Roma Tre University
24th Jun 2016, Rome, IT

BGPStream 2016

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

caida
www.caida.org

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

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?

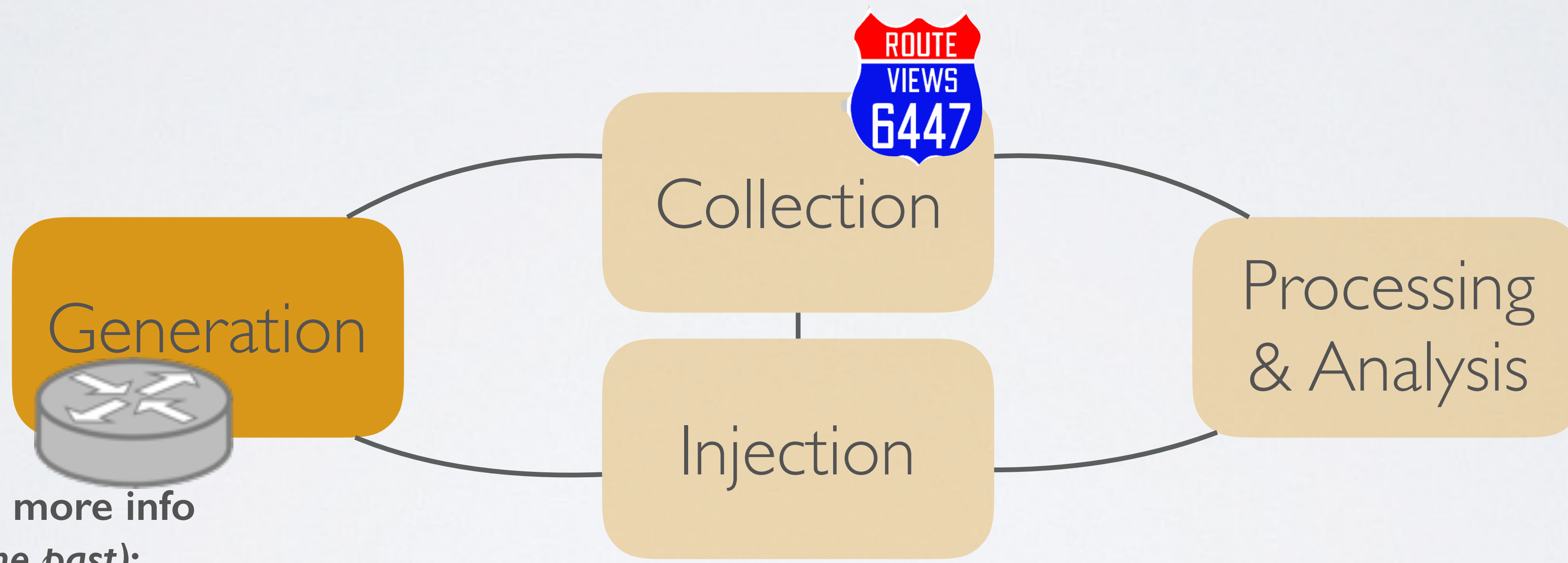
Defining problems and make ***protocol engineering*** decisions through realistic evaluations is difficult also because **we know little about the structure and dynamics of the BGP ecosystem!**

- 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 2013

MEASURING BGP

two issues - somehow related

1. Literature shows that **we need more/better data**
   - more info from the protocol/routers



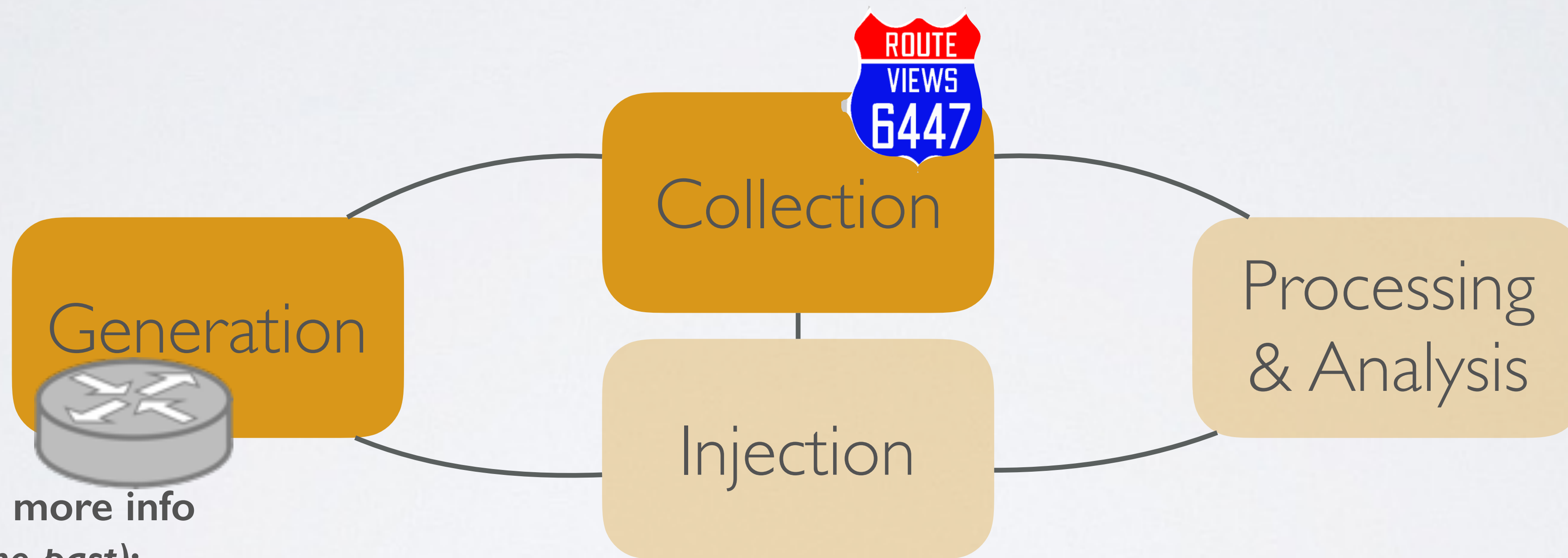
Attempts to generate more info
***(not much traction in the past)*:**

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

MEASURING BGP

two issues - somehow related

1. Literature shows that **we need more/better data**
   - more info from the protocol/routers, more collectors,

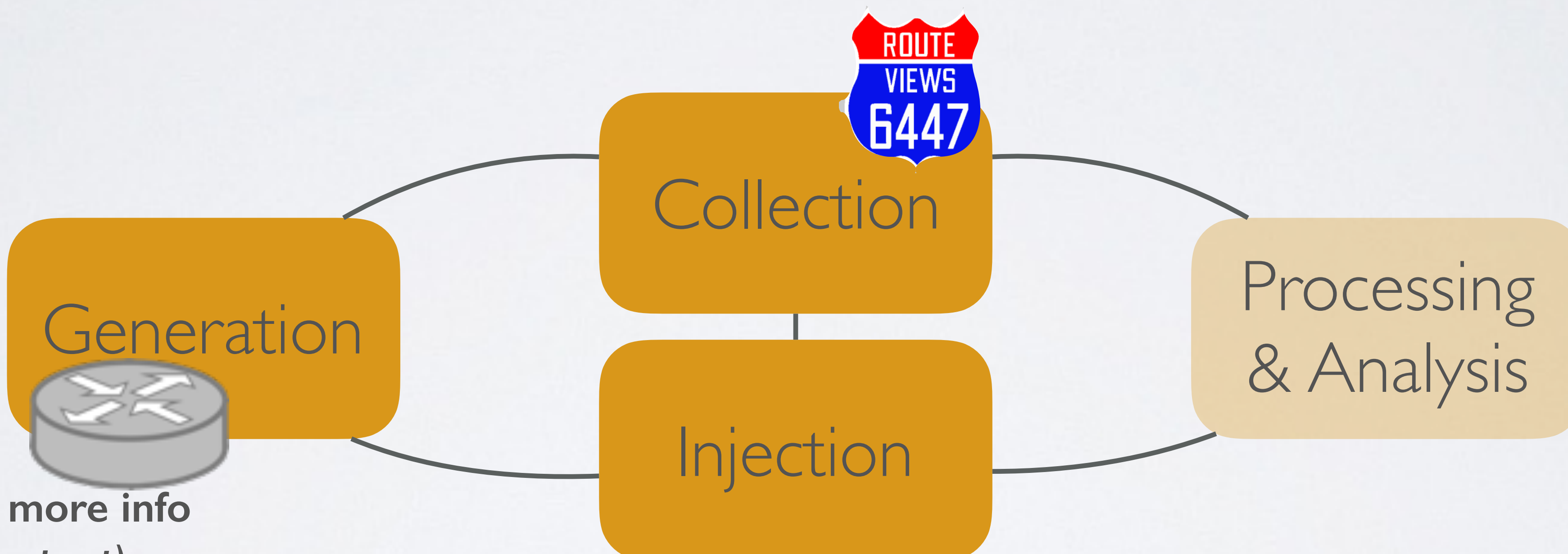


Attempts to generate more info
***(not much traction in the past)*:**
- **`RFC 4384 BGP Communities for Data Collection`**
- **`draft-ymbk-grow-bgp-collector-communities`**

MEASURING BGP

two issues - somehow related

1. Literature shows that **we need more/better data**
   - more info from the protocol/routers, more collectors, more experimental testbeds, …



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

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

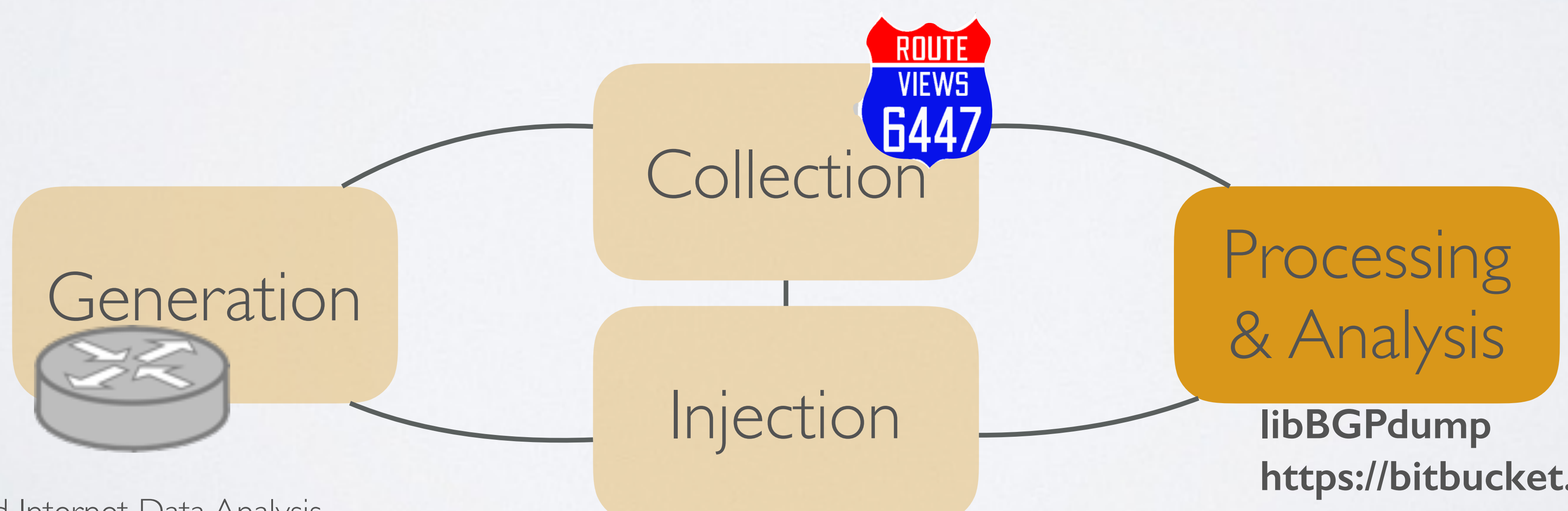
Inject/Receive Routes & Traffic.
PEERING - http://peering.usc.edu
Schlinker et al. *"PEERING: An AS for Us"*, HotNets 2014

