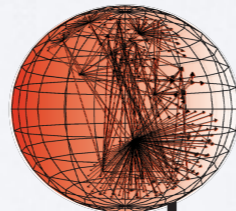


mPlane Workshop
30th Nov 2015, Heidelberg, DE

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

**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.

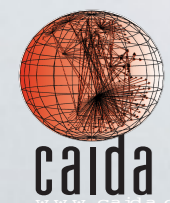
- **Performance**

- 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!**

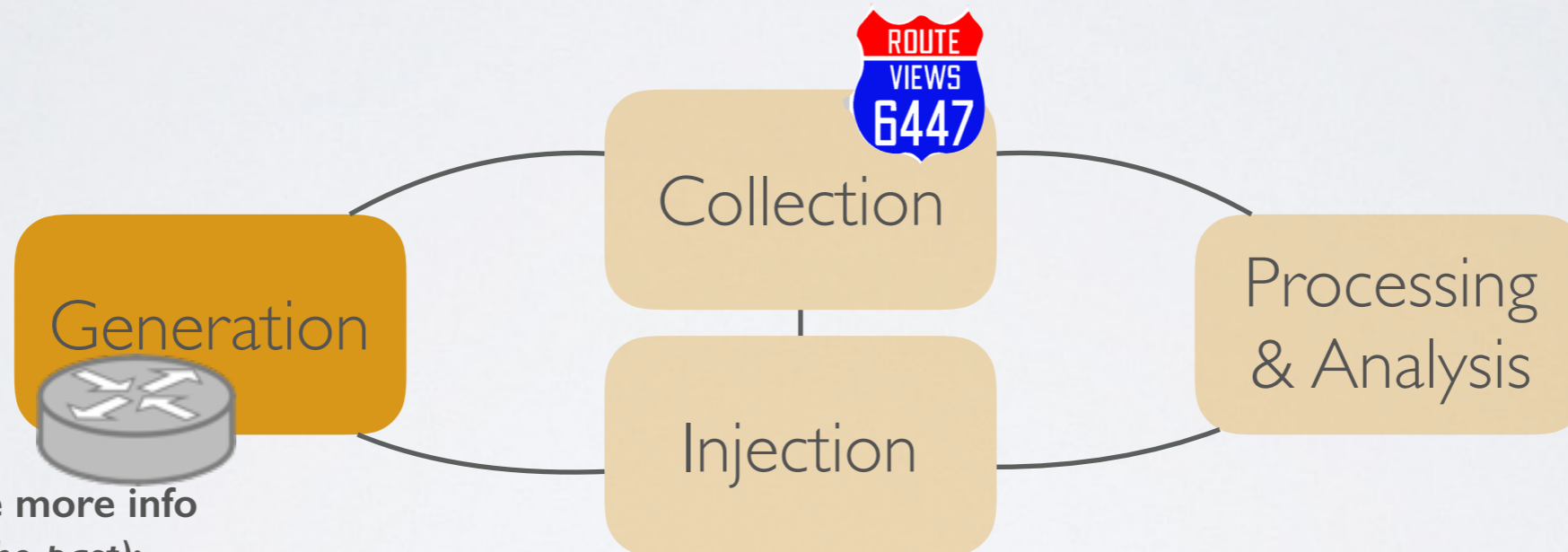
- AS-level topology
 - Gregori et al. “On the *incompleteness* of the AS-level graph: a novel methodology for BGP route collector placement”, IMC 2012
- AS relationships
 - Giotsas et al. “*Inferring* Complex AS Relationships”, IMC 2014
- AS interactions: driven by relationships, policies, network conditions, 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

