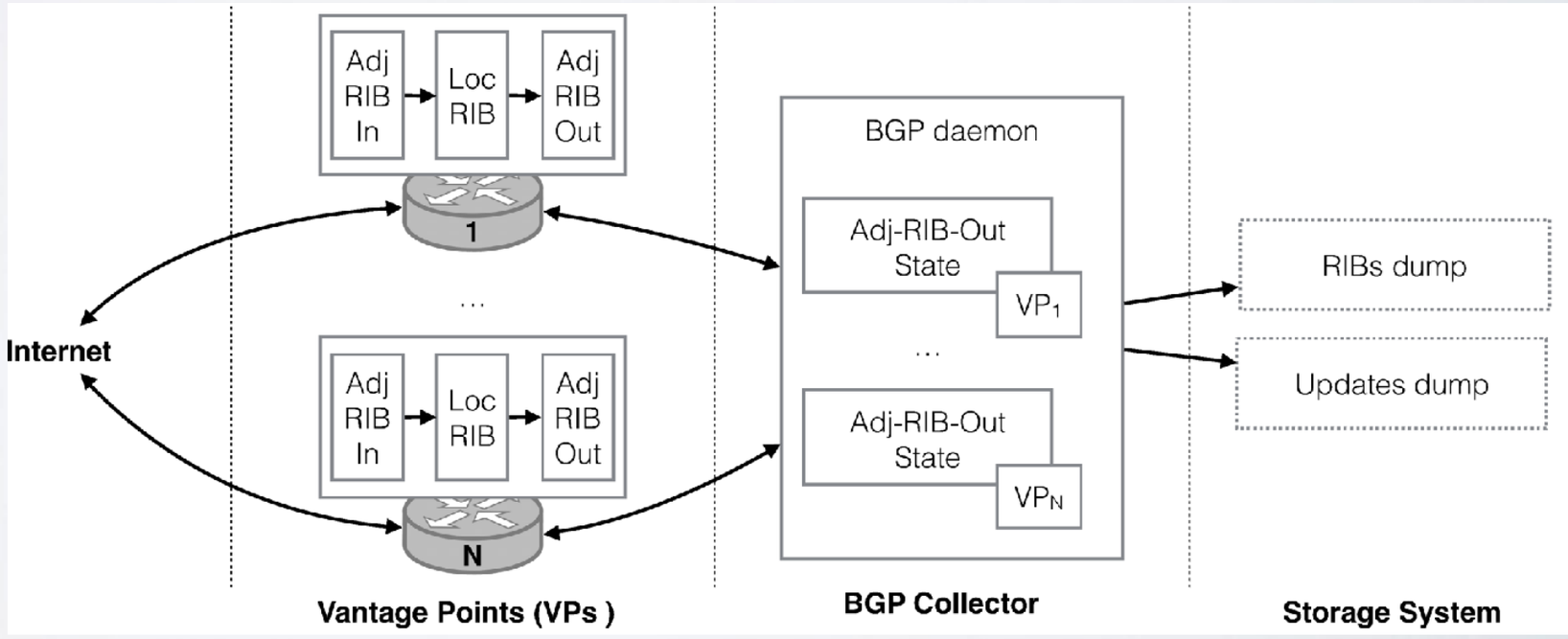


572 ASNs with 2,951 VPs in 77 countries

DATA COLLECTION

Collectors

- Route Collector
  - Establishes BGP peering sessions with one or more real routers (monitors/VPs)
  - Each VP sends to the collector update messages (updates) each time the Adj-RIB-out changes, reflecting changes to its Loc-RIB
- Dumps
  - For each VP, the collector maintains a session state and an image of the Adj-RIB-out table derived from updates. The collector periodically dumps:
  - ***RIB dumps***: a snapshot of the union of the maintained Adj-RIB-out tables (every few hrs)
  - ***Updates dumps***: the update messages received from all its VPs since the last dump, along with state changes



DATA COLLECTION

Public Collectors Projects

- An impressive coverage of the Internet topology!
  - typically data is archived (http/ftp access) in MRT format

- RouteViews
  - ~370 monitors
- RIPE RIS
  - ~500 monitors
  - A few monitors streaming live (web socket, json format)
- Packet Clearing House
- Colorado State BGPmon
  - Streaming live xml format

DATA COLLECTION

Monitors: Full vs Partial Feed

- What does a monitor share
  - Normally, a BGP session with a collector is configured as if the VP was offering transit service to the collector. ***full-feed***
    - This way, the collector potentially knows, at each instant, all the preferred-routes that the VP will use to reach the rest of the Internet – *note this is a partial view of the Internet topology graph visible to that router.*

- A ***partial-feed*** VP instead, will provide through its Adj-RIB-Out only a subset of the routes in its Loc-RIB, e.g., routes to its own networks, or learned through its customers.

DATA COLLECTION

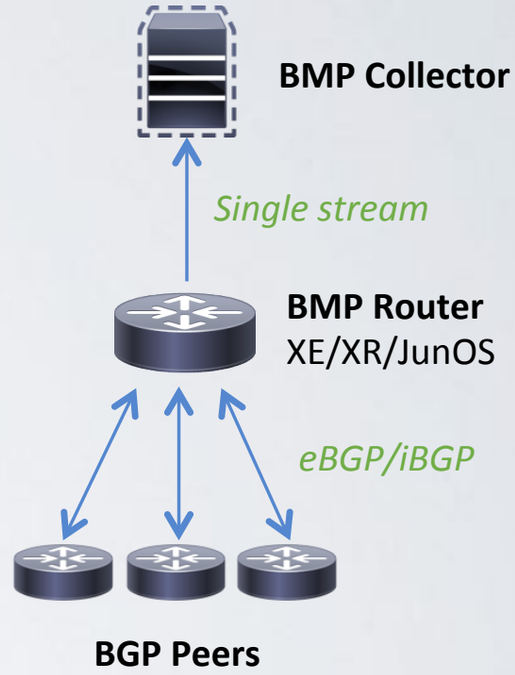
BGP Monitoring Protocol (BMP) - RFC 7854

- 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



credit: Serpil Bayraktar, Cisco

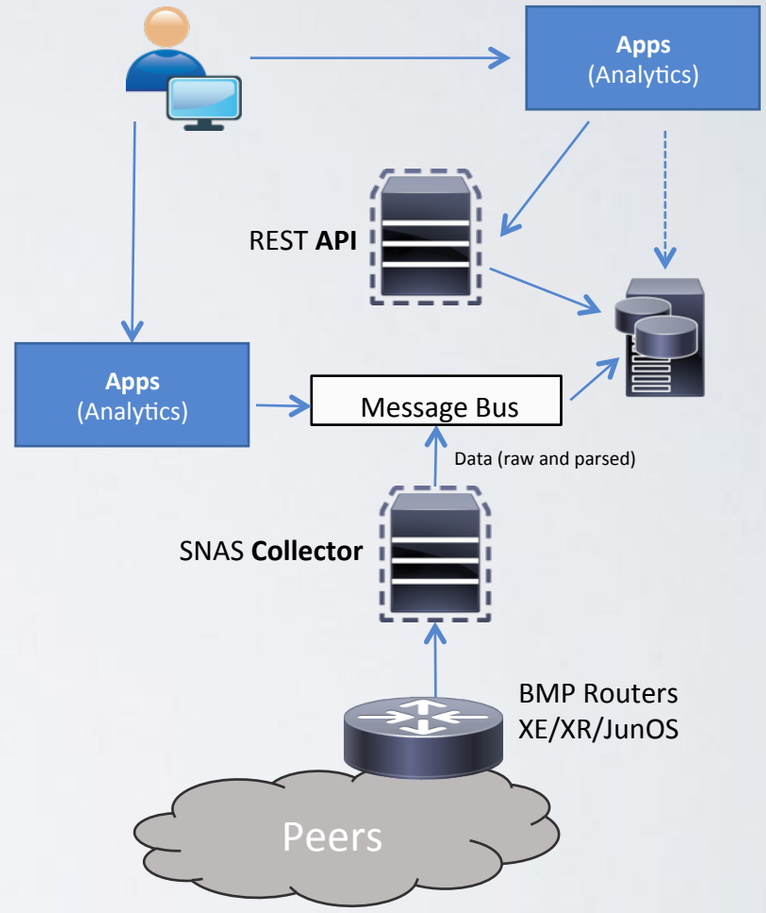
DATA COLLECTION

OpenBMP/SNAS

- 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.

www.openbmp.org



credit: Serpil Bayraktar, Cisco

MEASURING BGP

Data Injection

- **PEERING Testbed**
- **ExaBGP**

DATA INJECTION

ExaBGP

- ExaBGP - a "BGP swiss army knife"
  - An application providing an easy way to interact with BGP networks
  - The program is designed to allow the injection of arbitrary routes into a network, including IPv6 and FlowSpec.

https://github.com/Exa-Networks/exabgp/wiki

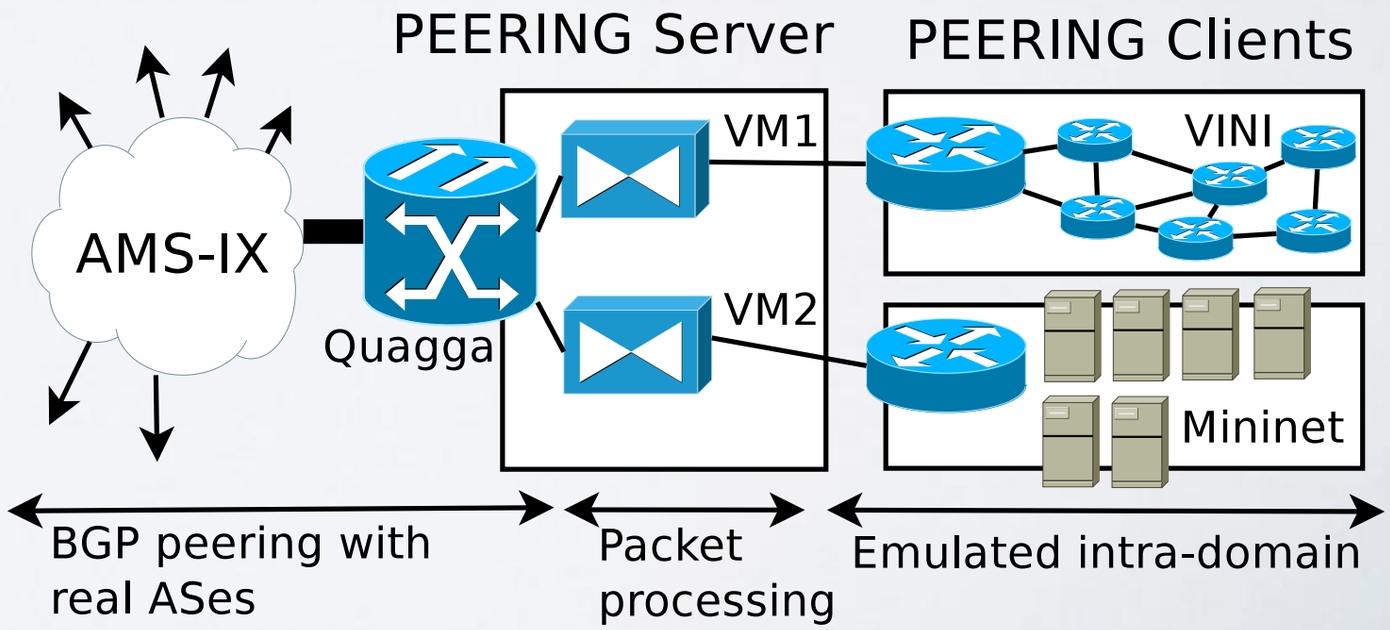
DATA INJECTION

PEERING Testbed

- Inject/Receive Routes & Traffic
  - The testbed can multiplex multiple simultaneous research experiments, each of which independently makes routing decisions and sends and receives traffic.
  - Peering at multiple locations, including major IXPs
- Made of two components
  - Transit Portal: BGP multiplexing service and autonomous system (AS 47065)
  - Extended version of Mininet (MiniNExT) to emulate a complex network topology

https://peering.usc.edu

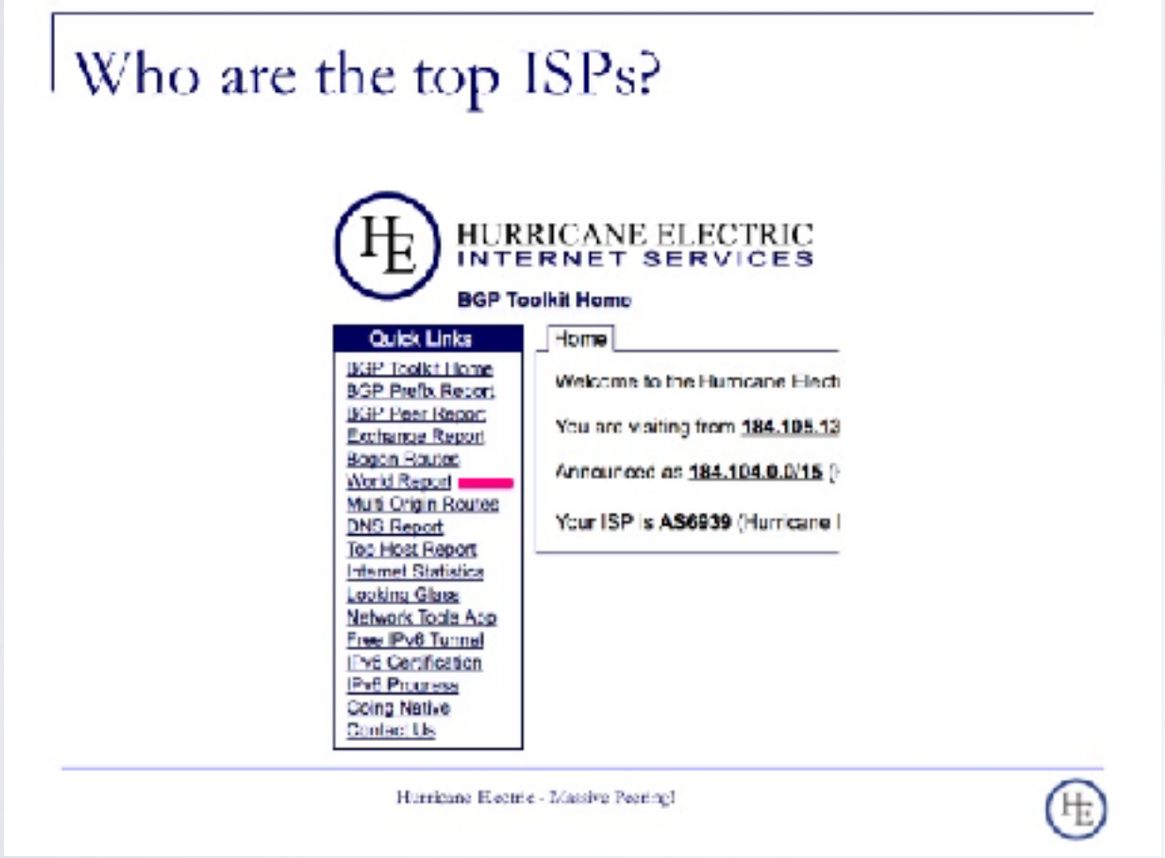
Schlinker et al. "PEERING: An AS for Us", HotNets 2014