MEASURING BGP

two issues - somehow related

1. Literature shows that **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, …



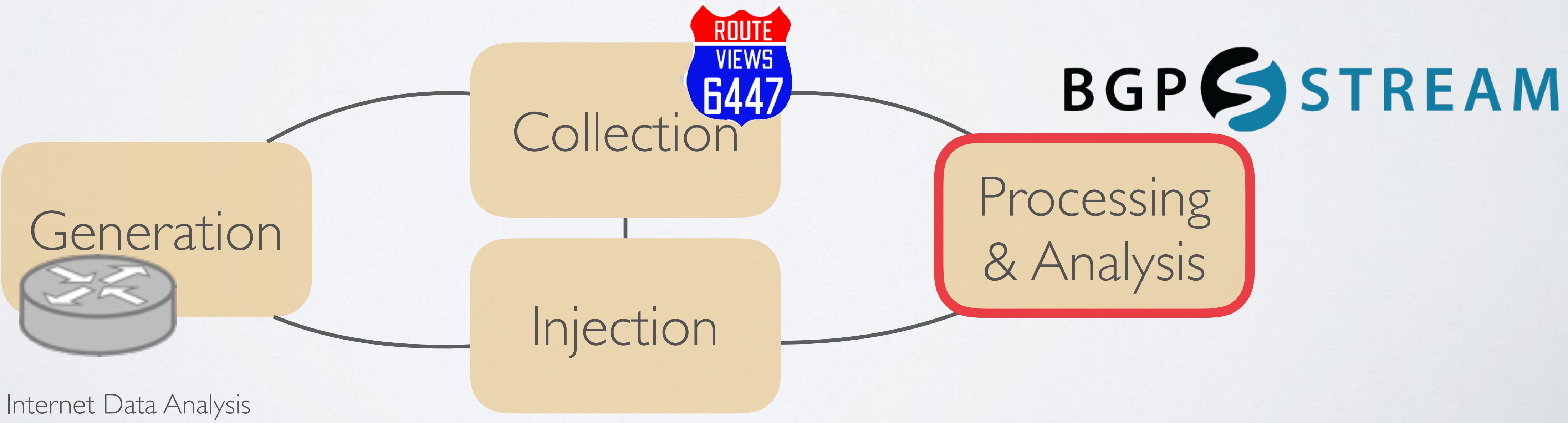
libBGPdump
https://bitbucket.org/ripencc/bgpdump

MEASURING BGP

two issues - somehow related

1. Literature shows that **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, …



INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

- Country-level Internet Blackouts during the Arab Spring

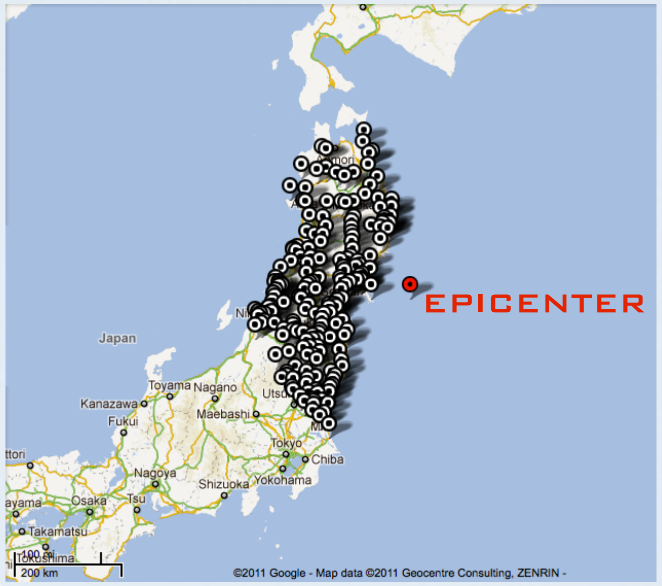
In collaboration with Roma Tre

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

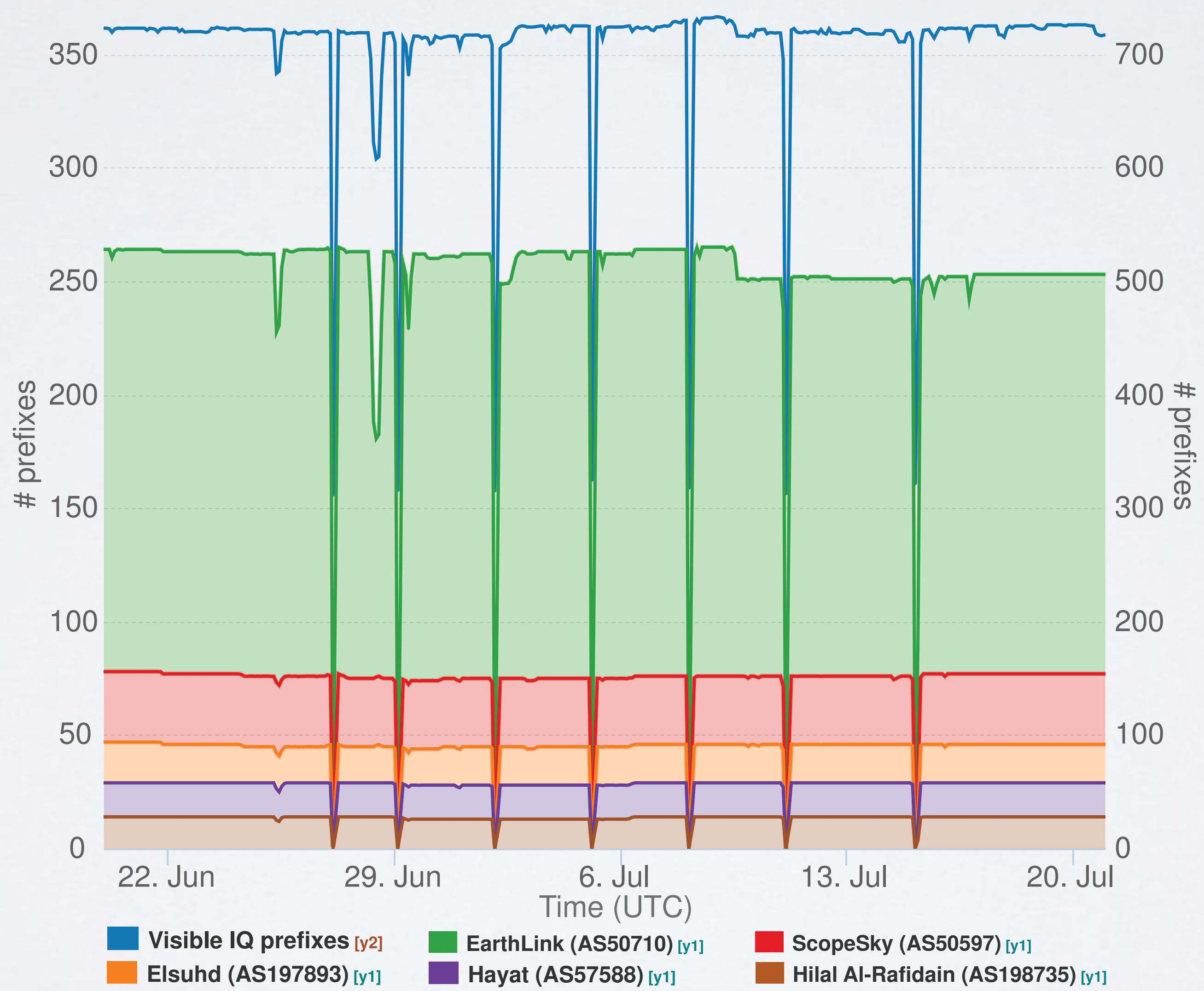


Japan, Mar 2011
Earthquake of magnitude 9.0

INSPIRING PROJECTS (1/2)