I. 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):

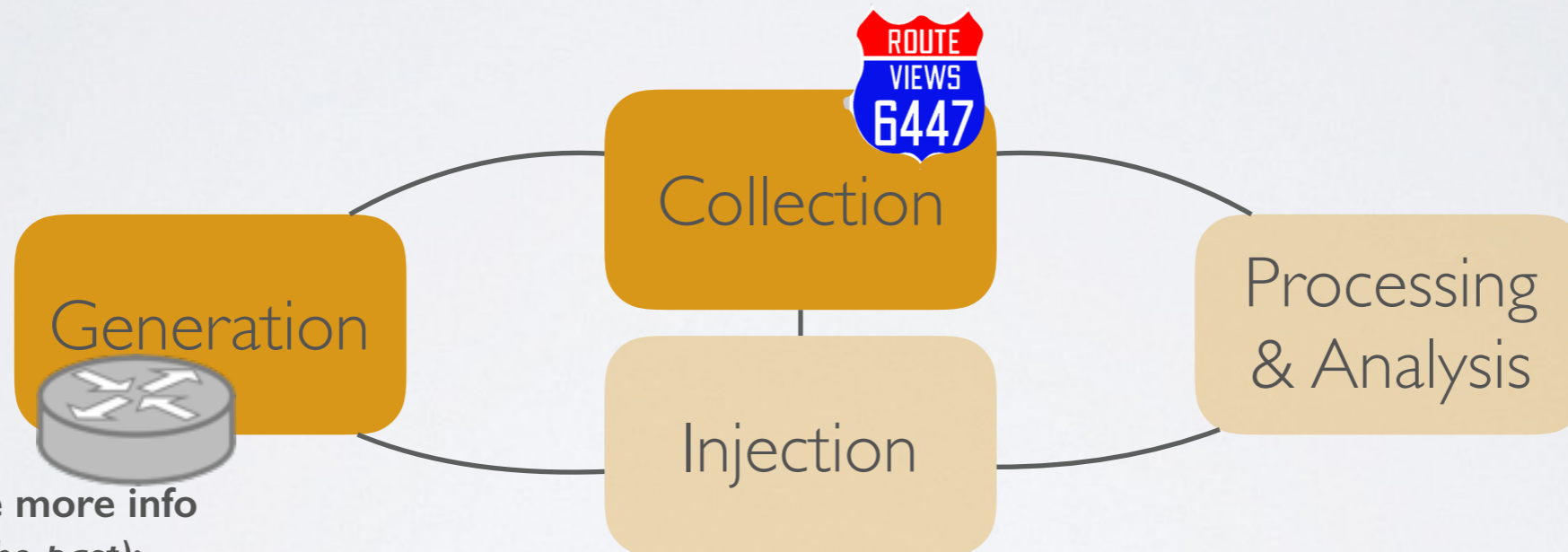
- RFC 4384 BGP Communities for Data Collection
- draft-ymbk-grow-bgp-collector-communities

MEASURING BGP

two issues - somehow related

I. 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):

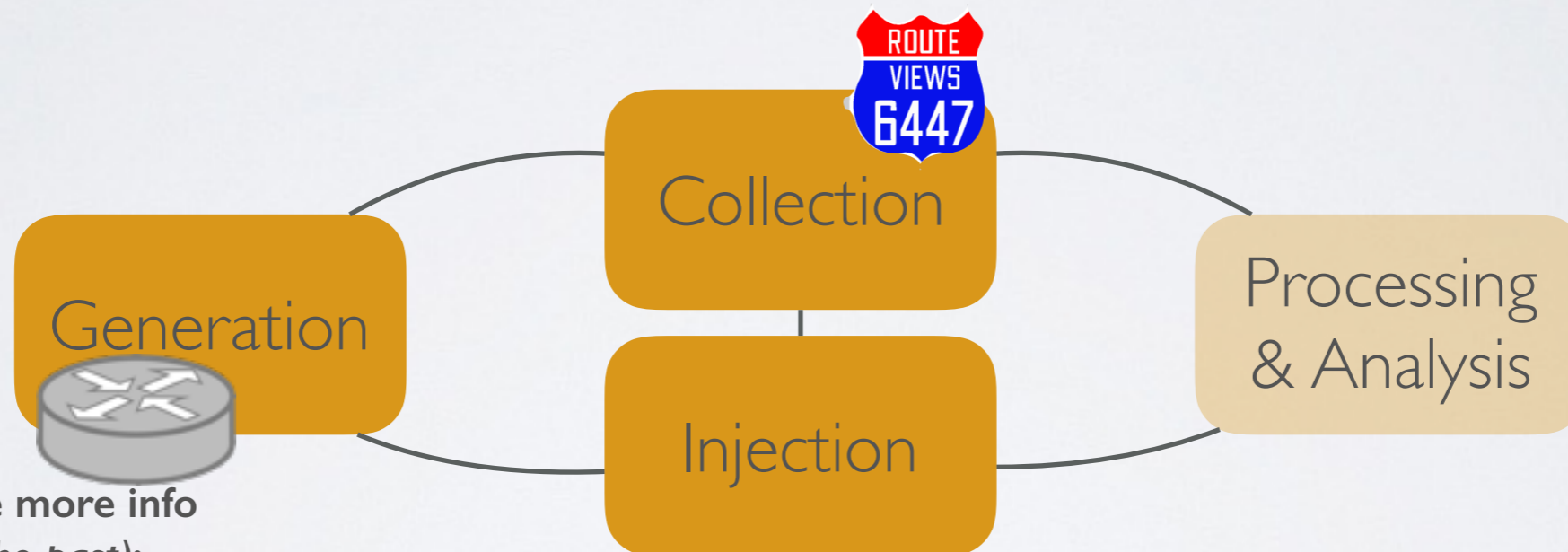
- RFC 4384 BGP Communities for Data Collection
- draft-ymbk-grow-bgp-collector-communities