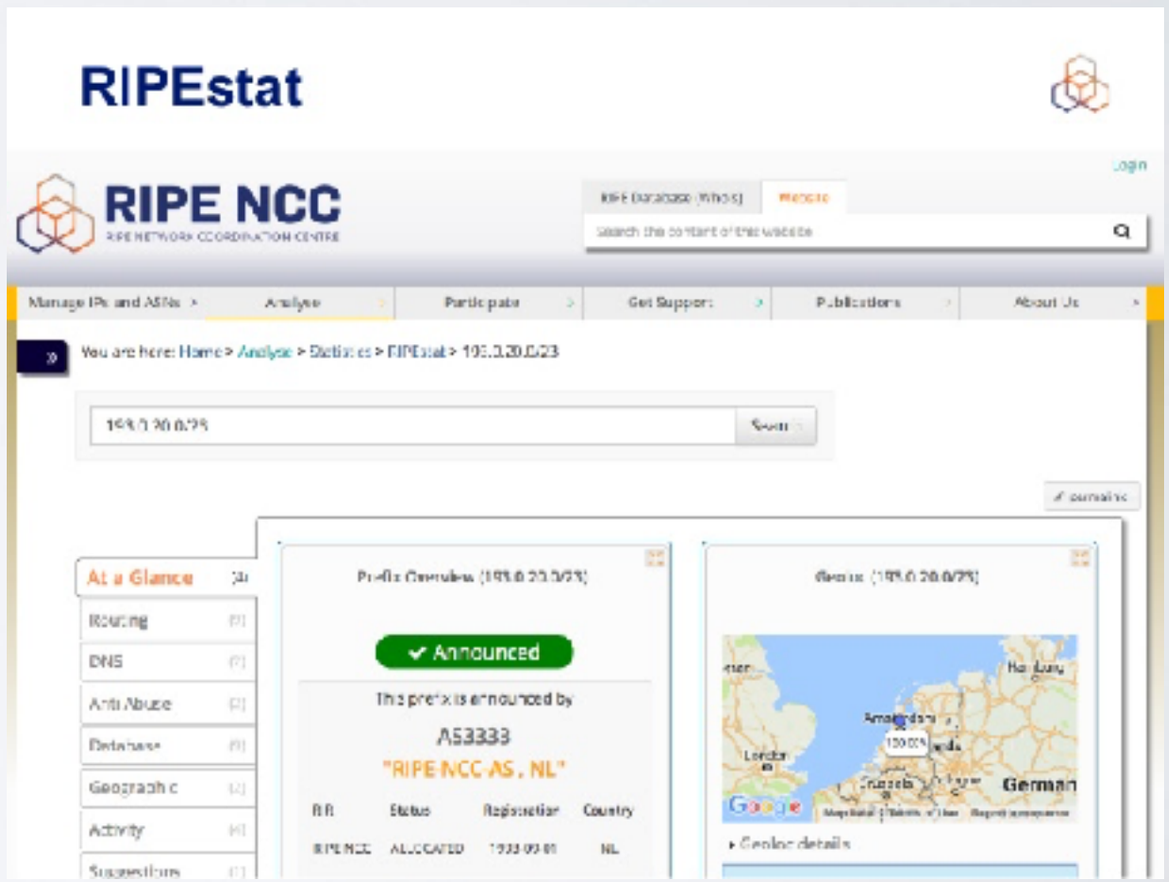
WEB INTERFACES TO DBs

BGP Data, Whois, Routing Registries, …

Hurricane Electric BGP Toolkit
http://bgp.he.net



RIPEstat
https://stat.ripe.net



MEASURING BGP

Processing & Analysis

1. Of course we need more/better data
   - more info from the protocol/routers, more collectors, more experimental testbeds, …

2. But we also **need better tools to learn from the data**
   - to make data analysis: *easier, faster, able to cope with BIG and heterogeneous data*
   - to monitor BGP in near-realtime
   - tightening data collection, processing, visualization, …

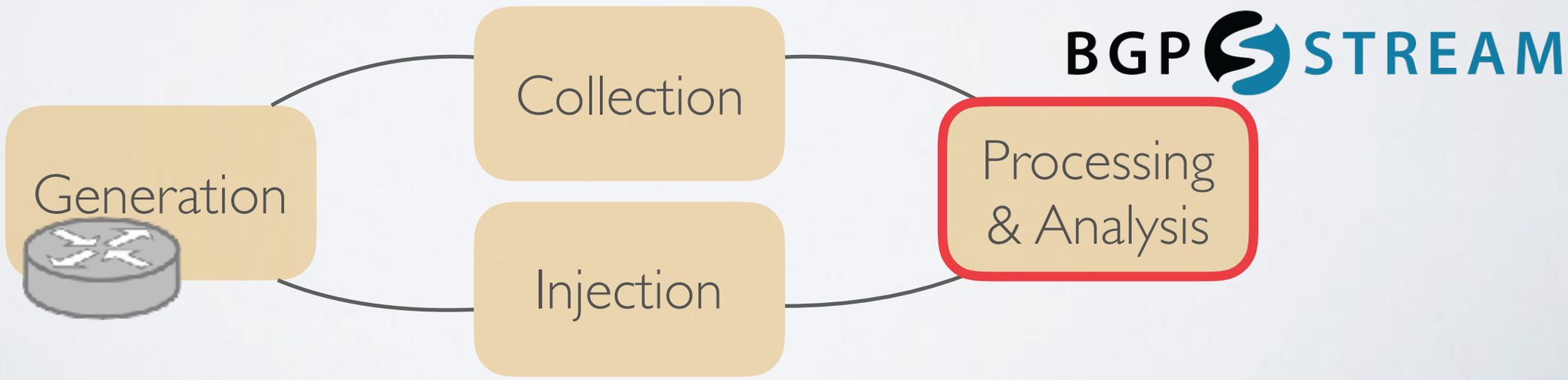
- libBGPdump
https://bitbucket.org/ripencc/bgpdump

MEASURING BGP

two issues - somehow related

1. Of course we need more/better data
   - more info from the protocol/routers, more collectors, more experimental testbeds, …

2. But we also **need better tools to learn from the data**
   - to make data analysis: *easier, faster, able to cope with BIG and heterogeneous data*
   - to monitor BGP in near-realtime
   - tightening data collection, processing, visualization, …



BGPSTREAM

overview

- A software framework for **historical** and **live** BGP data analysis
- 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
  - Simple API
  - Facilitates reproducibility and repeatability


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

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

ABSTRACT

1. INTRODUCTION

2. BACKGROUND

Orsini et al. "BGPStream: a software framework for live and historical BGP data analysis", IMC 2016

BGPSTREAM

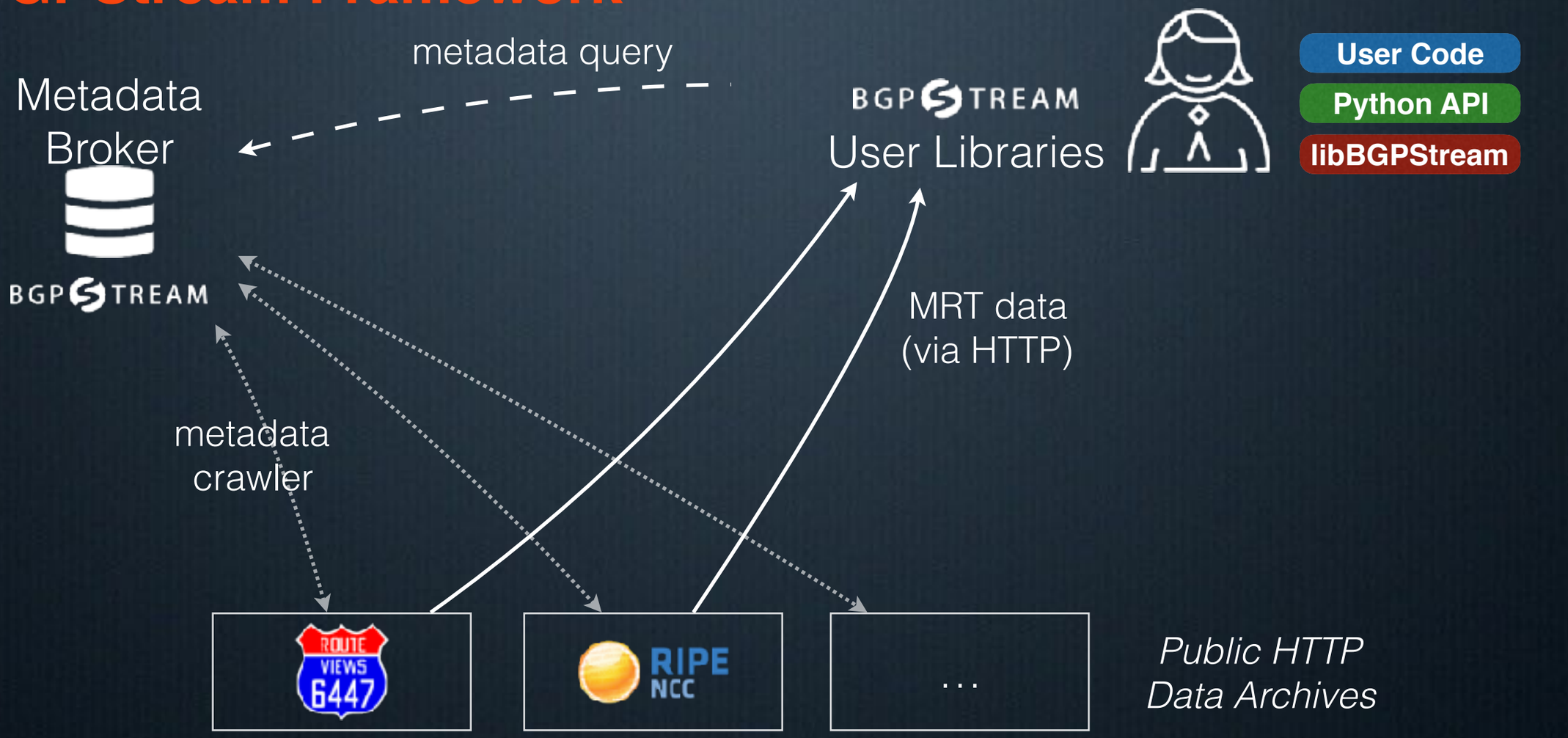
it's real!

- ***bgpstream.caida.org***
  - download it! (version 1.1)
  - active development - *github.com/caida/bgpstream*
  - Docs & Tutorials
- lots of people are using it!
- coordination with RouteViews, Colorado State BGPMon, RIPE NCC
- BGP Hackathon February 2016, NANOG Hackathon in June, …
- Collaboration with Cisco to natively support BMP
- V2 coming soon!

State of the Art?