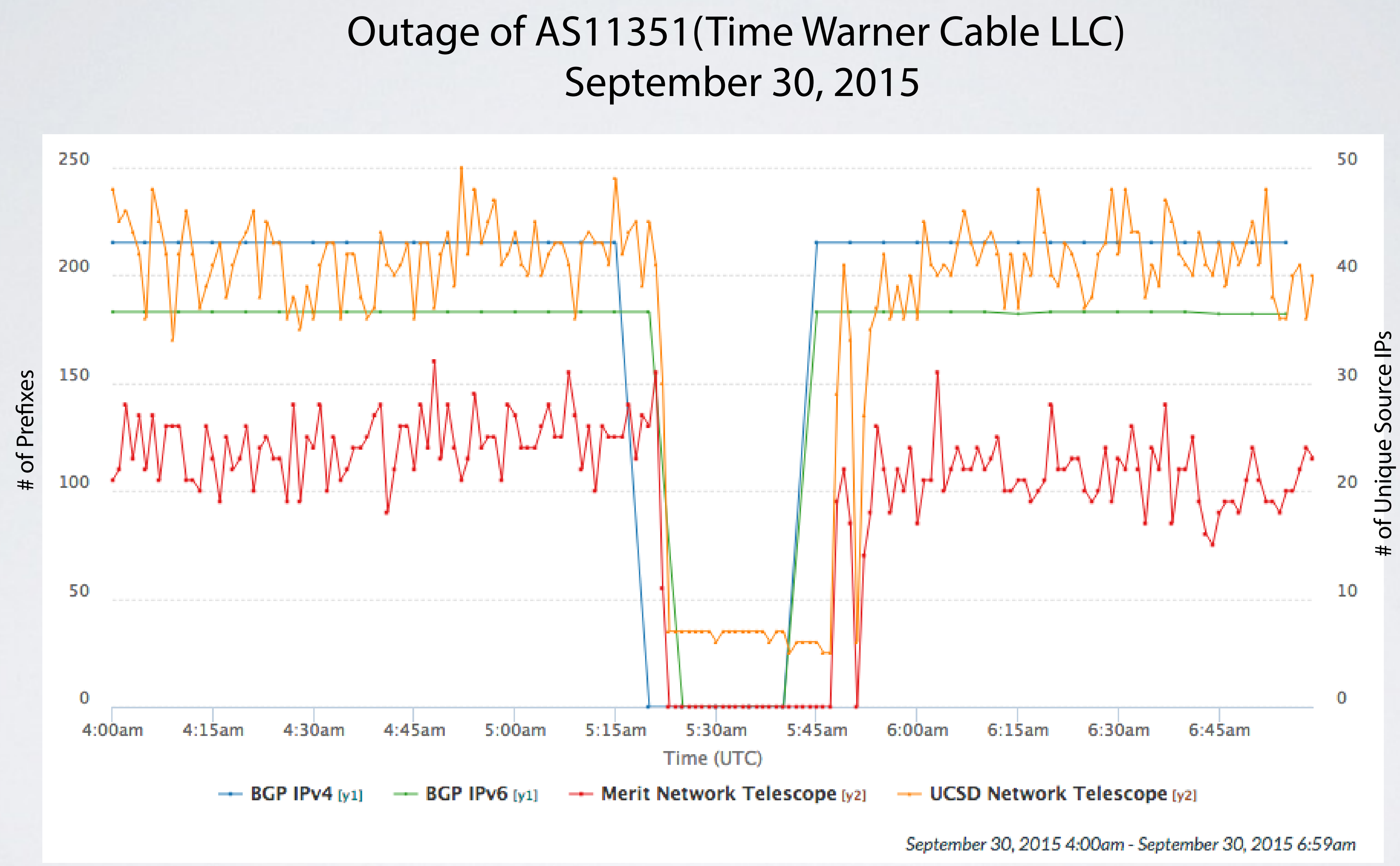
IODA: Detection and Analysis of Internet Outages

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



Center for Applied Internet Data Analysis
University of California San Diego

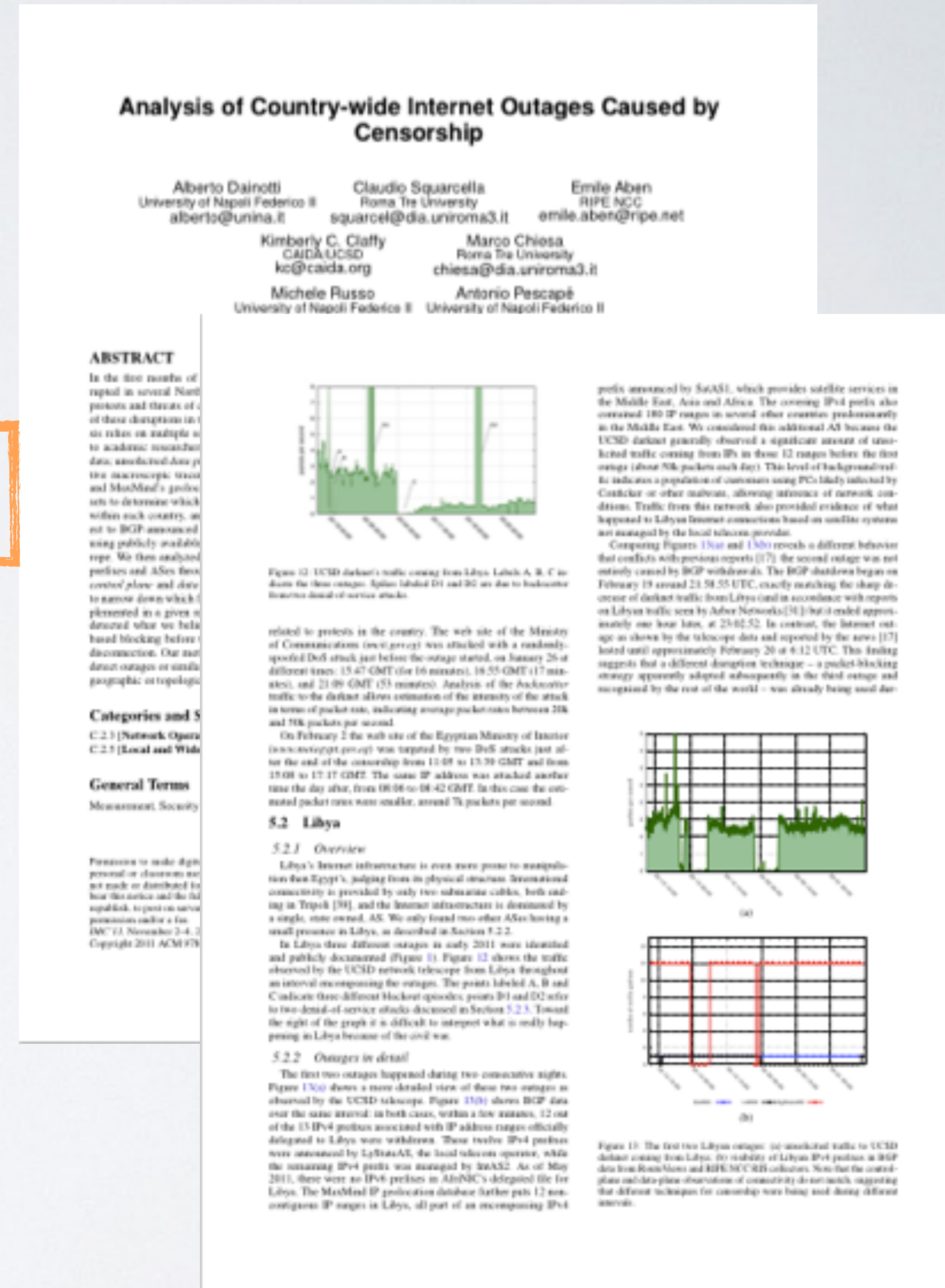
www.caida.org/funding/ioda/

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

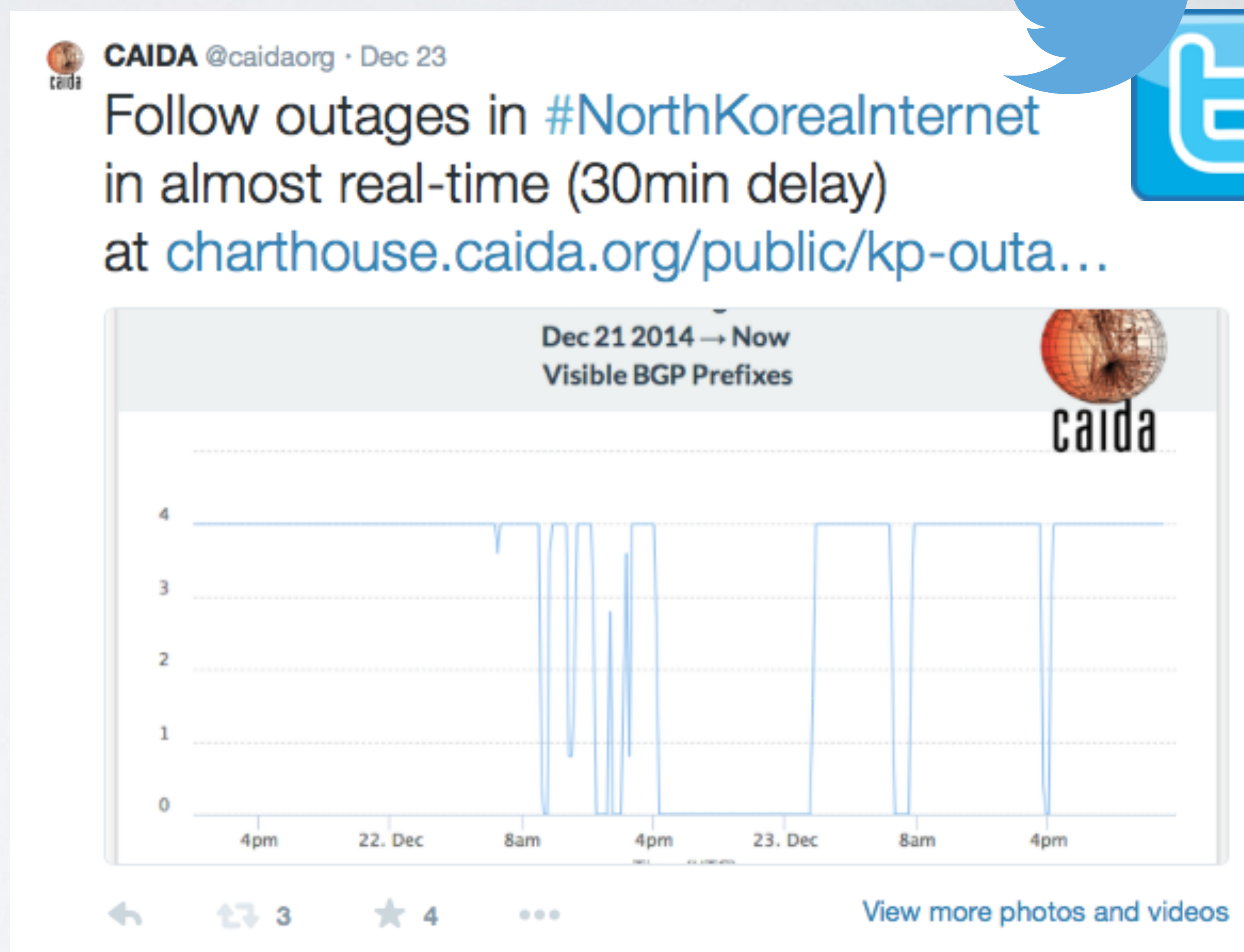
Alberto Dainotti, University of Napoli Federico II, alberto@unina.it
Claudio Squarcella, Roma Tre University, squarcel@dia.uniroma3.it
Emile Aben, RIPE NCC, emile.aben@ripe.net
Kimberly C. Claffy, CAIDA/UCSD, kc@caida.org
Marco Chiesa, Roma Tre University, chiesa@dia.uniroma3.it
Michele Russo, University of Napoli Federico II
Antonio Pescapè, University of Napoli Federico II