MEASURING BGP

two issues - somehow related

I. 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

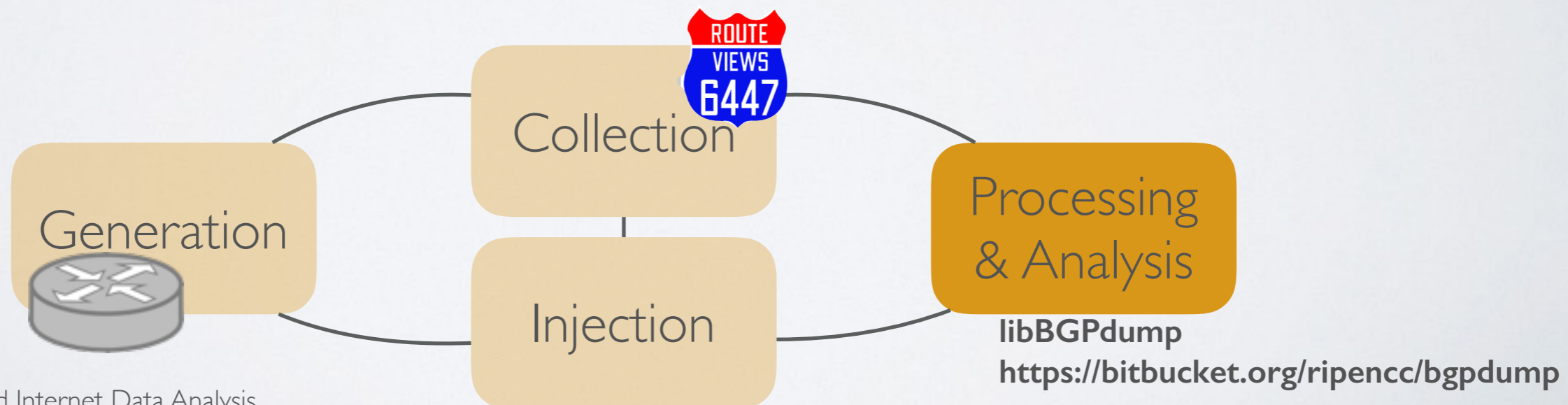
two issues - somehow related

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, ...