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

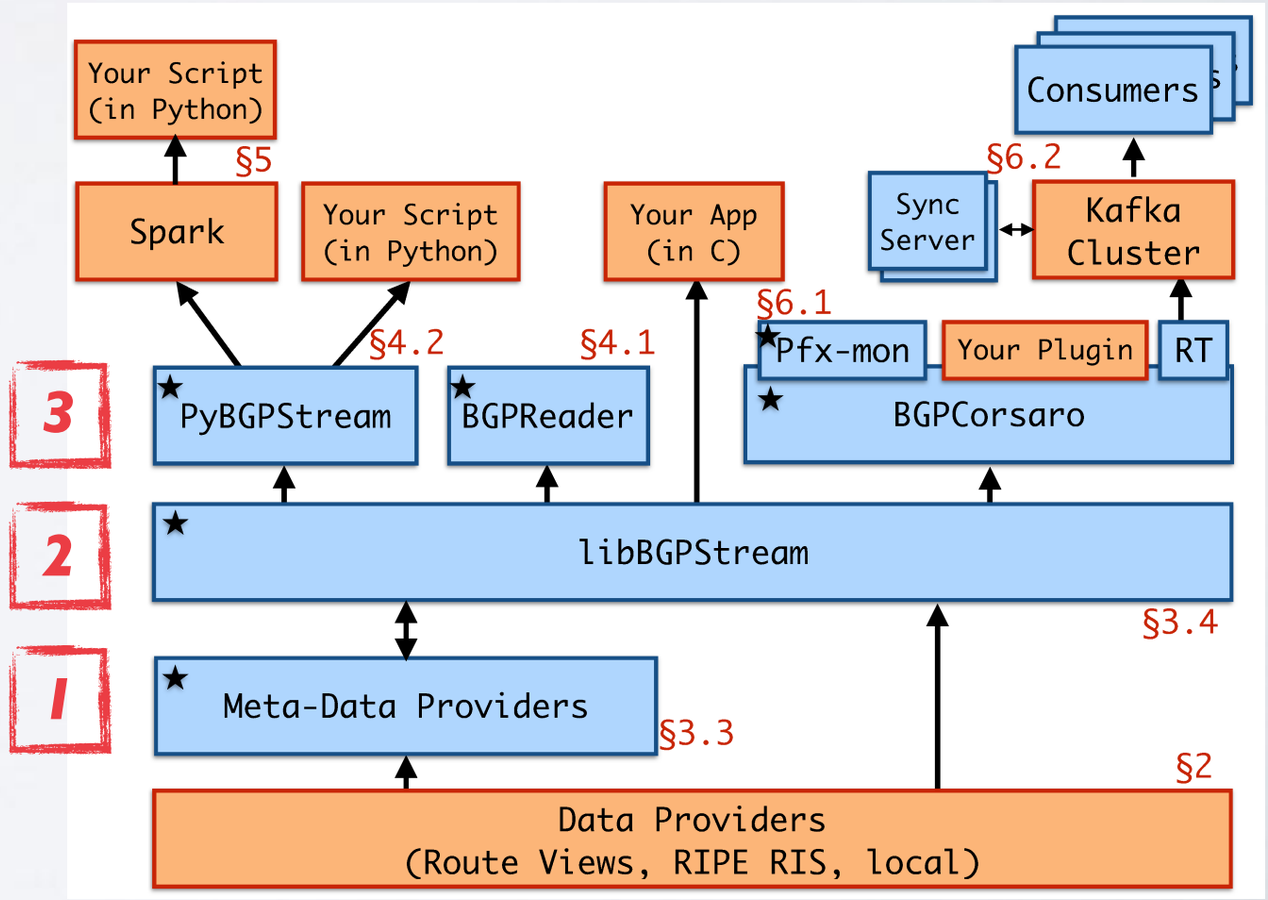
The BGPStream Framework



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*



C API

specifying a stream

BGP STREAM

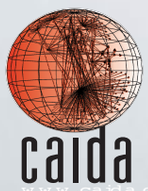
```
int main(int argc, const char **argv)                                    1
{                                                                        2
  bgpstream_t *bs = bgpstream_create();                                  3
  bgpstream_record_t *record = bgpstream_record_create();                4
  bgpstream_elem_t *elem = NULL;                                         5
  char buffer[1024];                                                     6
                                                                         7
  /* Define the prefix to monitor for (2403:f600::/32) */                8
  bgpstream_pfx_storage_t my_pfx;                                        9
  my_pfx.address.version = BGPSTREAM_ADDR_VERSION_IPV6;                  10
  inet_pton(BGPSTREAM_ADDR_VERSION_IPV6, "2403:f600::", &my_pfx.address.ipv6);  11
  my_pfx.mask_len = 32;                                                  12
                                                                         13
  /* Set metadata filters */                                             14
  bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "rrc00");    15
  bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "route-views2");  16
  bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_RECORD_TYPE, "updates");  17
  /* Time interval: 01:20:10 - 06:32:15 on Tue, 12 Aug 2014 UTC */       18
  bgpstream_add_interval_filter(bs, 1407806410, 1407825135);             19
                                                                         20
  /* Start the stream */                                                 21
  bgpstream_start(bs);                                                   22
                                                                         23
```

LIBBGPSTREAM API

BGPStream record

- **A "BGPStream record" encapsulates an MRT record**
- Dumps are composed of multiple MRT records, whose type is specified in their header
  - -an update message is stored in a single MRT record, but *(see next slide)* update messages related to multiple prefixes can be in the same MRT record

Field	Type	Function
project	string	project name (e.g., Route Views)
collector	string	collector name (e.g., rrc00)
type	enum	RIB or Updates
dump time	long	time the containing dump was begun
position	enum	first, middle, or last record of a dump
time	long	timestamp of the MRT record
status	enum	record validity flag
MRT record	struct	de-serialized MRT record



LIBBGPSTREAM API

BGPStream elem

- **An MRT record may group elements of the same type but related to different VPs or prefixes**
  - *e.g., routes to the same prefix from different VPs (in a RIB dump record)*
  - *e.g., announcements from the same VP to multiple prefixes, but sharing a common path (in a Updates dump record)*
- **libBGPStream decomposes a record into a set of individual elements (*BGPStream elems*)**

Field	Type	Function
type	enum	route from a RIB dump, announcement, withdrawal, or state message
time	long	timestamp of MRT record
peer address	struct	IP address of the VP
peer ASN	long	AS number of the VP
prefix*	struct	IP prefix
next hop*	struct	IP address of the next hop
AS path*	struct	AS path
old state*	enum	FSM state (before the change)
new state*	enum	FSM state (after the change)

* denotes a field conditionally populated based on type

C API

while loop

BGP STREAM

```
/* Start the stream */                                                    21
bgpstream_start(bs);                                                     22
                                                                         23
/* Read the stream of records */                                         24
while (bgpstream_get_next_record(bs, record) > 0) {                      25
  /* Ignore invalid records */                                           26
  if (record->status != BGPSTREAM_RECORD_STATUS_VALID_RECORD) {          27
    continue;                                                            28
  }                                                                      29
  /* Extract elems from the current record */                            30
  while ((elem = bgpstream_record_get_next_elem(record)) != NULL) {      31
    /* Select only announcements and withdrawals, */                     32
    /* and only elems that carry information for 2403:f600::/32 */       33
    if ((elem->type == BGPSTREAM_ELEM_TYPE_ANNOUNCEMENT ||               34
         elem->type == BGPSTREAM_ELEM_TYPE_WITHDRAWAL) &&                35
        bgpstream_pfx_storage_equal(&my_pfx, &elem->prefix)) {           36
      /* Print the BGP information */                                    37
      bgpstream_elem_snprintf(buffer, 1024, elem);                       38
      fprintf(stdout, "%s\n", buffer);                                   39
    }                                                                    40
  }                                                                      41
}                                                                        42
                                                                         43
```

BGPREADER

BGP STREAM

command-line tool for ASCII output w/ filters

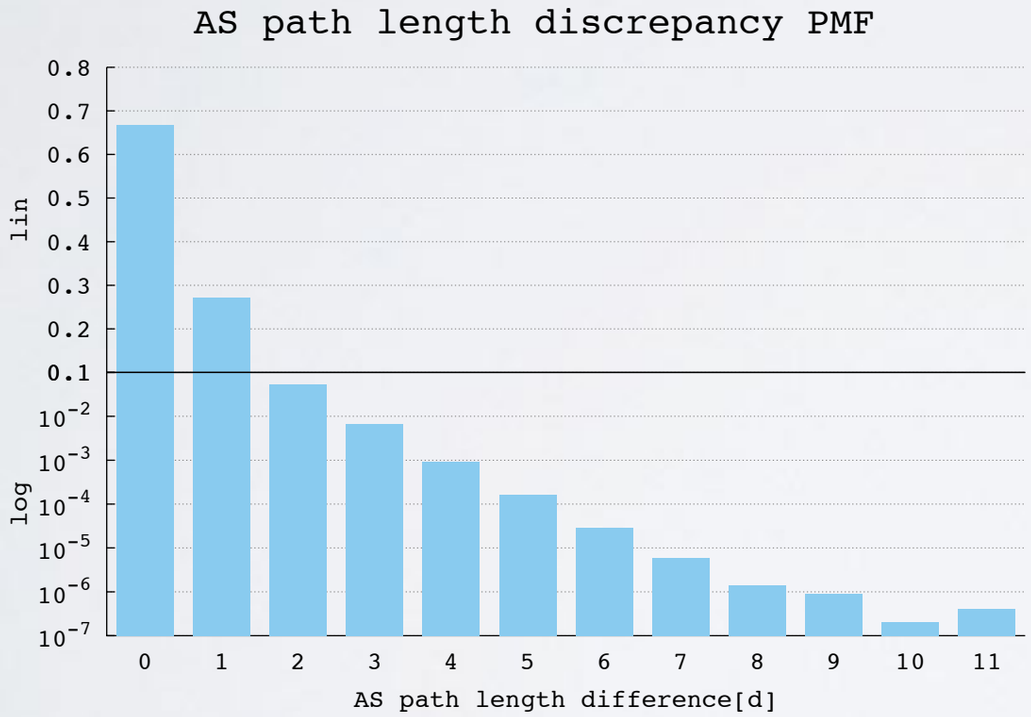
```
$ bgpreader -w 1445306400,1445306402 -c route-views.sfmix
R|B|1445306400|routeviews|route-views.sfmix
R|R|1445306400|routeviews|route-views.sfmix|32354|206.197.187.5|1.0.0.0/24|206.197.187.5|32354 15169|15169|||
...
R|R|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|2c0f:ffd8::/32|
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||
...
```