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

IODA TODAY

live Internet monitoring

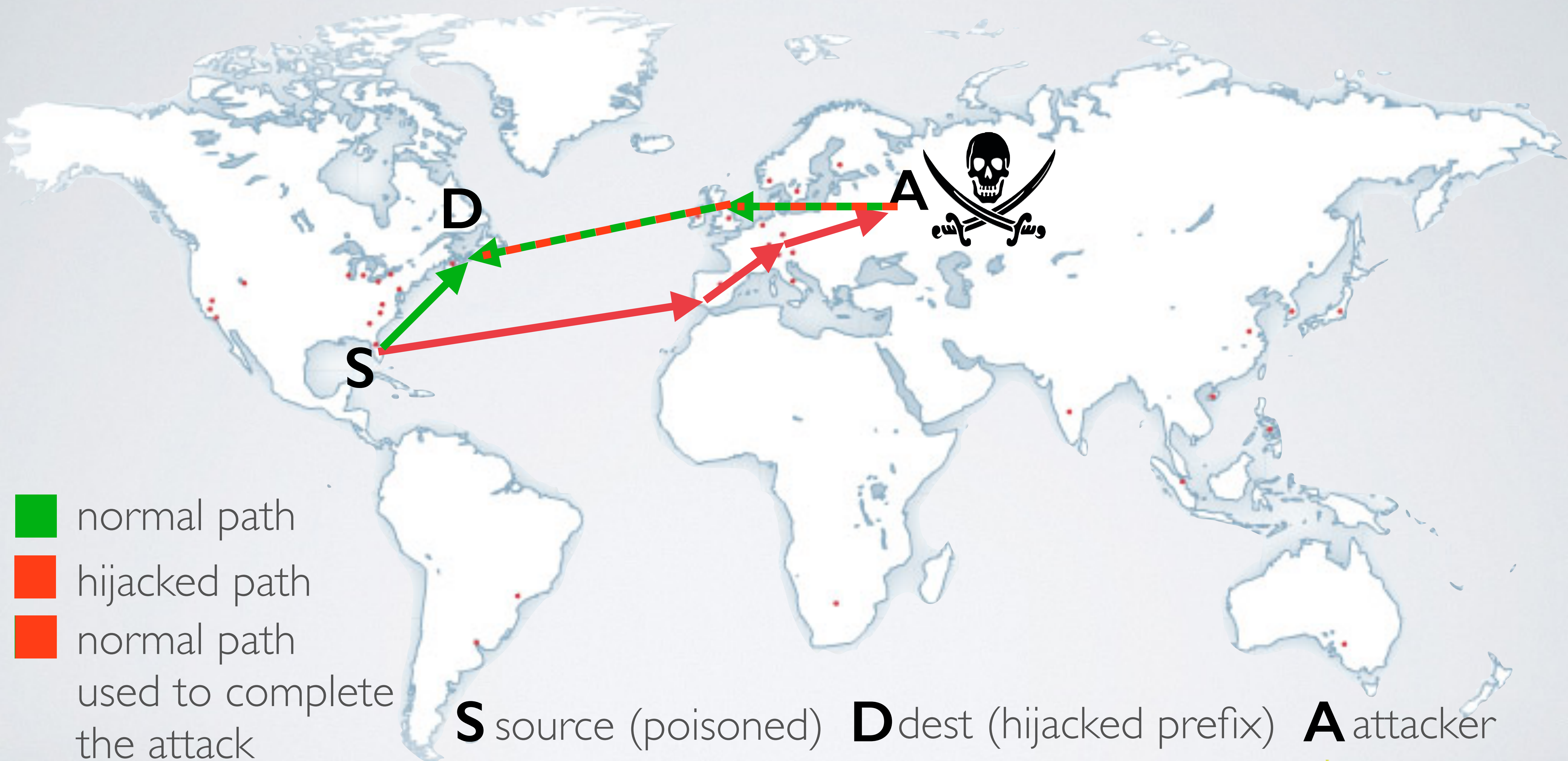
In Dec. 2014 we made it possible for anybody to follow the North Korean disconnection almost live



https://charthouse.caida.org/public/kp-outage

INSPIRING PROJECTS (2/2)

Hijacks: detection of MITM BGP attacks



caida

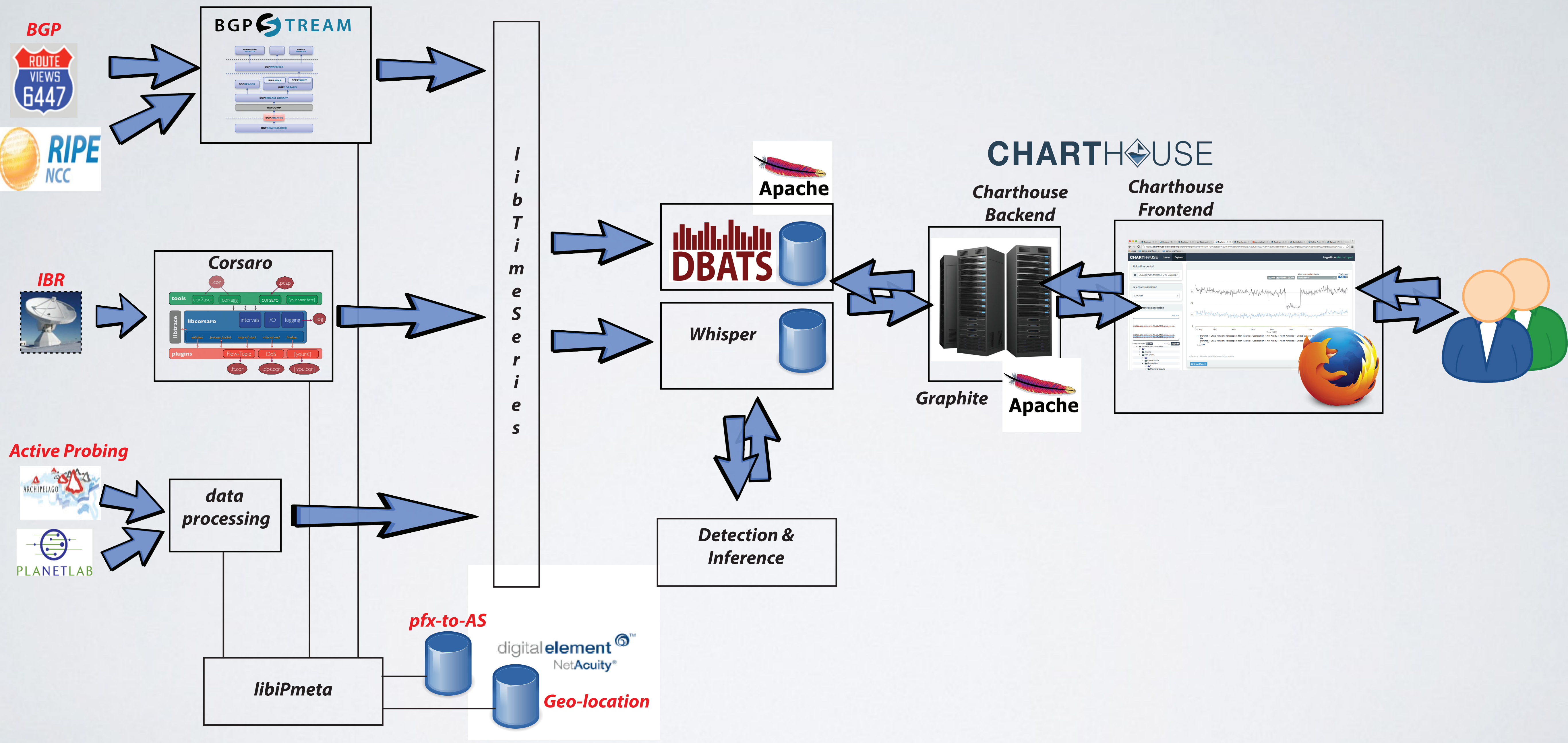
Stony Brook University

www.caida.org/funding/hijacks/

COMCAST

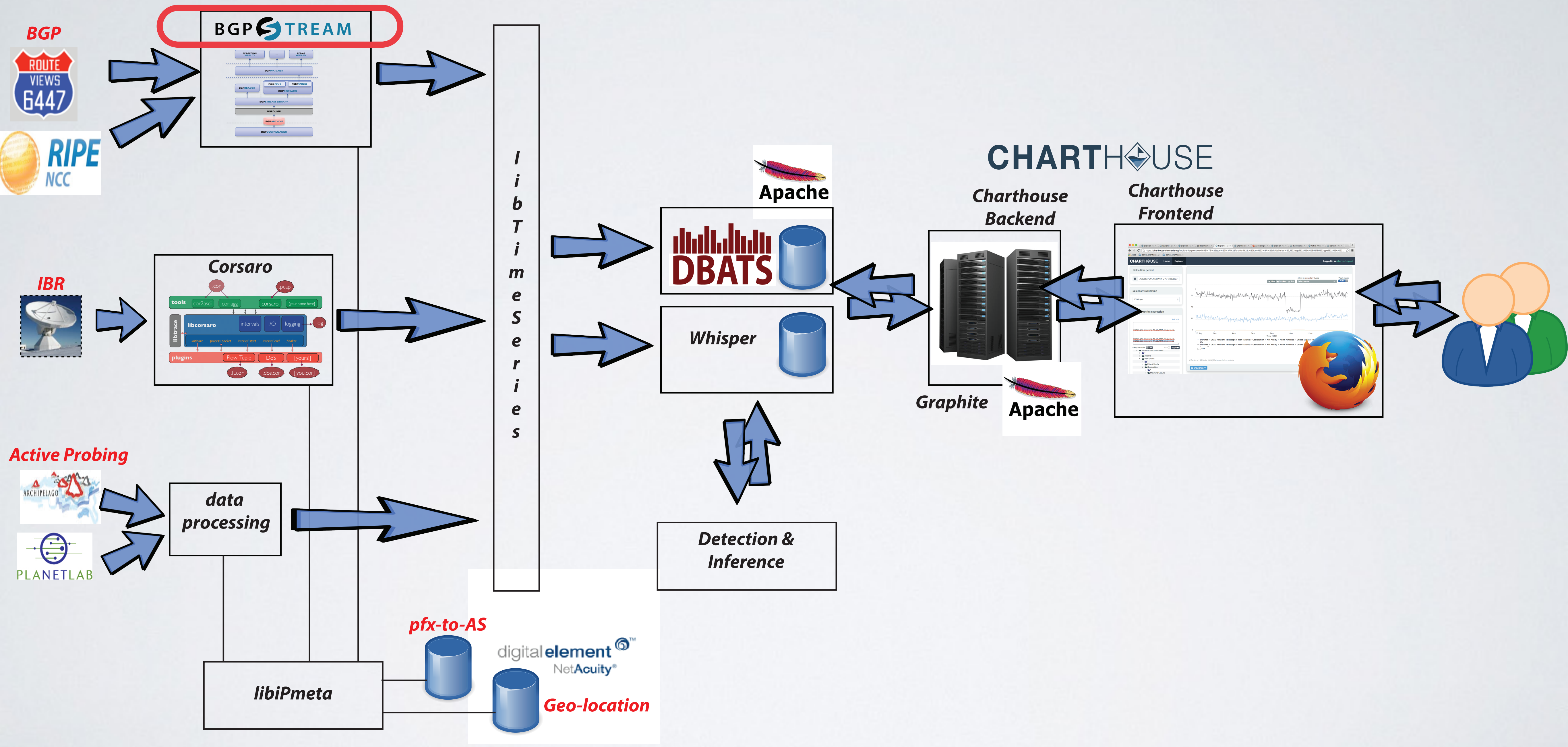
IODA SYSTEM DIAGRAM

(toy diagram)