MEASURING BGP

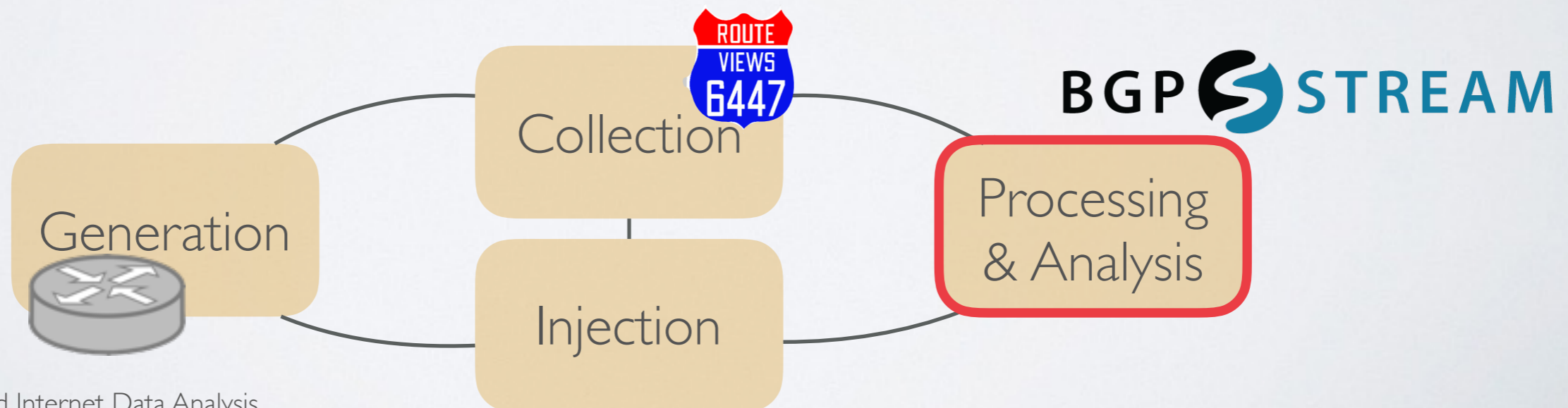
two issues - somehow related

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

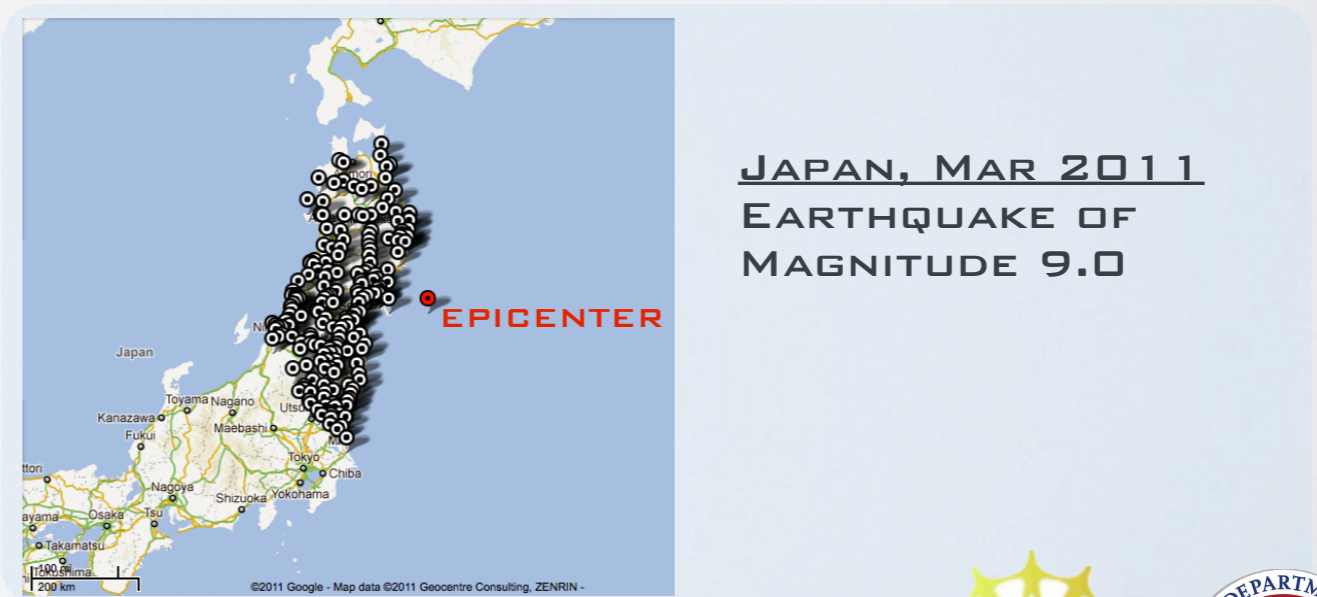
- Country-level Internet Blackouts during the Arab Spring