PYBGPSTREAM

BGP STREAM

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)




```python
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

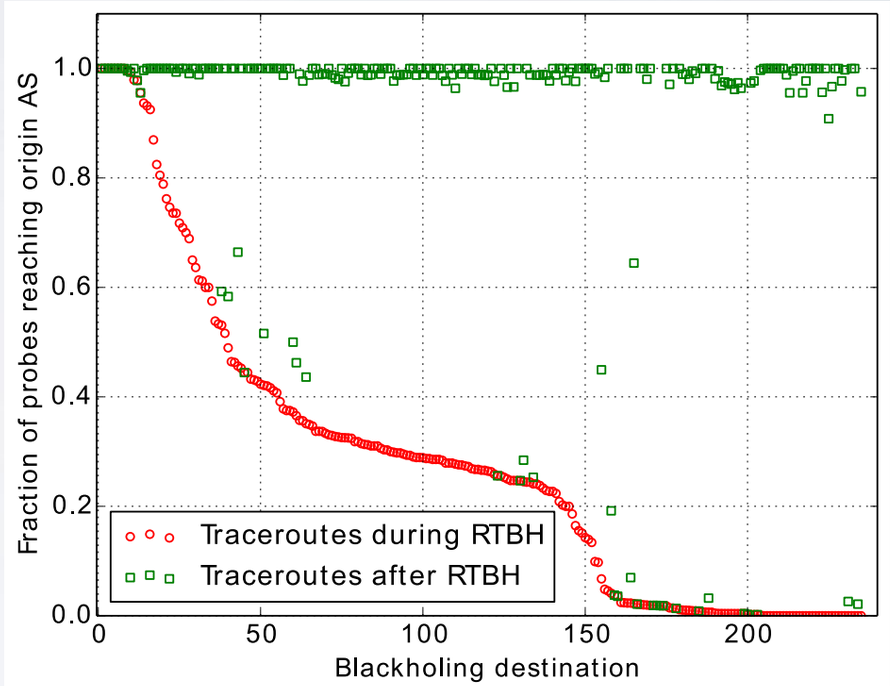
Center for Applied Internet Data Analysis
University of California San Diego

PYBGPSTREAM

BGP STREAM

Example: timely combine with active measurements

- We monitor **community-based black-holing**
  - Victim of DoS attack announces prefix with special community attribute to request 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***



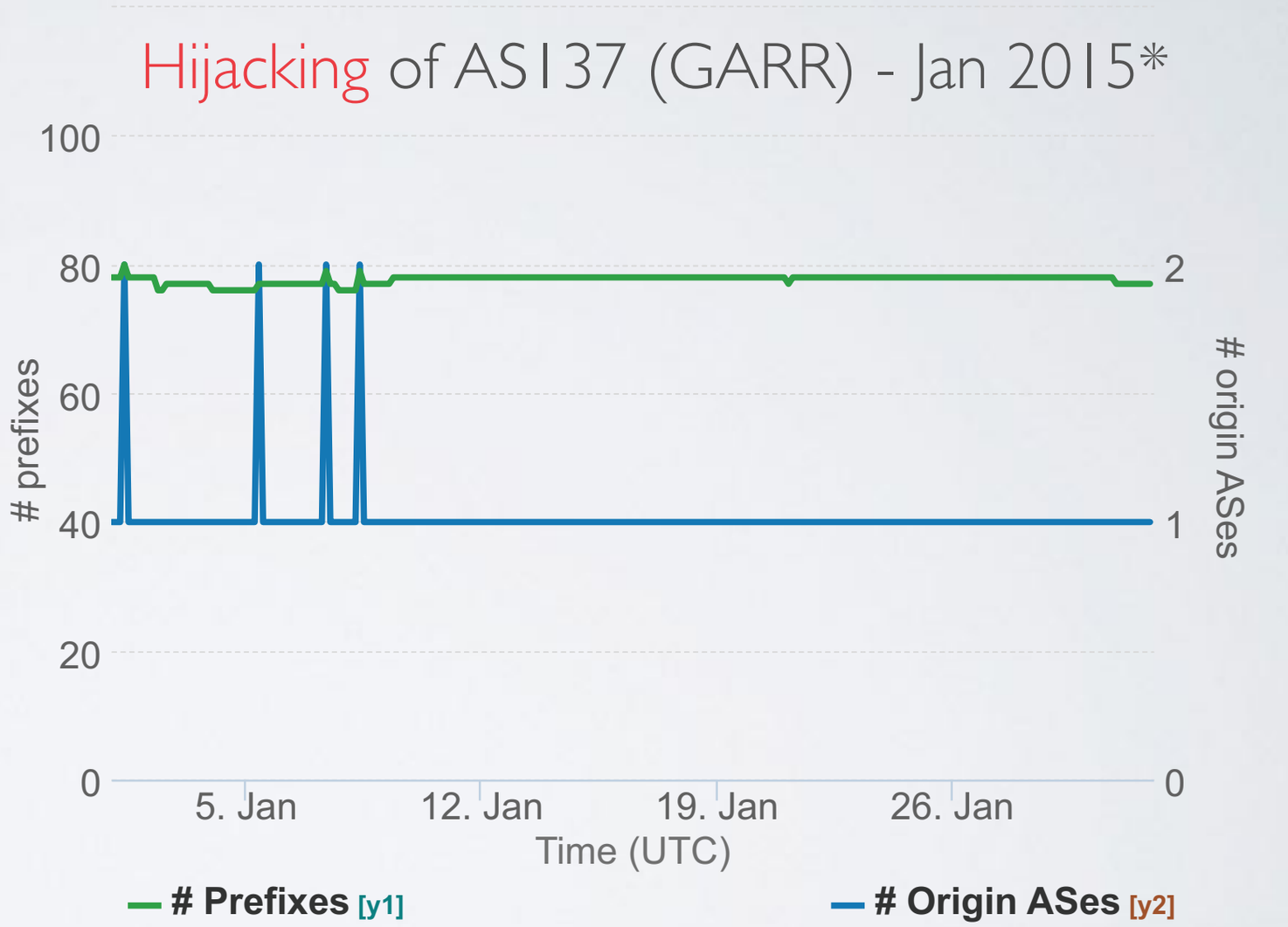
BGPCORSARO

BGPSTREAM

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*



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

NO MANUAL DOWNLOADS

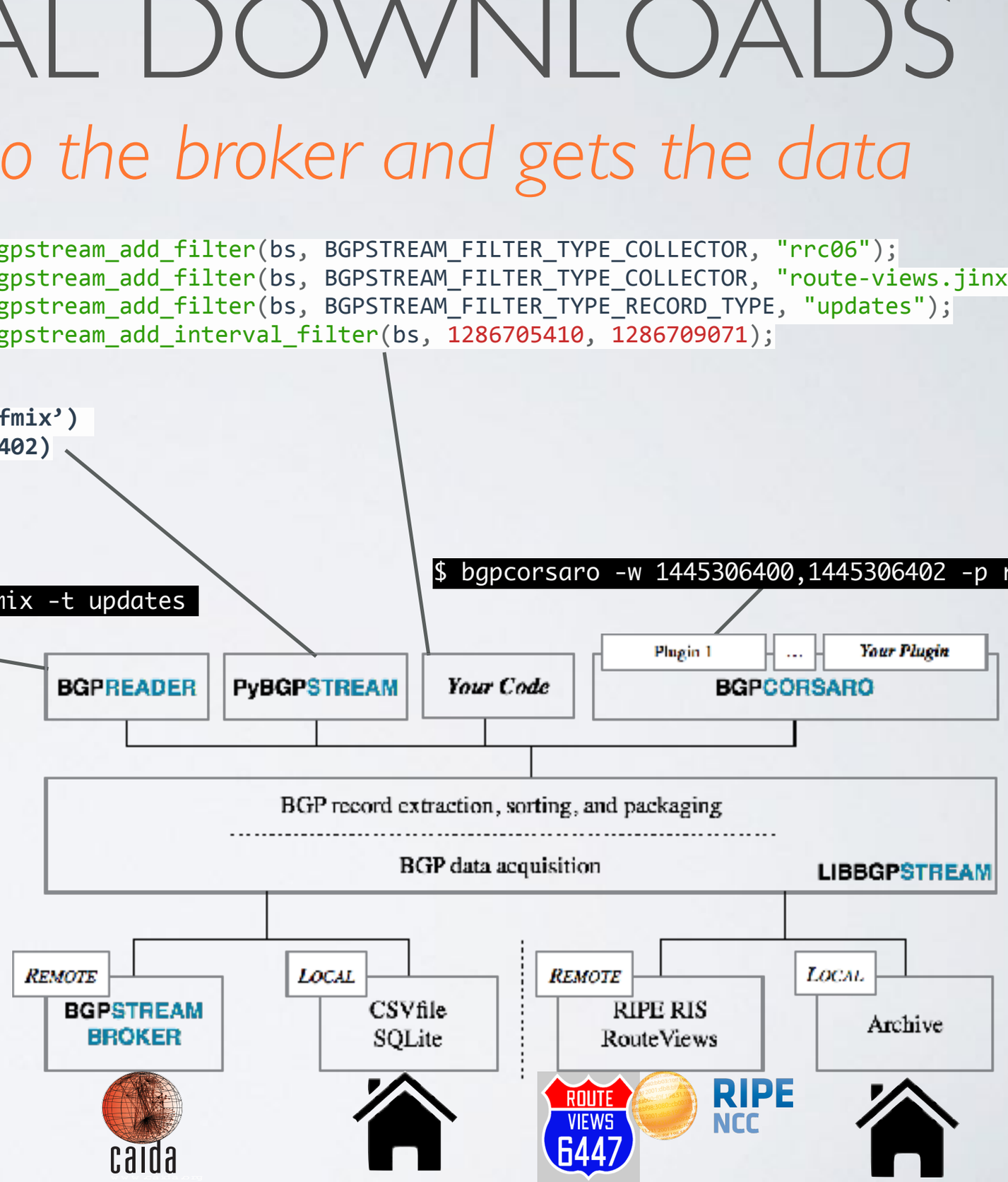
libBGPStream talks to the broker and gets the data

```
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
```



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

GET A LIVE STREAM

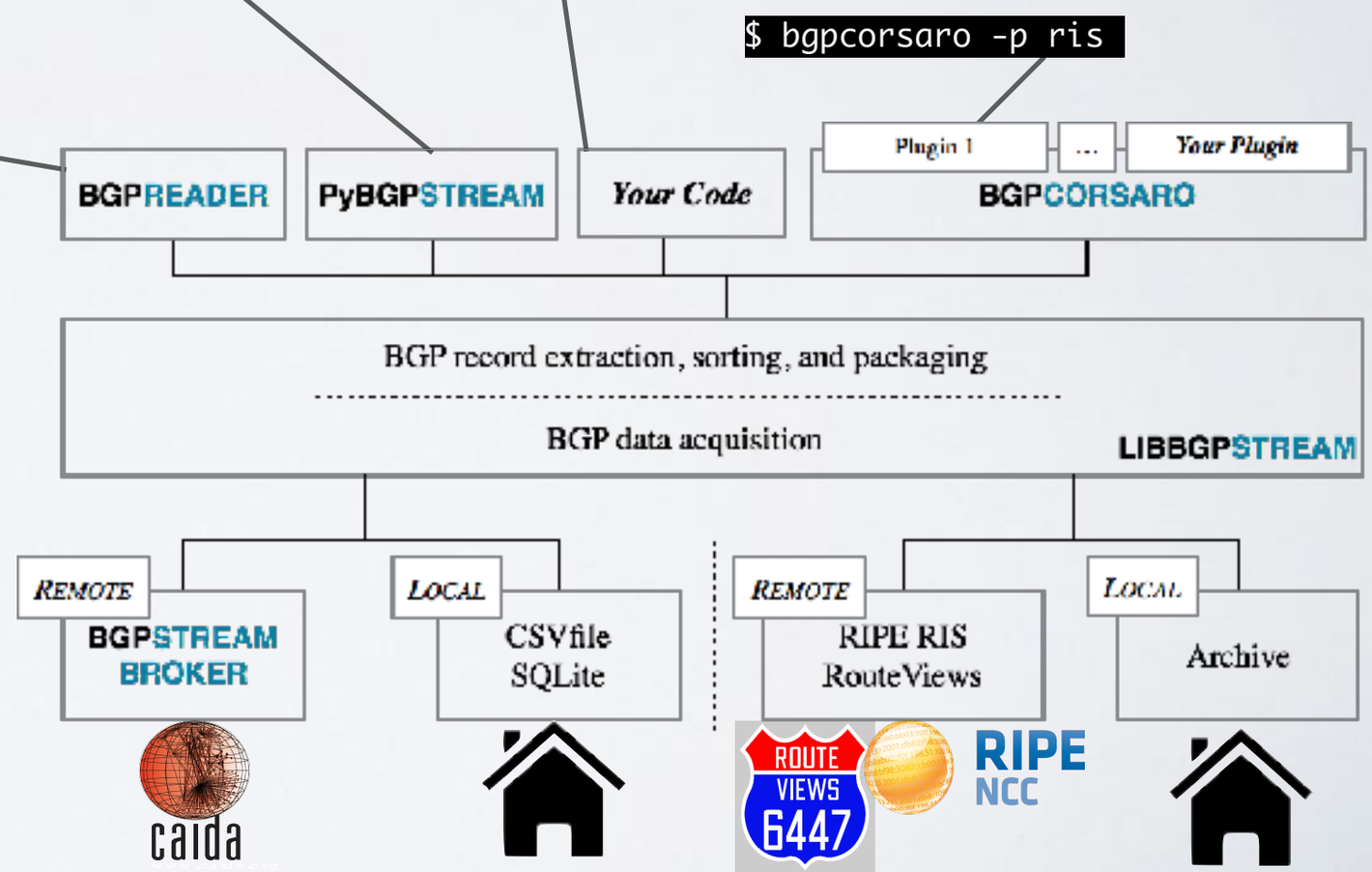
libBGPStream keeps retrieving data as it becomes available