IODA SYSTEM DIAGRAM

(toy diagram)



Center for Applied Internet Data Analysis
University of California San Diego

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

- Paper under submission at IMC '16

Orsini, King, Giordano, Giotsas, Dainotti

(older tech report on web site)


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

Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, Alberto Dainotti
CAIDA, UC San Diego

ABSTRACT

BGP Data at Router Level

1. INTRODUCTION

2. BACKGROUND

BGPSTREAM

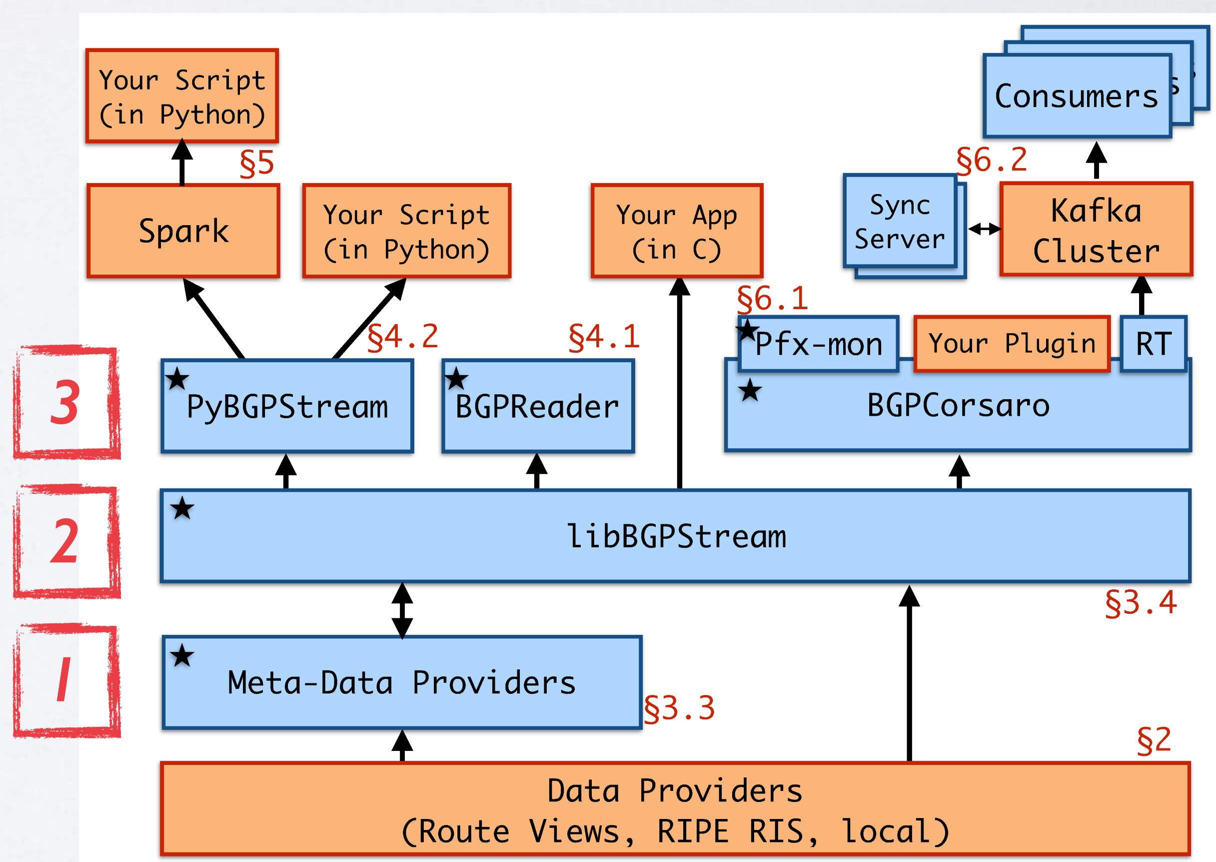
it's real!

- ***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 last February, NANOG Hackathon in June, …
- Funding from Cisco to collaborate and natively support OpenBMP

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

BGP STREAM

specifying a stream

```
int main(int argc, const char **argv)
{
  bgpstream_t *bs = bgpstream_create();
  bgpstream_record_t *record = bgpstream_record_create();
  bgpstream_elem_t *elem = NULL;
  char buffer[1024];

  /* Define the prefix to monitor for (2403:f600::/32) */
  bgpstream_pfx_storage_t my_pfx;
  my_pfx.address.version = BGPSTREAM_ADDR_VERSION_IPV6;
  inet_pton(BGPSTREAM_ADDR_VERSION_IPV6, "2403:f600::", &my_pfx.address.ipv6);
  my_pfx.mask_len = 32;

  /* Set metadata filters */
  bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "rrc00");
  bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_COLLECTOR, "route-views2");
  bgpstream_add_filter(bs, BGPSTREAM_FILTER_TYPE_RECORD_TYPE, "updates");
  /* Time interval: 01:20:10 - 06:32:15 on Tue, 12 Aug 2014 UTC */
  bgpstream_add_interval_filter(bs, 1407806410, 1407825135);

  /* Start the stream */
  bgpstream_start(bs);
```

LIBBGPSTREAM API

BGP record

- **A BGP record encapsulate 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 multiple update messages can be in the same MRT record (see next slide)

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

BGP 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

```c
/* Start the stream */
bgpstream_start(bs);

/* Read the stream of records */
while (bgpstream_get_next_record(bs, record) > 0) {
  /* Ignore invalid records */
  if (record->status != BGPSTREAM_RECORD_STATUS_VALID_RECORD) {
    continue;
  }
  /* Extract elems from the current record */
  while ((elem = bgpstream_record_get_next_elem(record)) != NULL) {
    /* Select only announcements and withdrawals, */
    /* and only elems that carry information for 2403:f600::/32 */
    if ((elem->type == BGPSTREAM_ELEM_TYPE_ANNOUNCEMENT ||
         elem->type == BGPSTREAM_ELEM_TYPE_WITHDRAWAL) &&
        bgpstream_pfx_storage_equal(&my_pfx, &elem->prefix)) {
      /* Print the BGP information */
      bgpstream_elem_snprintf(buffer, 1024, elem);
      fprintf(stdout, "%s\n", buffer);
    }
  }
}
```