```
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
```

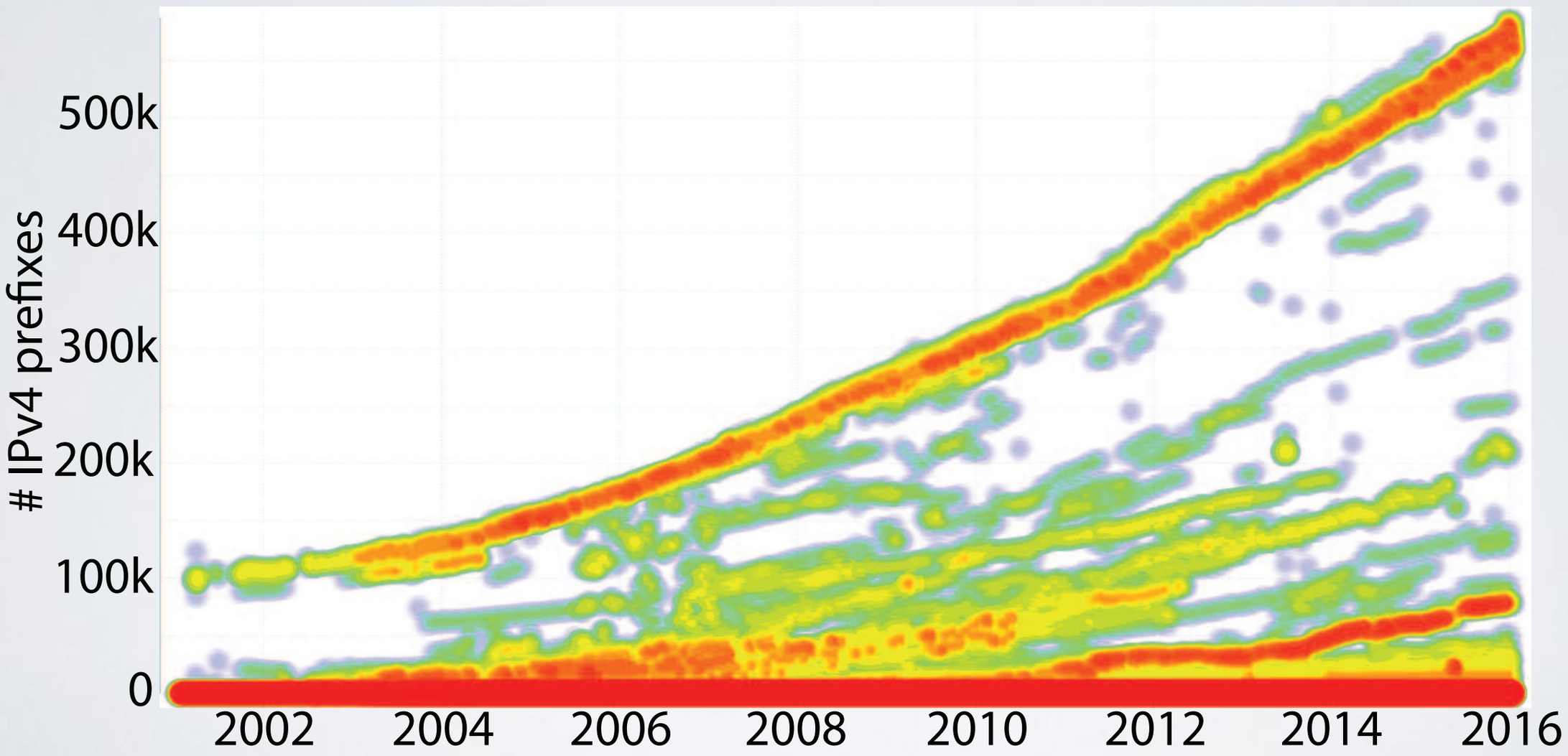
```
$ bgpcorsaro -p ris
```



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

CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

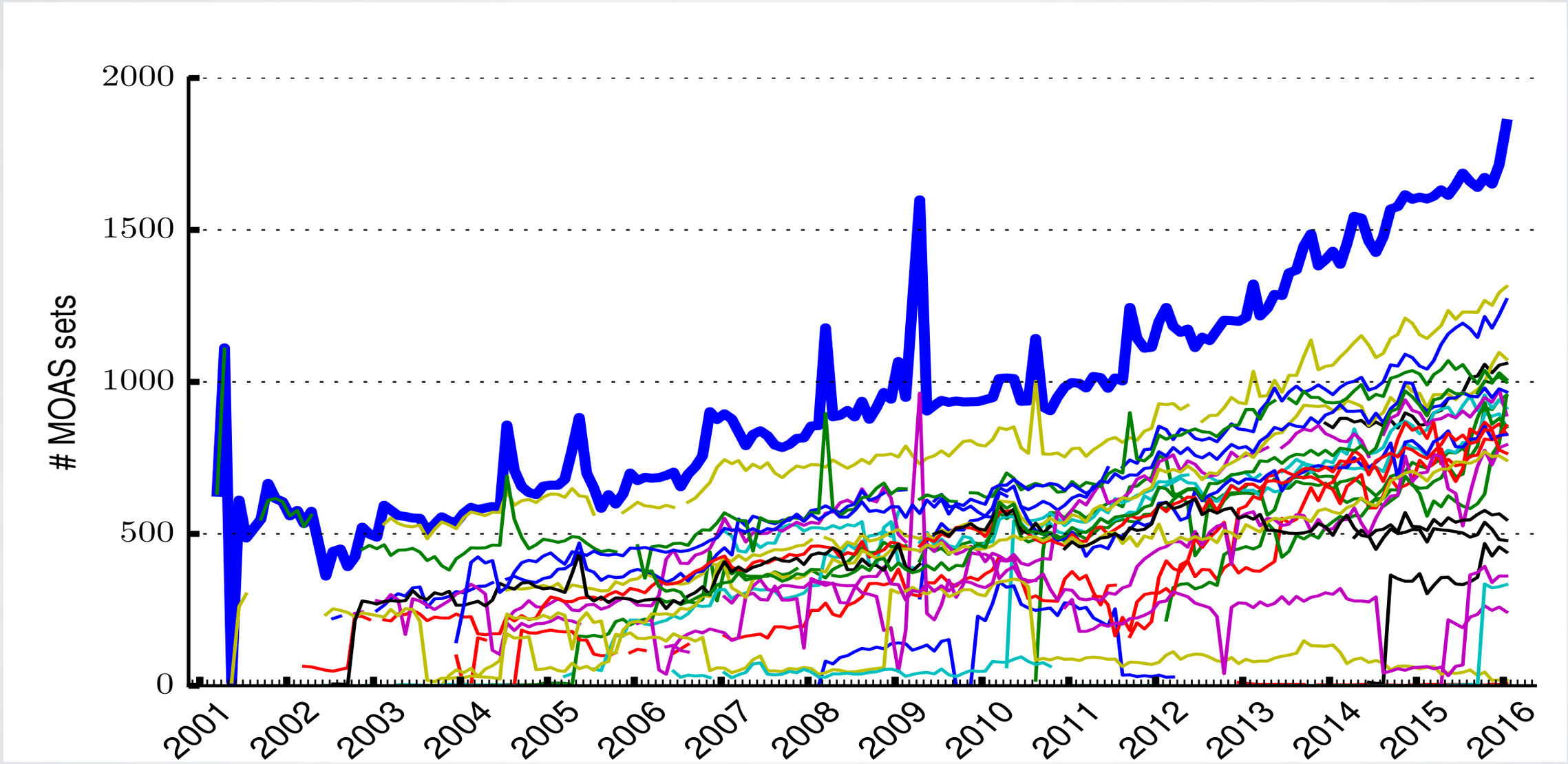


Center for Applied Internet Data Analysis
University of California San Diego

Code at www.caida.org/publications/papers/2016/bgpstream/supplemental

CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

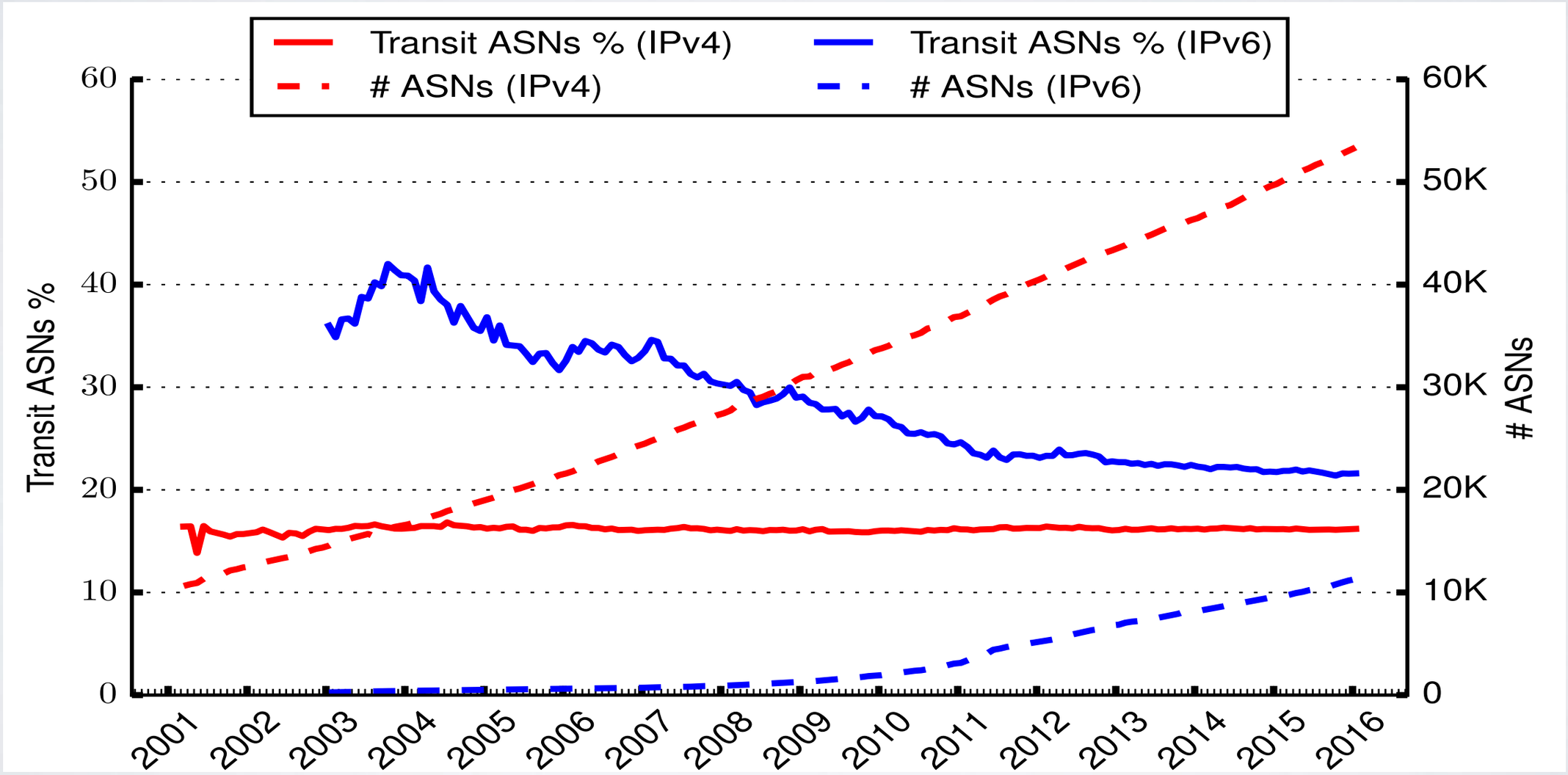


Center for Applied Internet Data Analysis
University of California San Diego

Code at www.caida.org/publications/papers/2016/bgpstream/supplemental

CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream



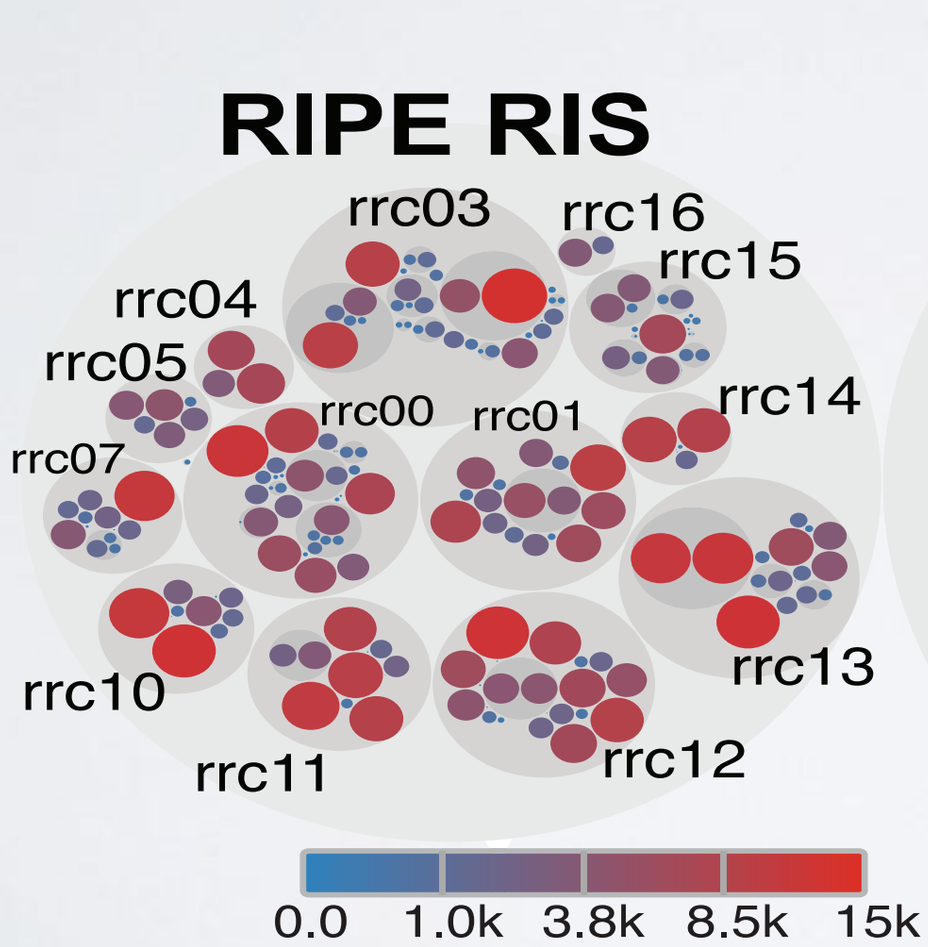
Code at www.caida.org/publications/papers/2016/bgpstream/supplemental

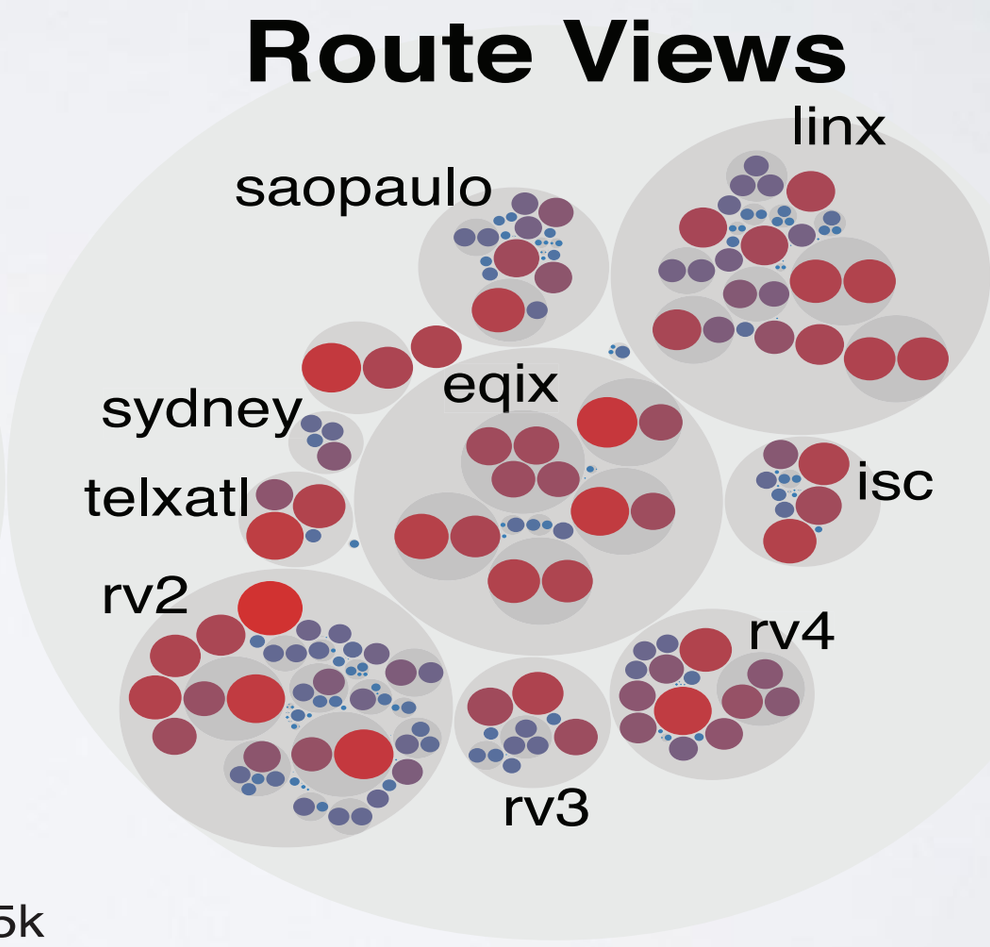
Center for Applied Internet Data Analysis
University of California San Diego

CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

BGP





INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

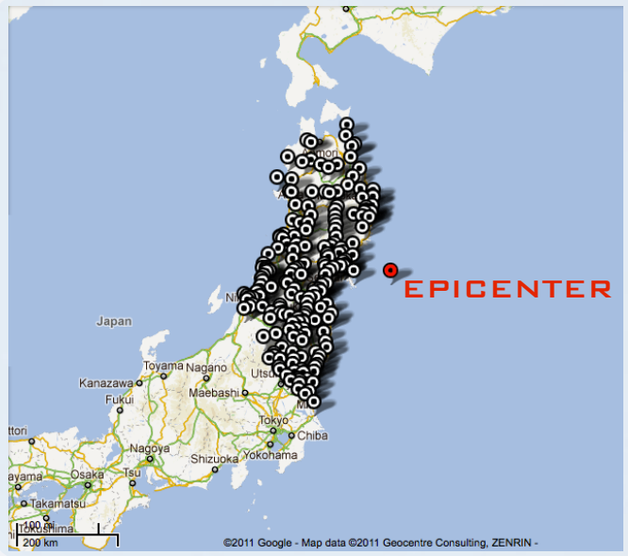
- Country-level Internet Blackouts during the Arab Spring

Dainotti et al. "Analysis of Country-wide Internet Outages Caused by Censorship" IMC 2011

Egypt, Jan 2011
Government orders to shut down the Internet

- Natural disasters affecting the infrastructure

Dainotti et al. "Extracting Benefit from Harm: Using Malware Pollution to Analyze the Impact of Political and Geophysical Events on the Internet" SIGCOMM CCR 2012

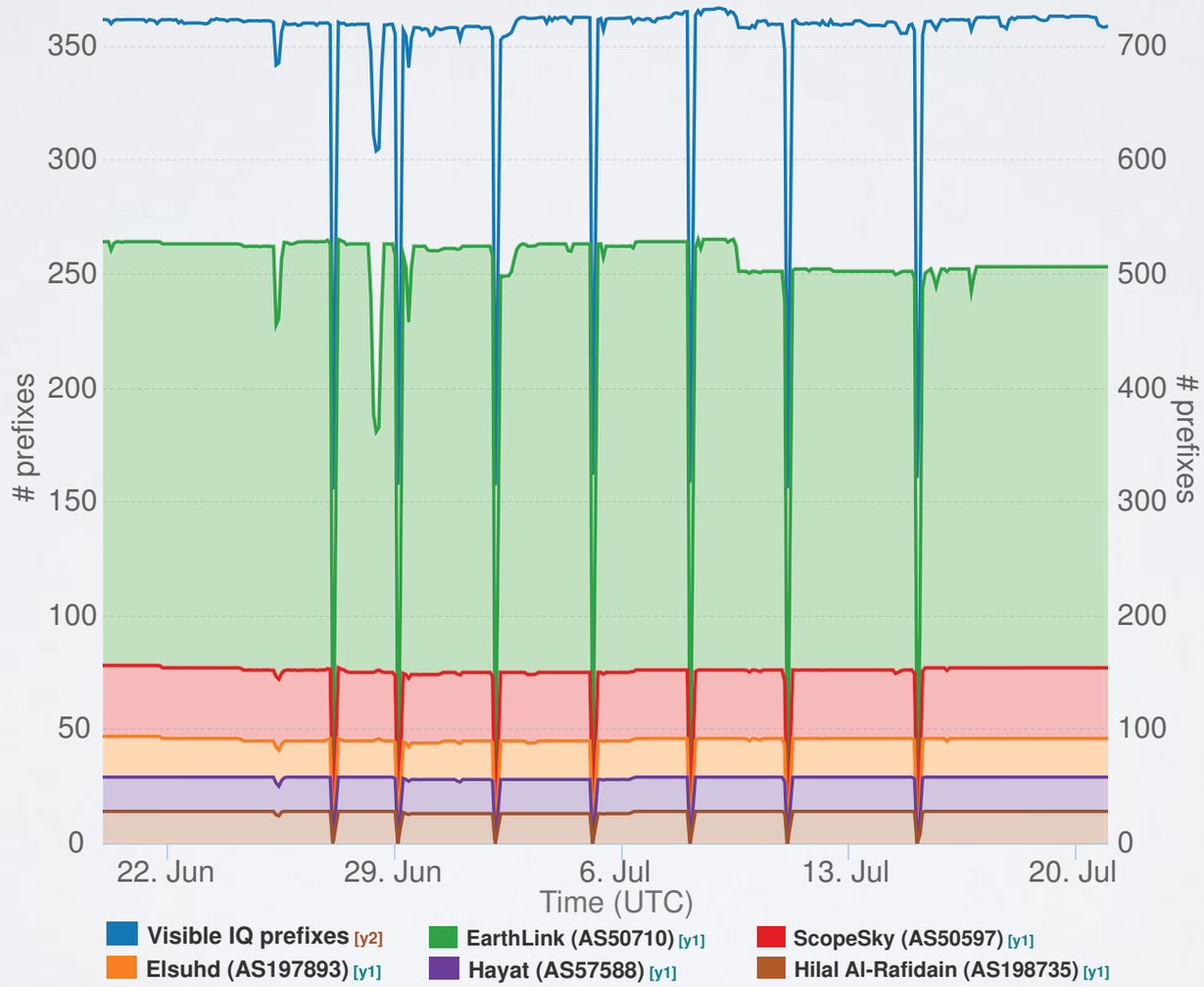


Japan, Mar 2011
Earthquake of Magnitude 9.0

INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

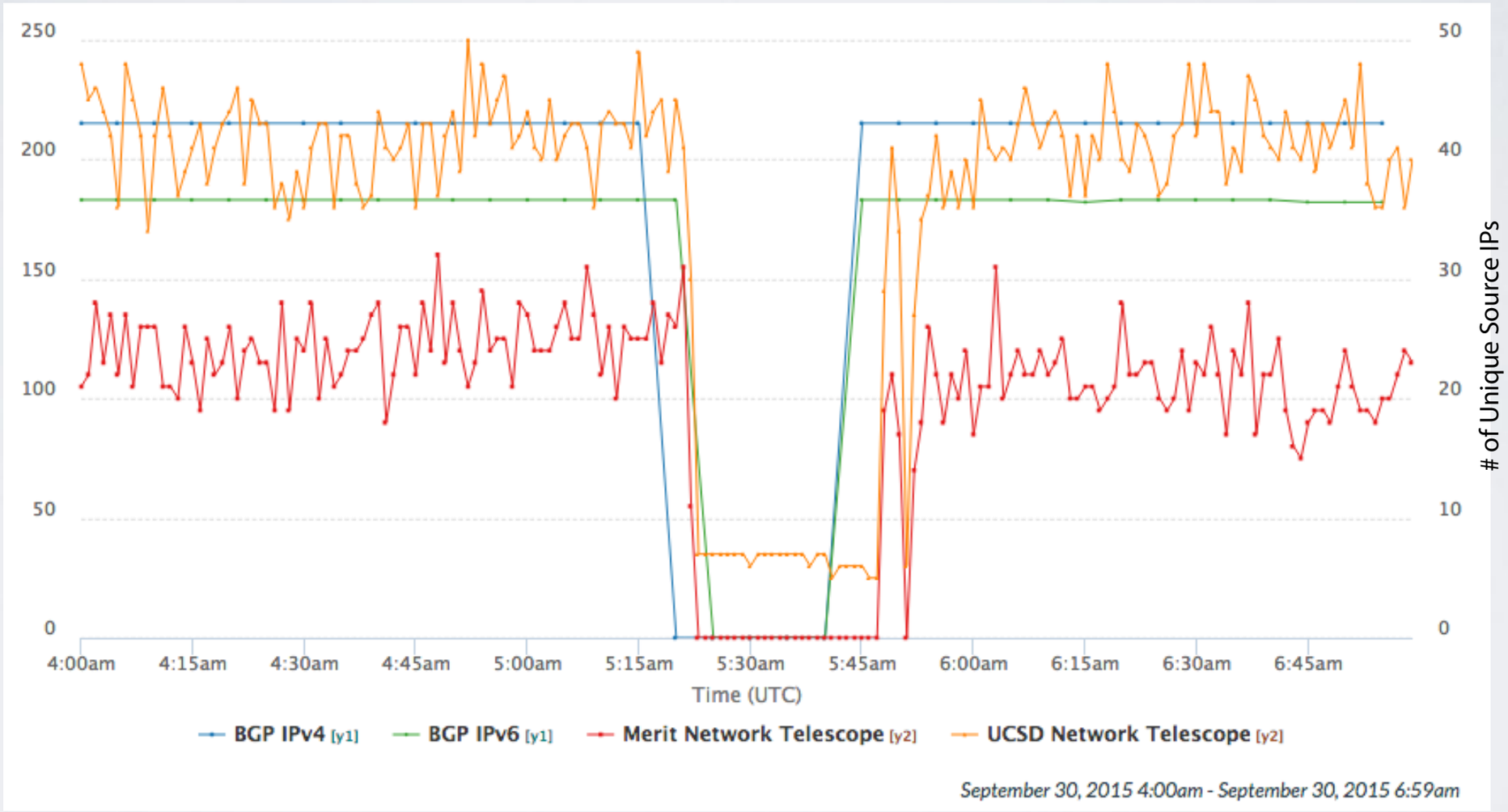
Country-wide Internet outages in Iraq that the government ordered in conjunction with the ministerial preparatory exams - Jul 2015



INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