BGPREADER

command-line tool for ASCII output w/ filters

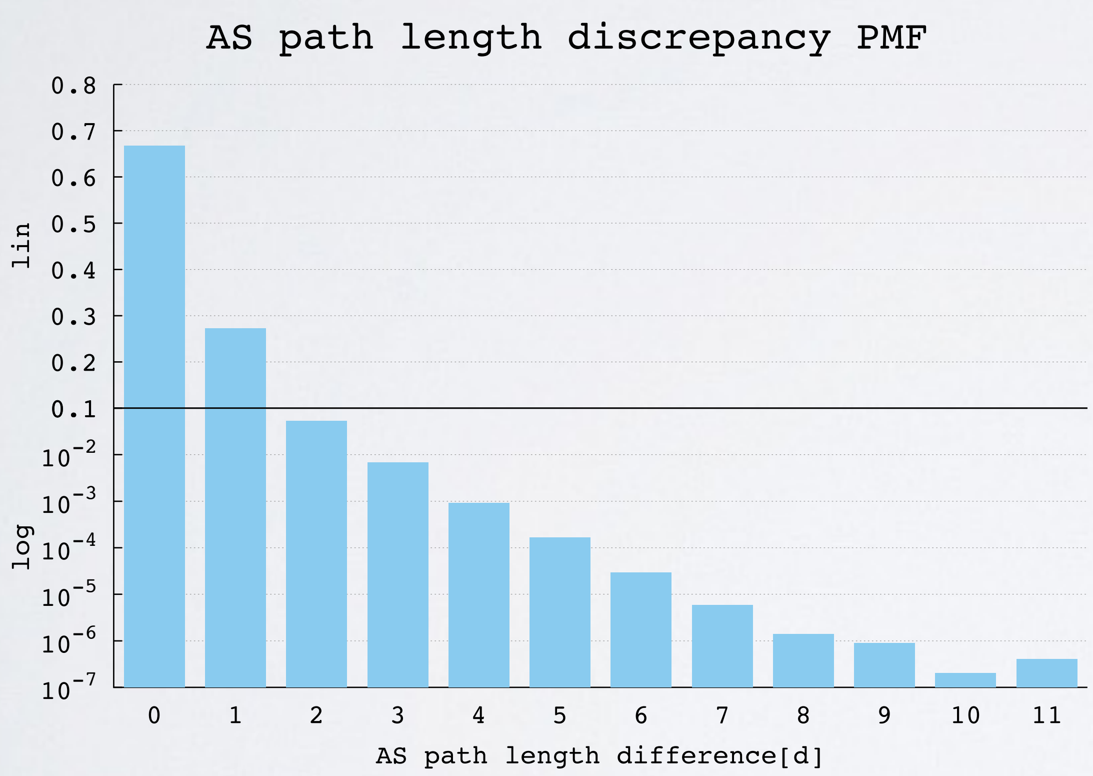
```
$ bgpreader -w 1445306400,1445306402 -c route-views.sfmix
R|B|1445306400|routeviews|route-views.sfmix
R|R|1445306400|routeviews|route-views.sfmix|32354|206.197.187.5|1.0.0.0/24|206.197.187.5|32354 15169|15169|||
...
R|R|1445306401|routeviews|route-views.sfmix|14061|2001:504:30::ba01:4061:1|2c0f:ffd8::/32|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

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

PYBGPSTREAM

BGP STREAM

Example: timely combine with active measurements

…… In the paper you'll find a case study that uses ***PyBGPStream*** *to detect blackholing (a mitigation measure against denial-of-service attacks) and triggers traceroute measurements from* ***RIPE Atlas*** *to better characterize the event*

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

Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, Alberto Dainotti
CAIDA, UC San Diego

ABSTRACT

We present BGPStream, an open-source software framework for the analysis of both historical and real-time Border Gateway Protocol (BGP) measurement data. Although BGP is a crucial operational component of the Internet infrastructure, and is the subject of research in the areas of Internet performance, security, topology, protocols, economics, etc., there is no efficient way of processing large amounts of distributed and/or live BGP measurement data. BGPStream fills this gap, enabling efficient investigation of events, rapid prototyping, and building complex tools large-scale monitoring applications (e.g., detection of connectivity disruptions or BGP hijacking attacks). We discuss the goals and architecture of BGPStream. We apply the components of the framework to different scenarios, and we describe the development and deployment of complex services for global Internet monitoring that we built on top of it.

BGP Data at Router Level

The Border Gateway Protocol (BGP) is the de-facto standard inter-domain routing protocol for the Internet: its primary function is to exchange reachability information among Autonomous Systems (ASes) [50]. Each AS announces to the others, by means of BGP update messages, the routes to its local prefixes and the preferred routes learned from its neighbors. Such messages provide information about how a destination can be reached through an ordered list of AS hops, called an *AS path*.

A BGP router maintains this reachability information in the *Routing Information Base* (RIB) [50], 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 ap-

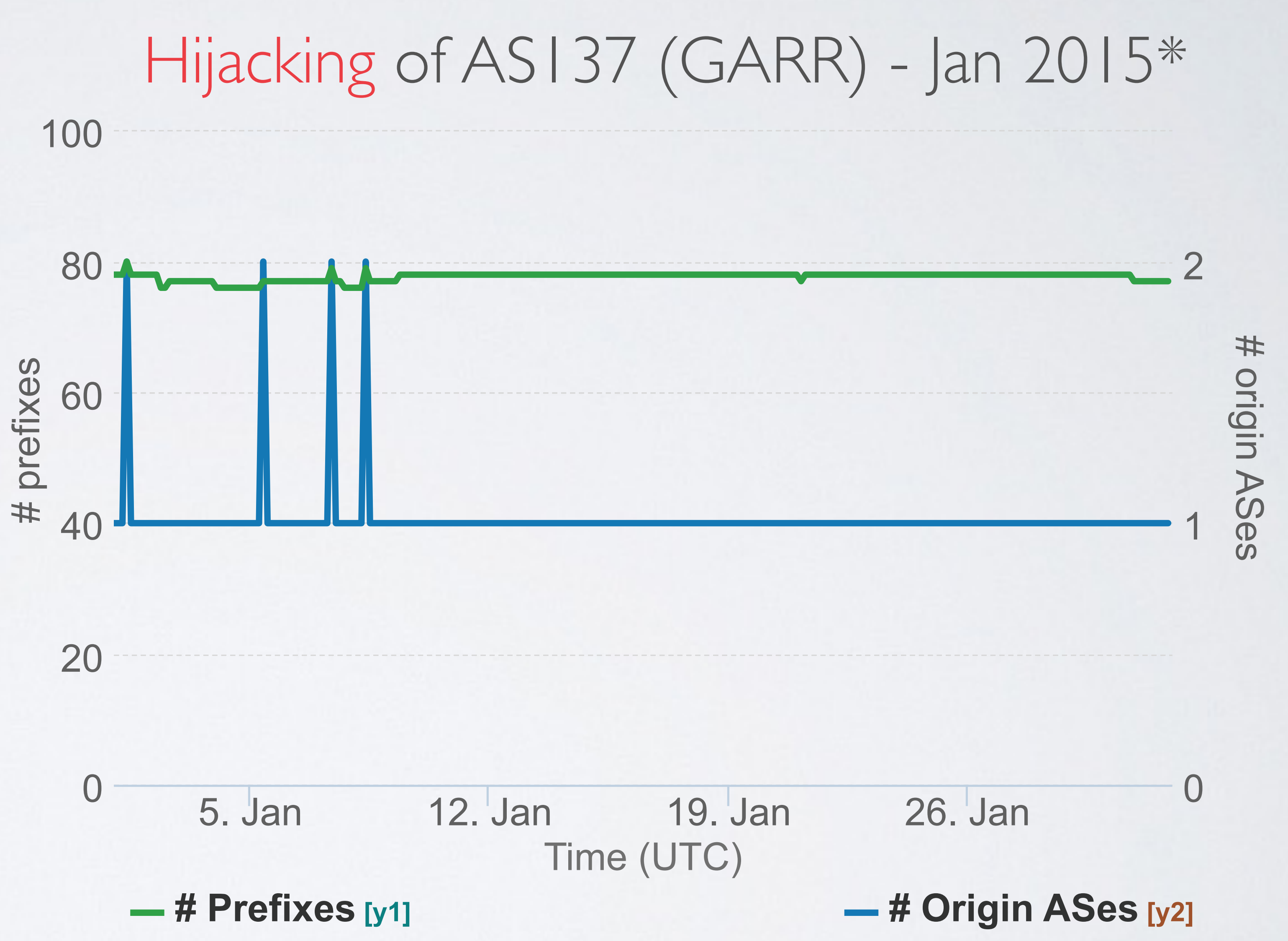
BGPCORSARO

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

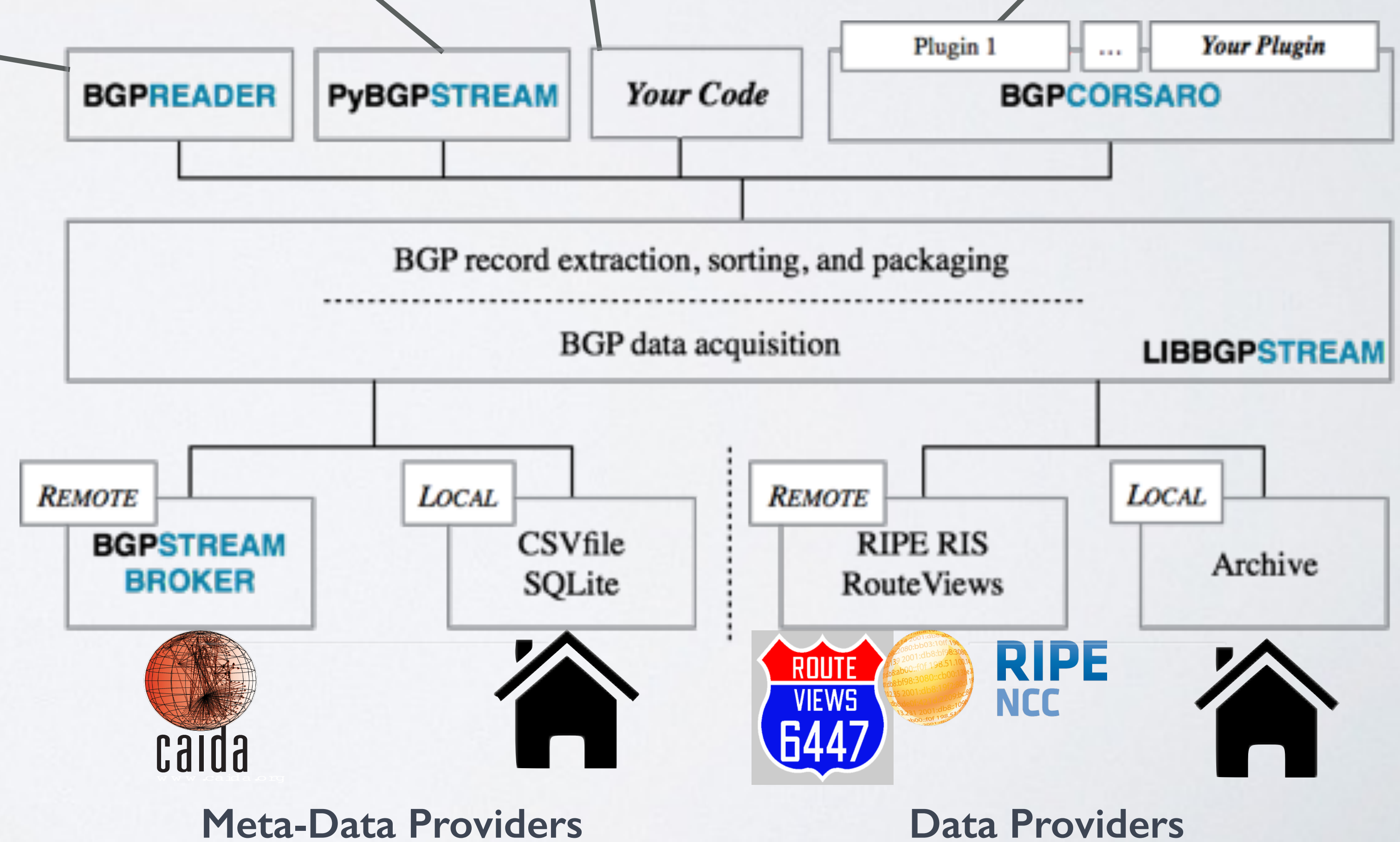
libBGPStream talks to the broker and gets the data

```
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

Center for Applied Internet Data Analysis
University of California San Diego

GET A LIVE STREAM

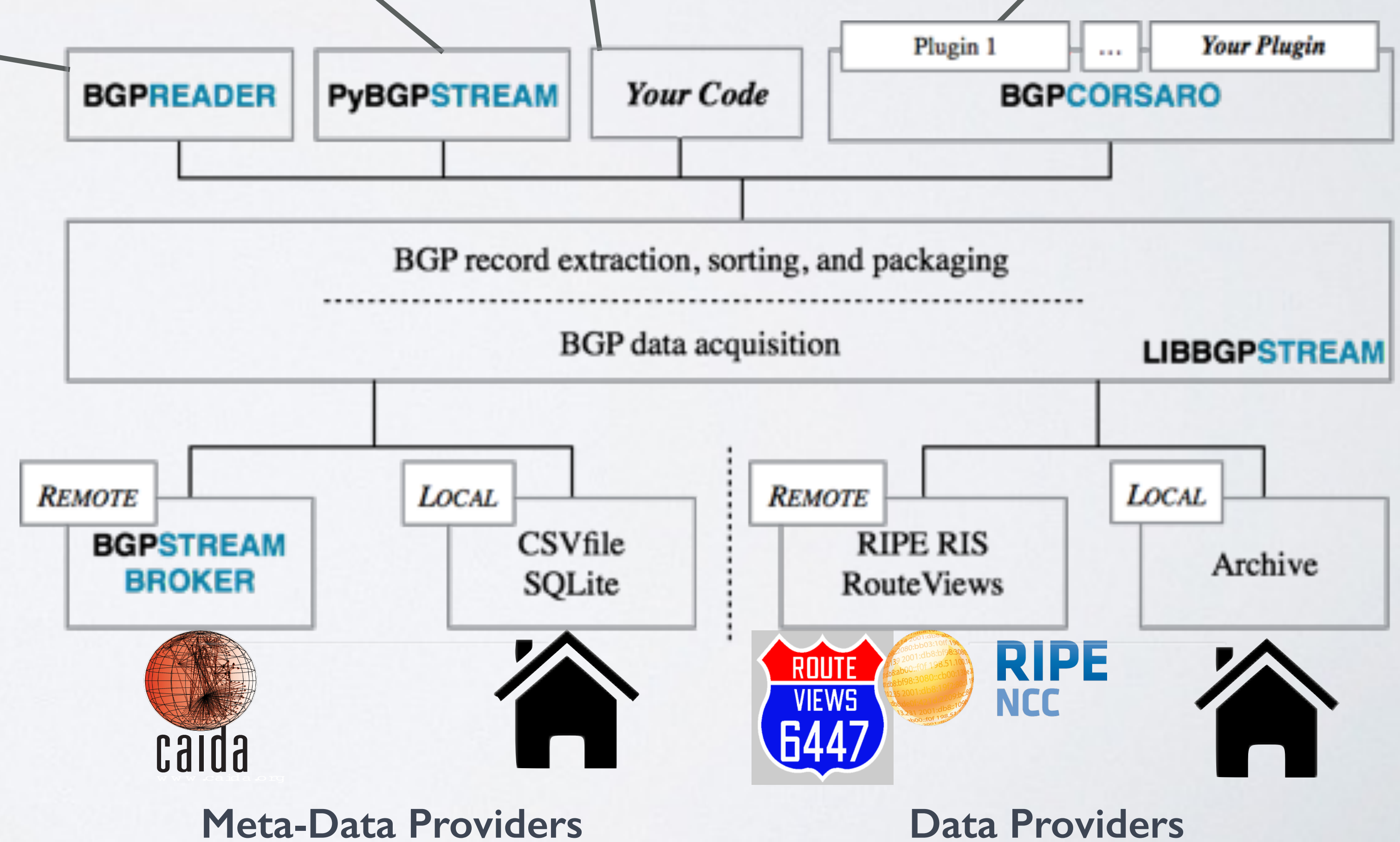
libBGPStream keeps retrieving data as it becomes available

```
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)
```

```
$ bgpcorsaro -p ris
```

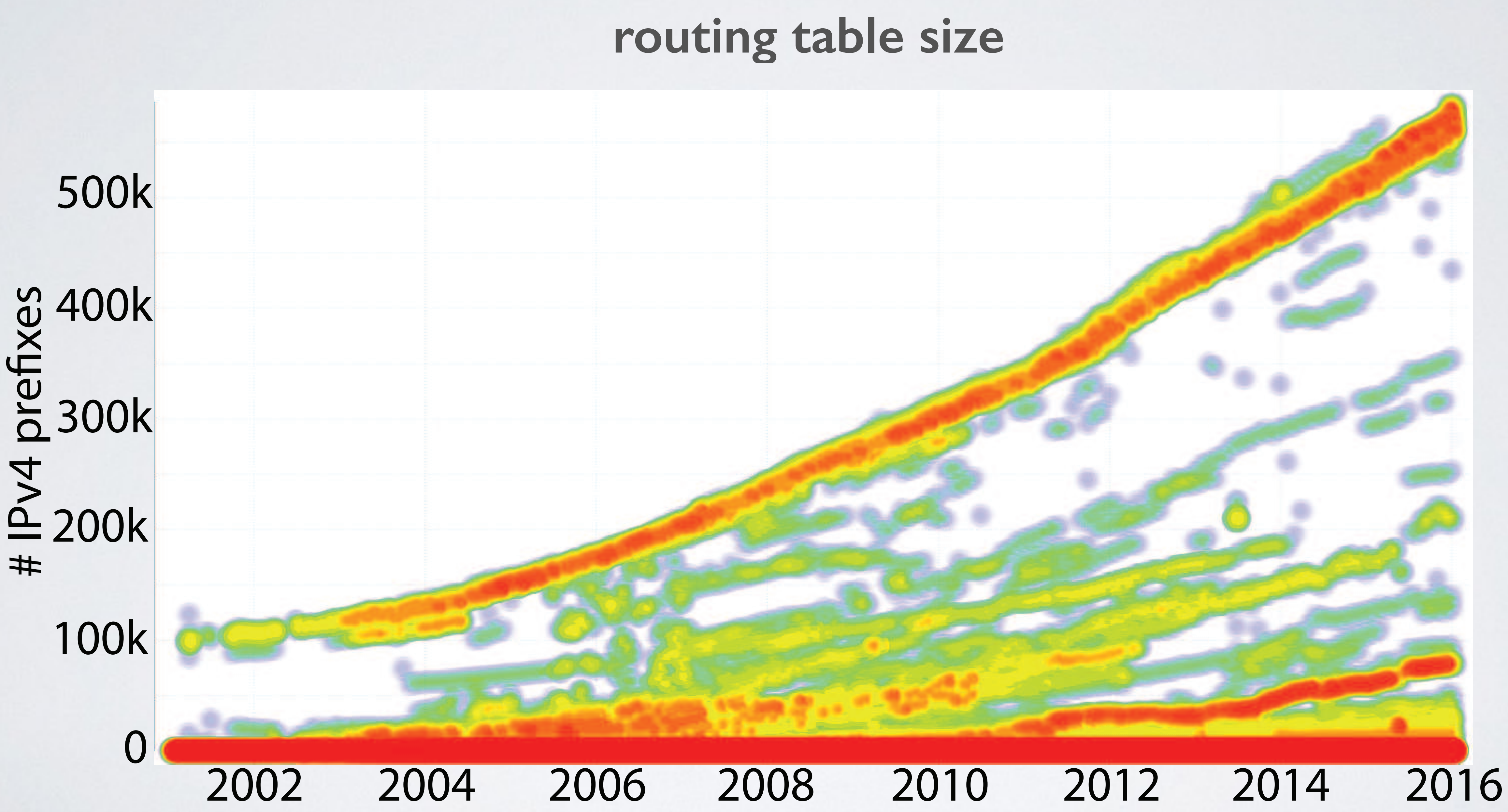
```
$ bgpreader -c route-views.sfmix -t updates
```