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

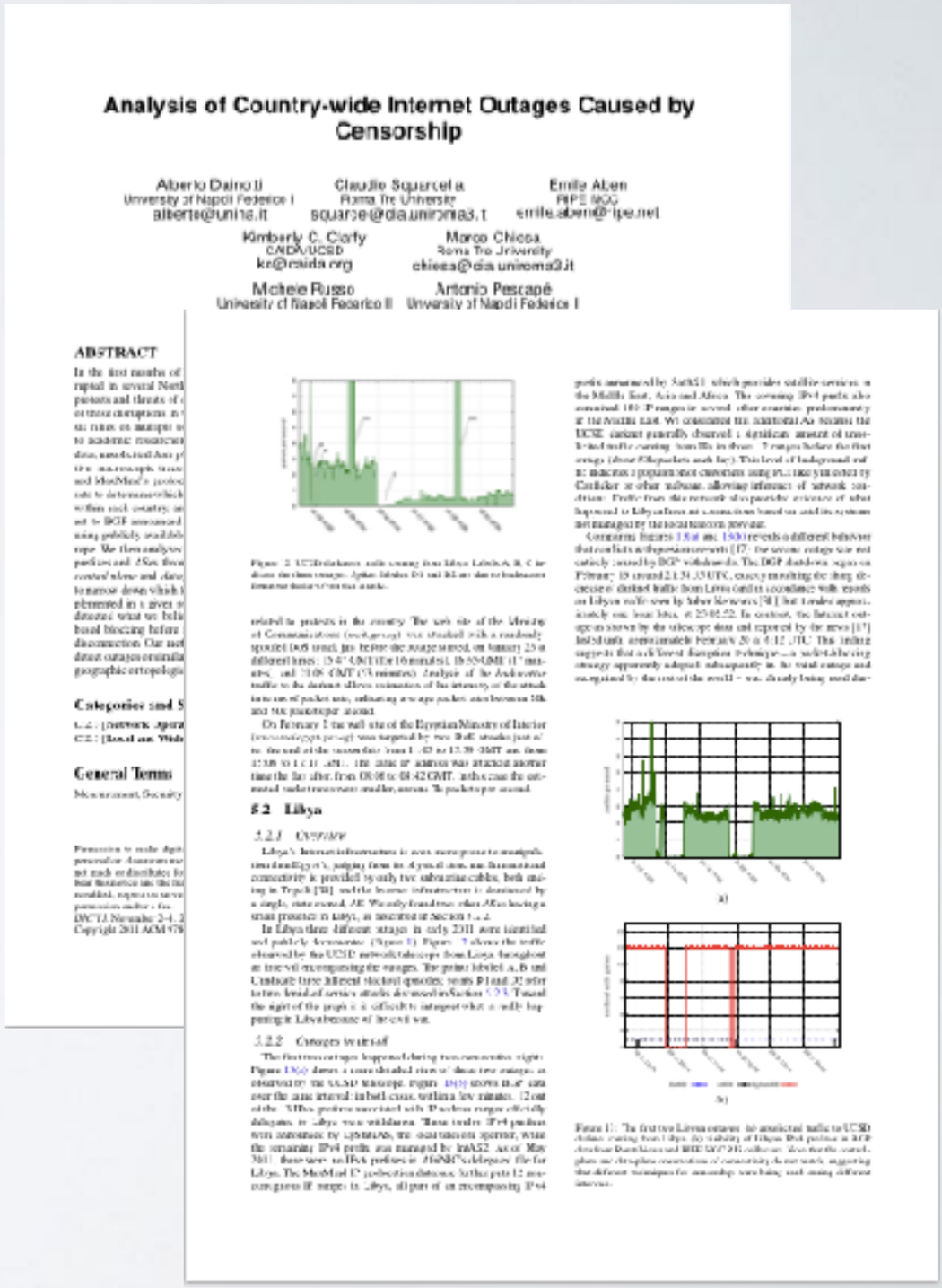


BEFORE IODA

post-event manual analysis

Egypt, Jan 2011
Government orders to shut down the Internet

4 months of work



Analysis of Country-wide Internet Outages Caused by Censorship

Alberto Dainotti
University of Napoli Federico II
alberto@unina.it

Claudio Squarcella
Roma Tre University
squarcel@dia.uniroma3.it

Emile Aben
RIPE NCC
emile.aben@ripe.net

Kimberly C. Claffy
CAIDA/UCSD
kc@caida.org

Marco Chiesa
Roma Tre University
chiesa@dia.uniroma3.it

Michele Russo
University of Napoli Federico II

Antonio Pescapé
University of Napoli Federico II

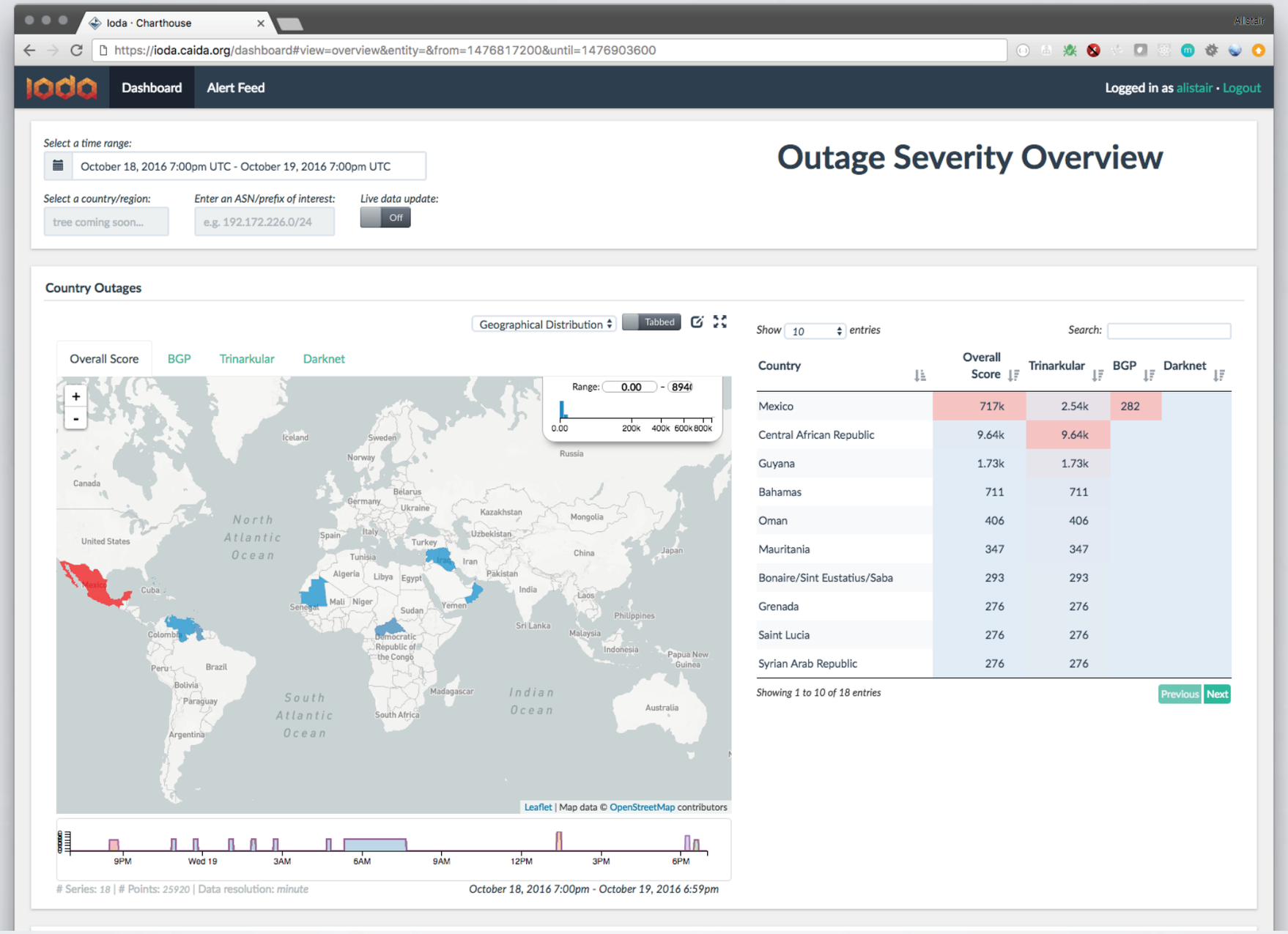
ABSTRACT

Dainotti et al. "Analysis of Country-wide Internet Outages Caused by Censorship" IMC 2011

Center for Applied Internet Data Analysis
University of California San Diego

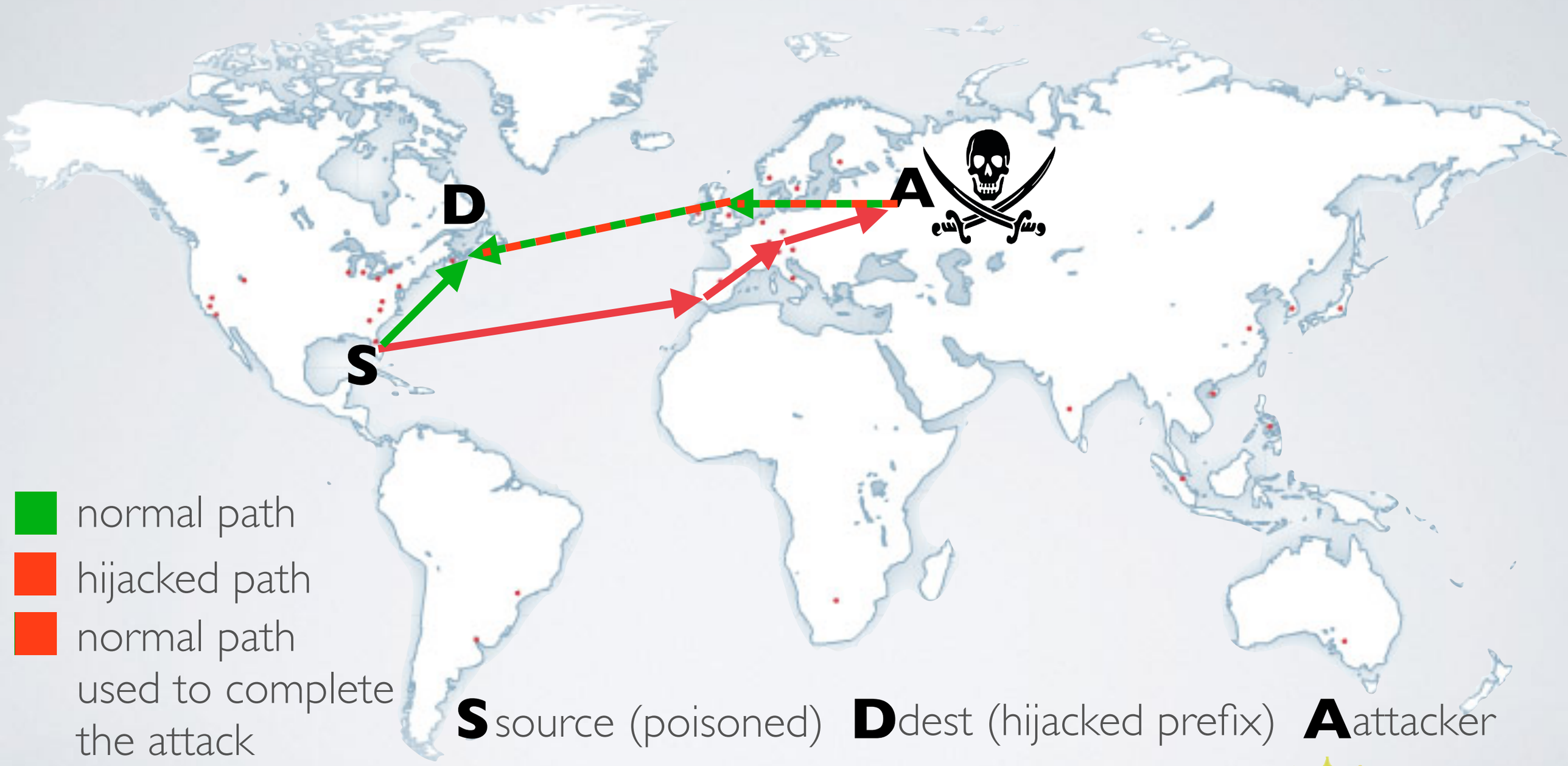
IODA AFTER 4 YEARS (TODAY)