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

CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

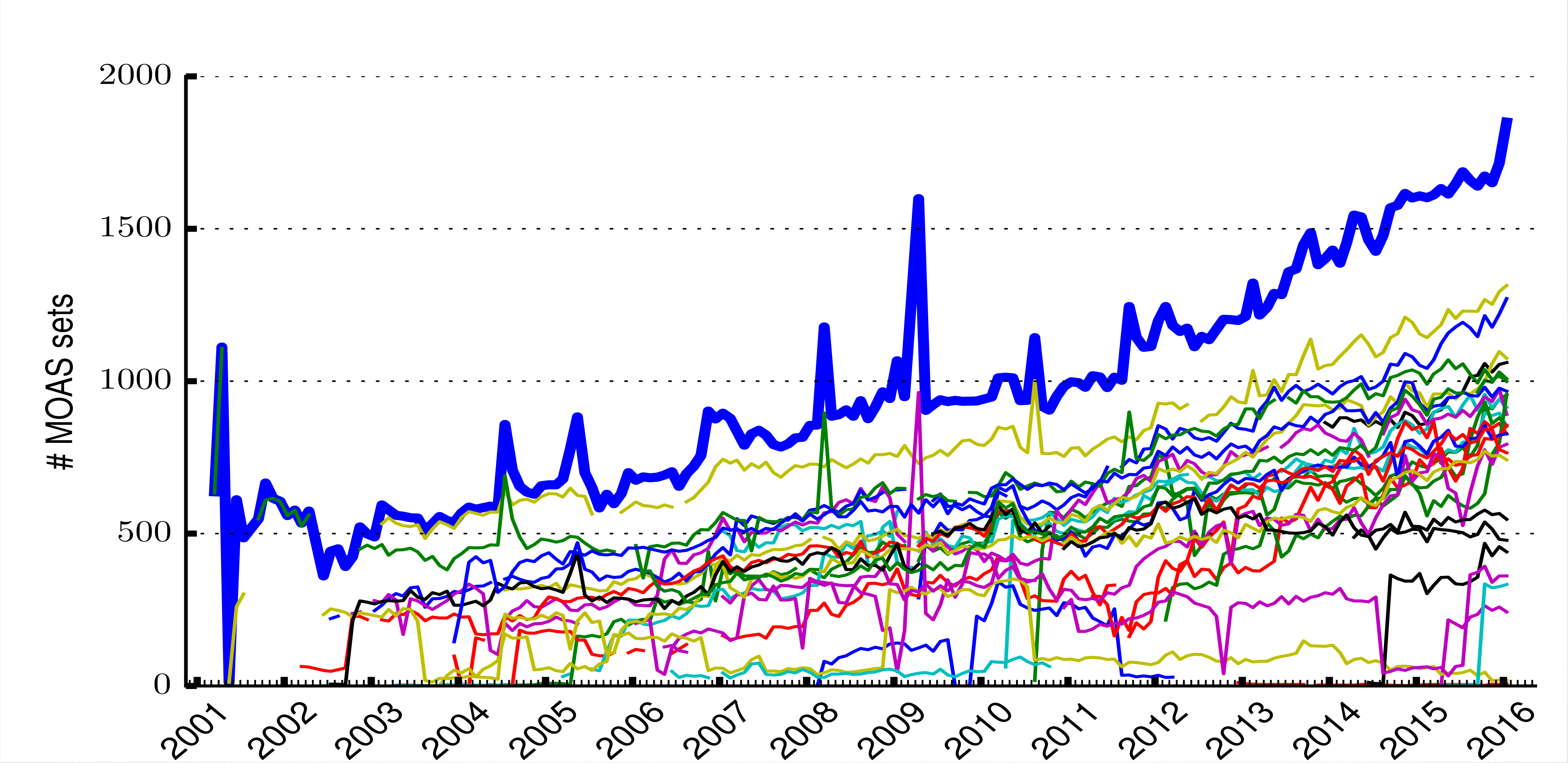


Center for Applied Internet Data Analysis
University of California San Diego

CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

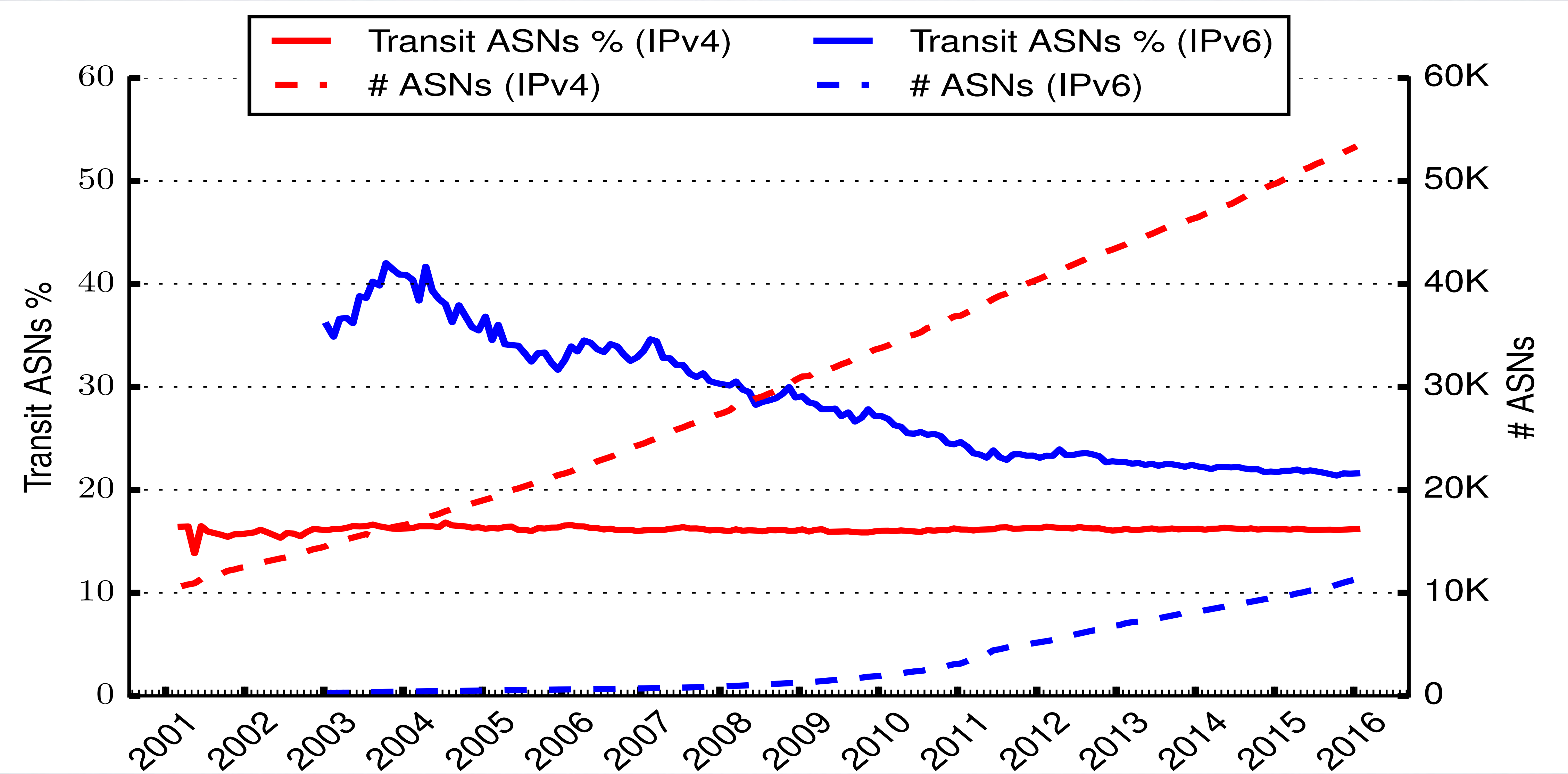
MOAS Sets



CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

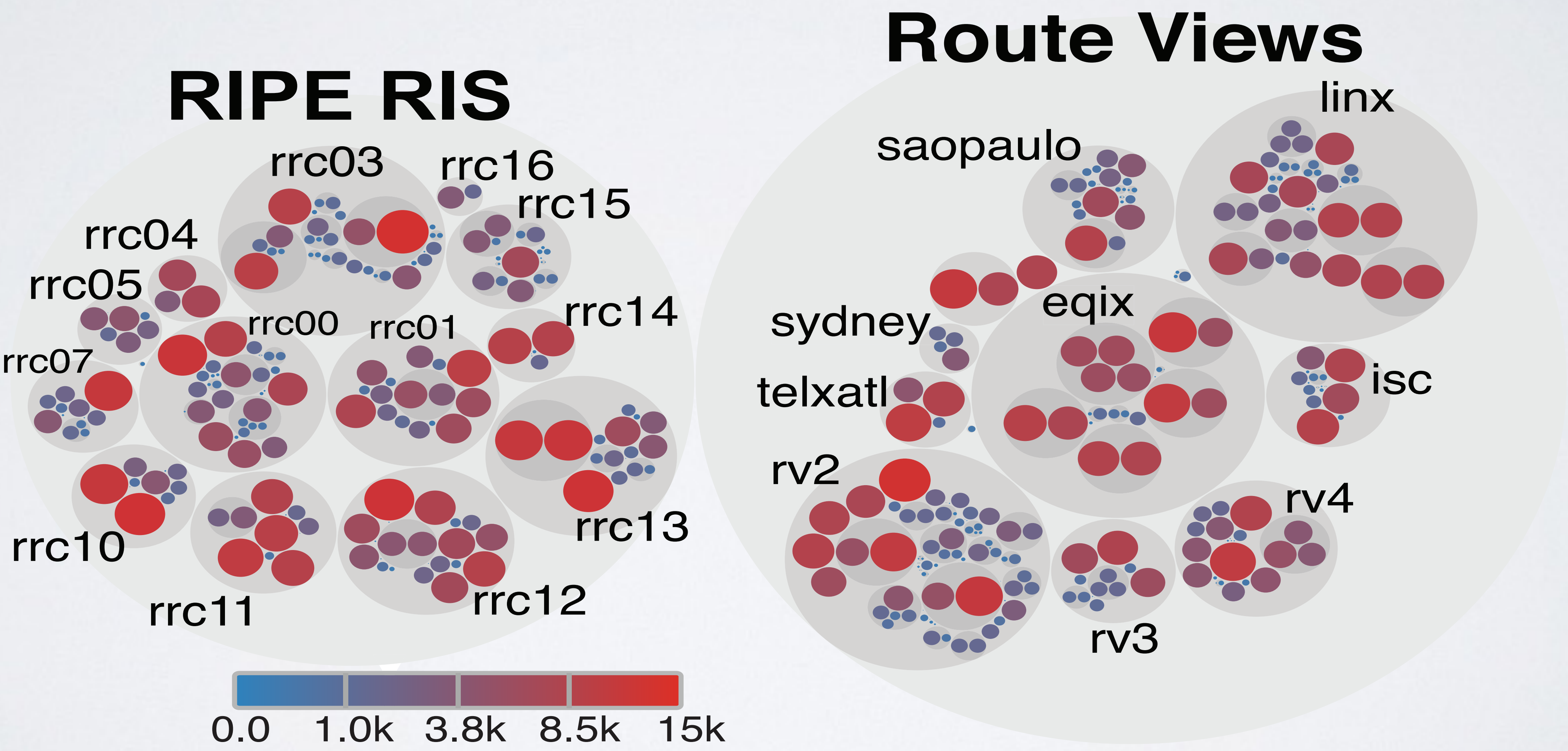
Transit ASes



CRUNCH BIG DATA

44Billion BGPElems processed w/ Spark + PyBGPStream

BGP communities



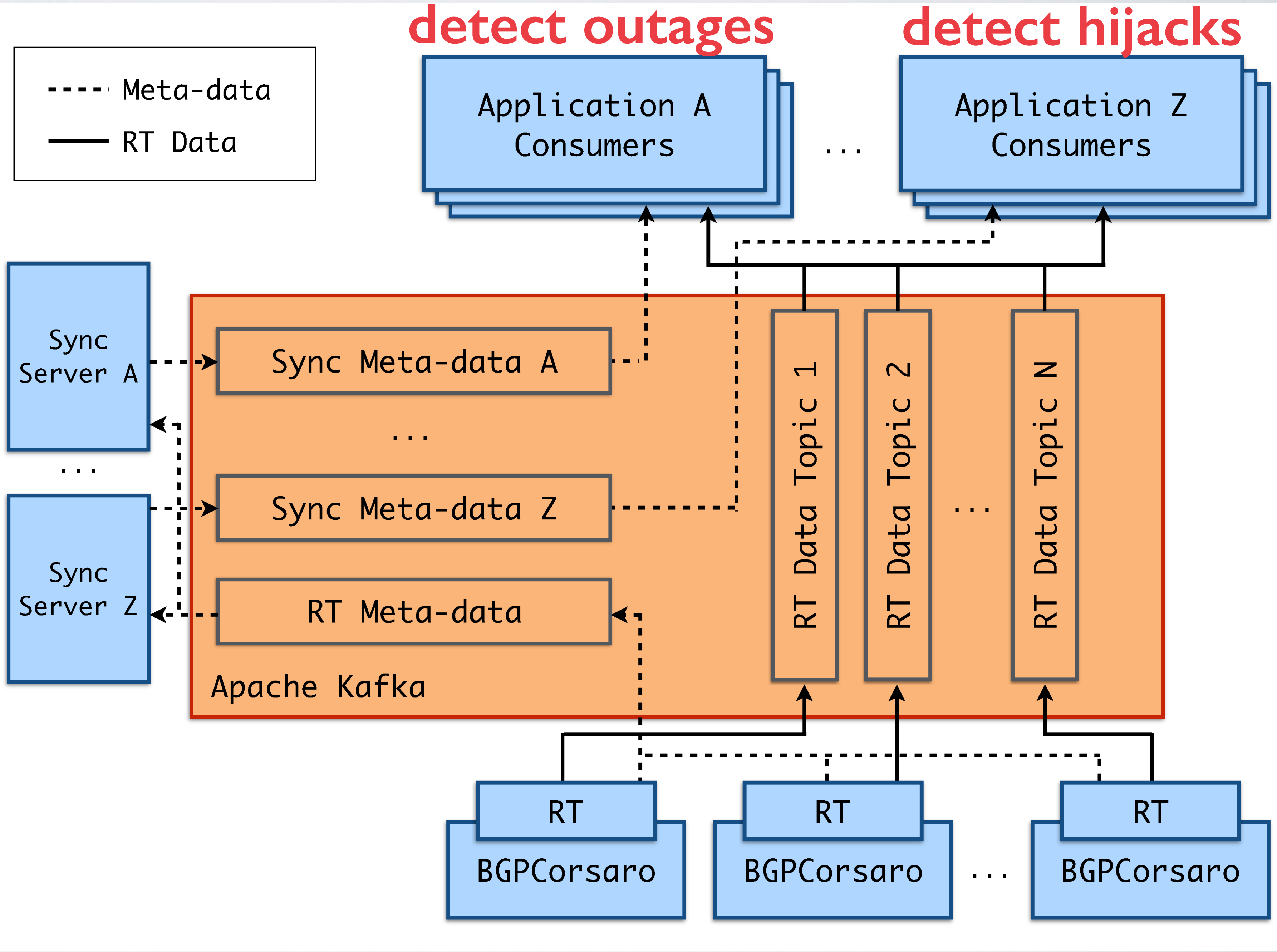
GLOBAL INTERNET MONITORING

how we built complex infrastructure enabling our projects

Live mode introduces the problem of sorting records from collectors that may publish data at variable times: trade-off between:

- *size of buffers*
- *completeness of data available to the application*
- *latency*

We solve this problem using Apache Kafka, Meta-data, and a Sync Server



THANKS

bgpstream.caida.org

alberto@caida.org