live detection and monitoring



INSPIRING PROJECTS (2/2)

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

caida

Stony Brook University

www.caida.org/funding/hijacks/

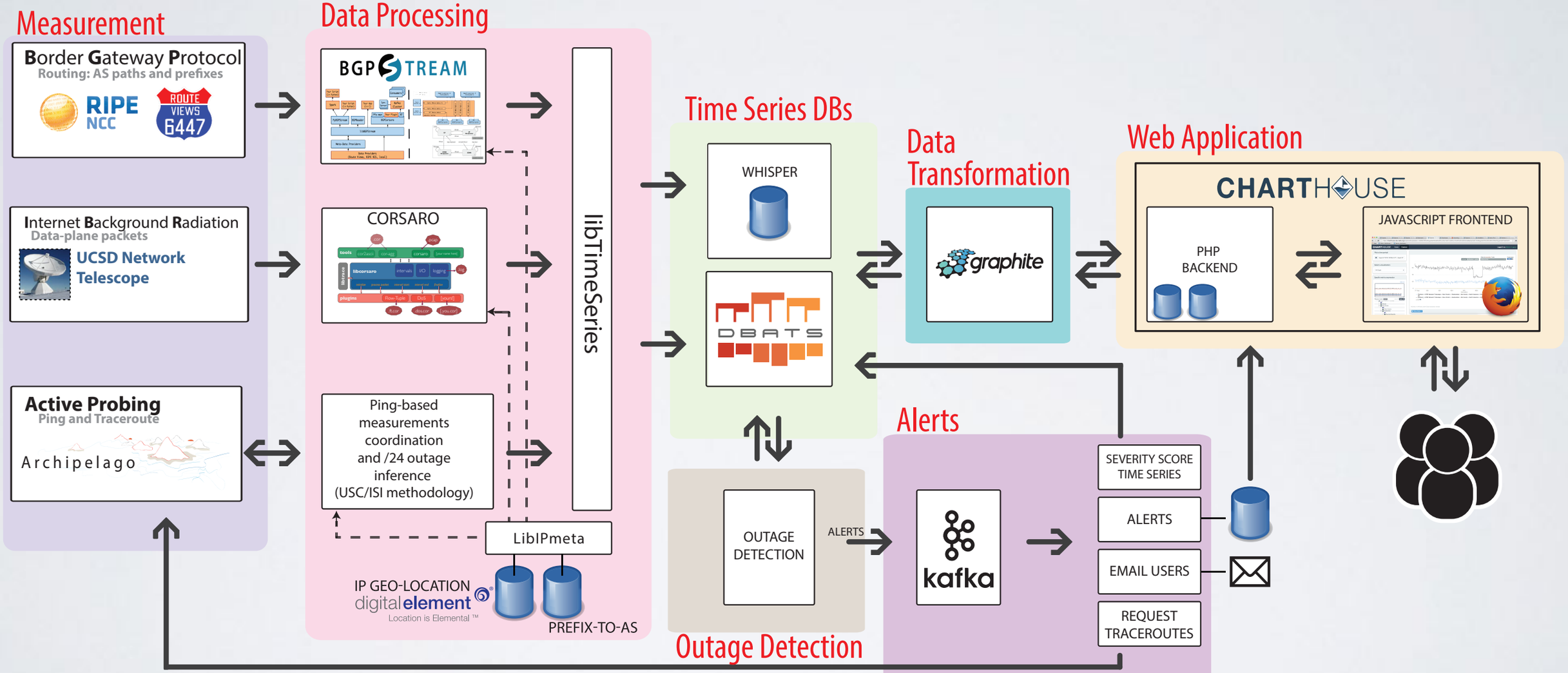
COMCAST

NSF

U.S. DEPARTMENT OF HOMELAND SECURITY

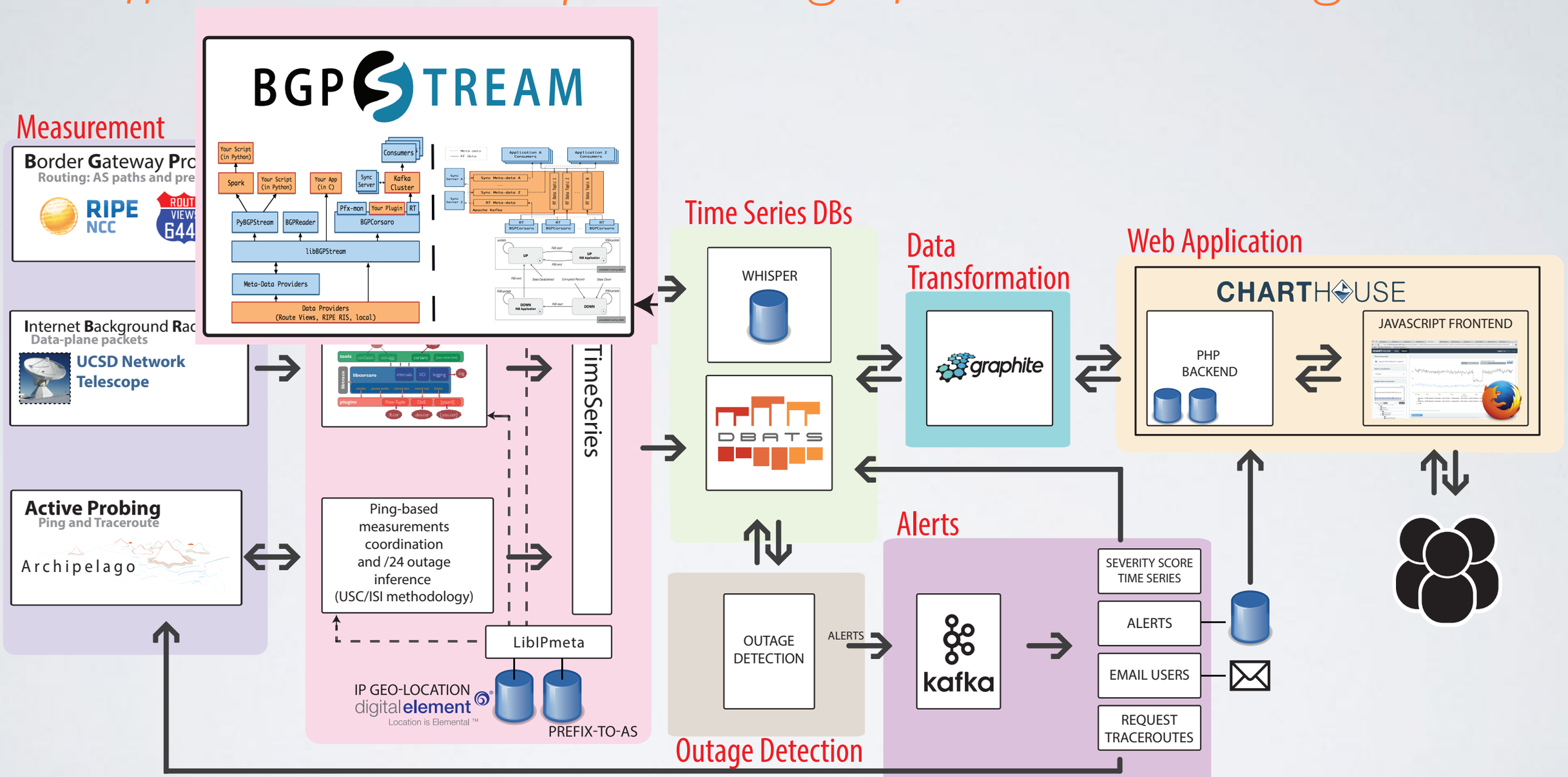
IODA'S CITY MAP

high-level system view



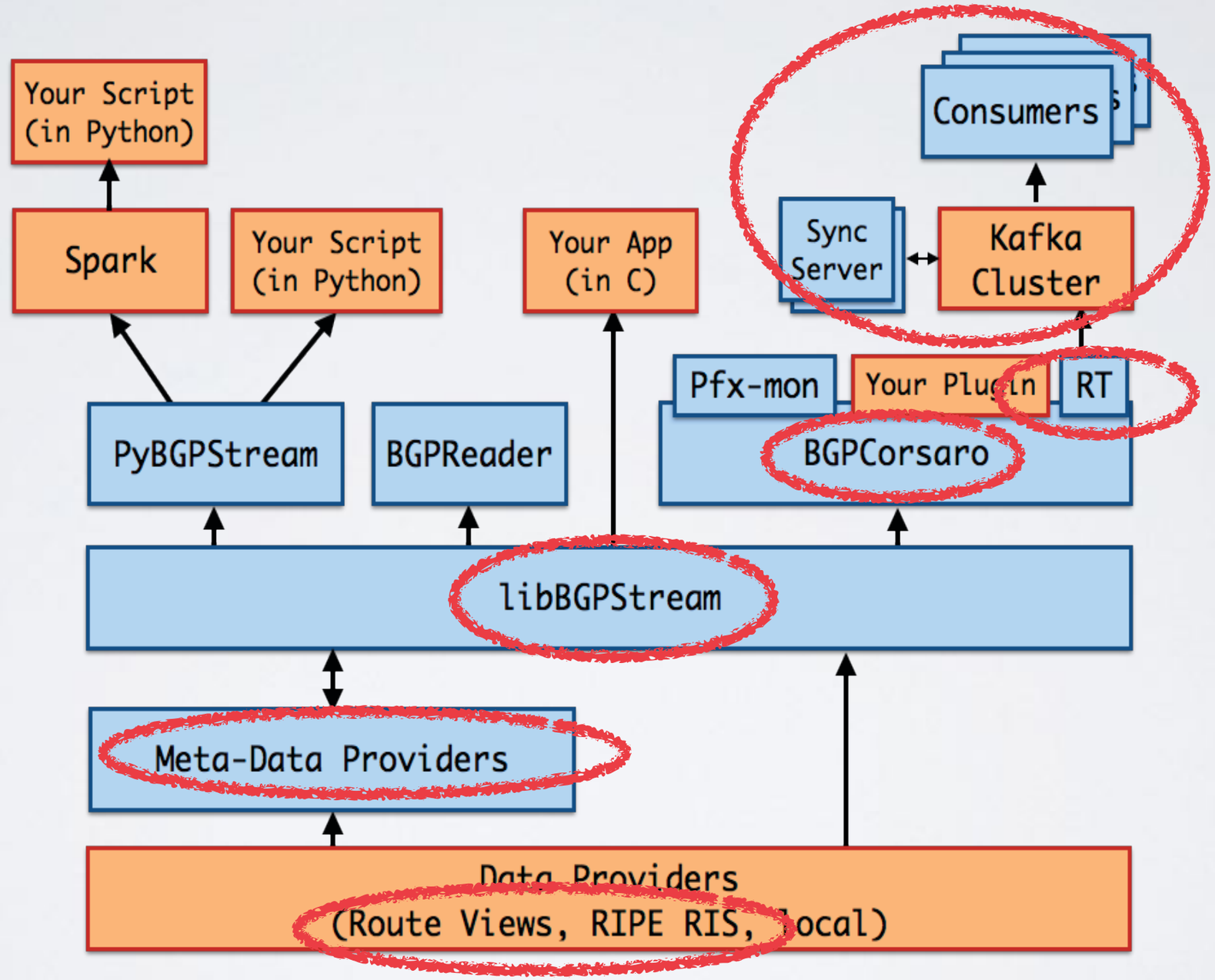
BGPSTREAM

efficient scalable processing of Internet routing data



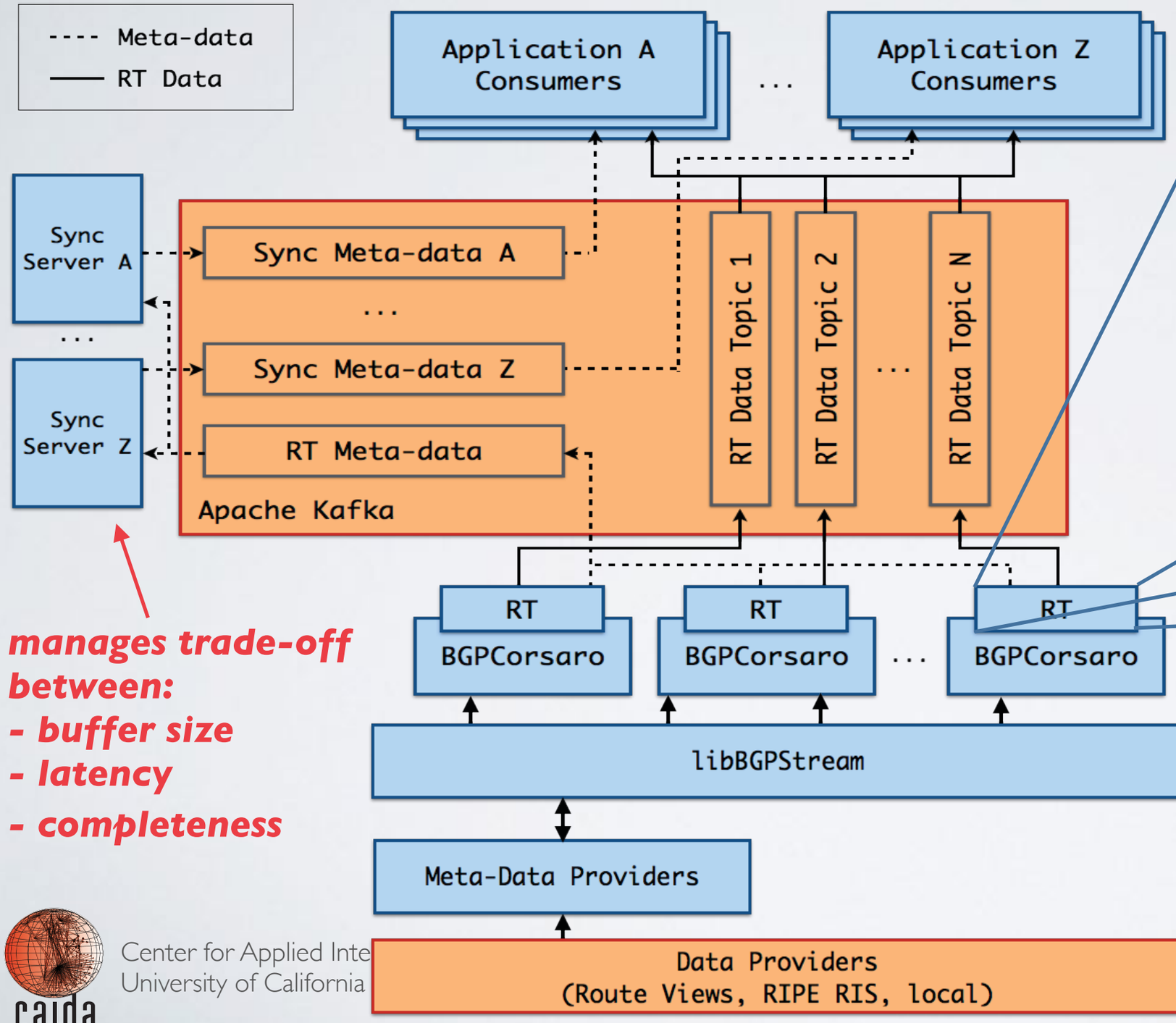
BGPSTREAM IN IODA

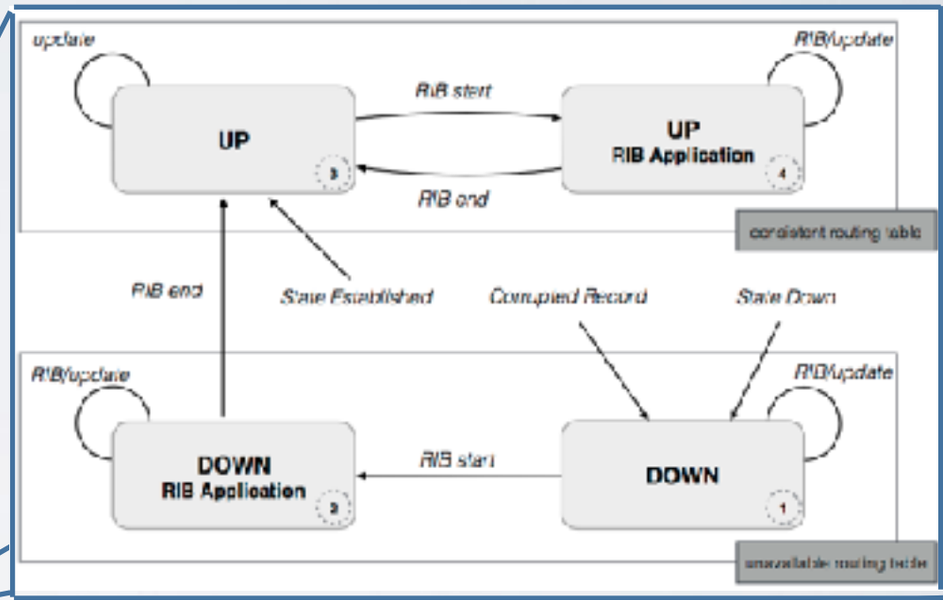
the toolchain we needed to process routing data



BGPSTREAM IN IODA

32 BGPCorsaro instances processing data from ~500 routers





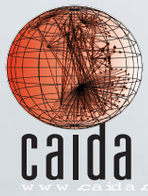
manages trade-off between:
- buffer size
- latency
- completeness

ensures data accuracy and integrity

THANKS

bgpstream.caida.org

alberto@caida.org



Center for Applied Internet Data Analysis
University of California San Diego