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



- Natural disasters affecting the infrastructure

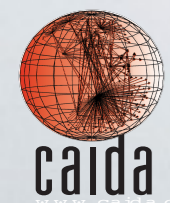
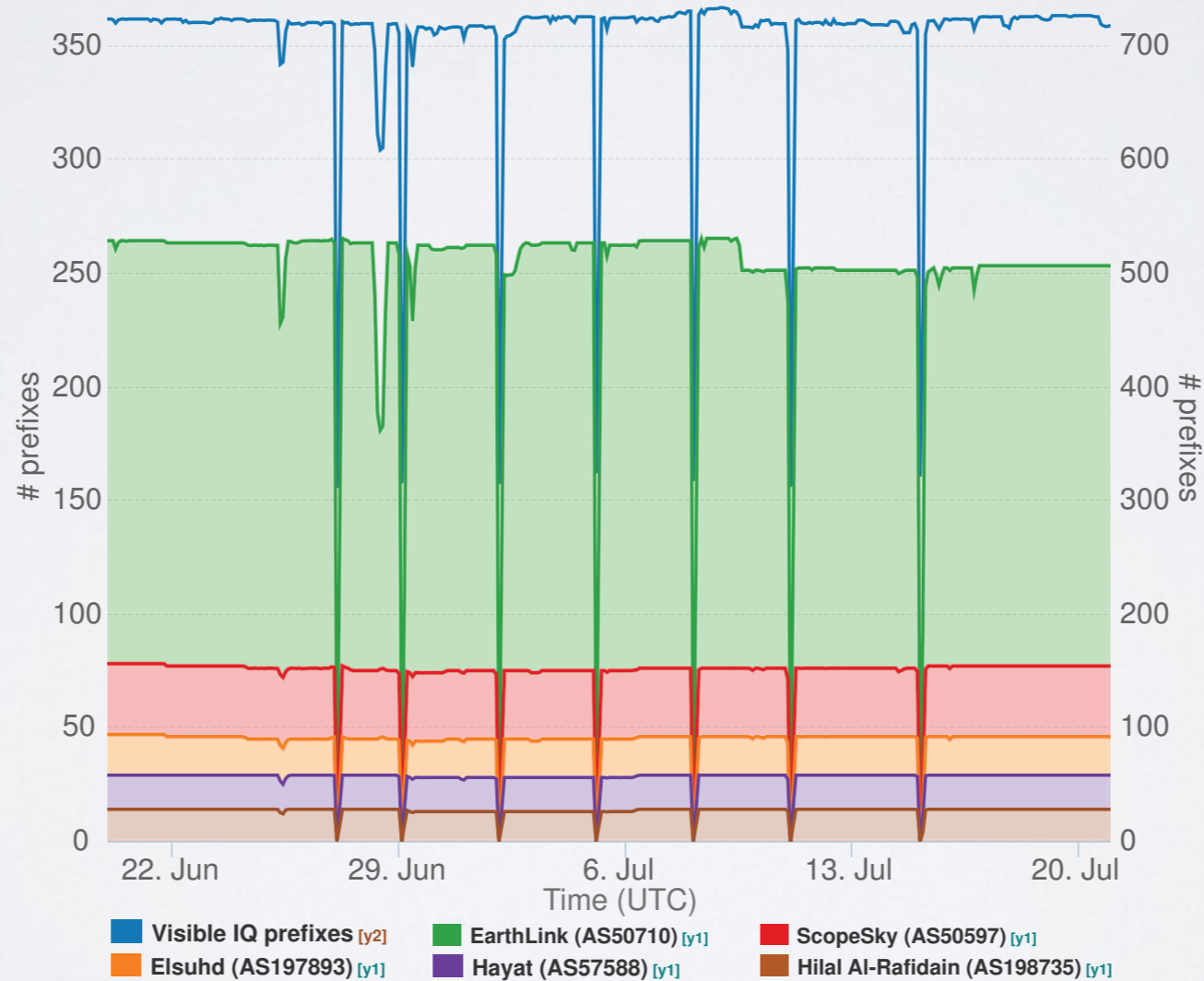
Dainotti et al. "Extracting Benefit from Harm: Using Malware Pollution to Analyze the Impact of Political and Geophysical Events on the Internet"
SIGCOMM CCR 2012



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



Center for Applied Internet Data Analysis
University of California San Diego

www.caida.org/funding/iodal/



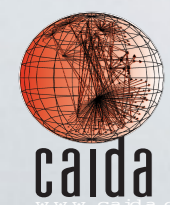
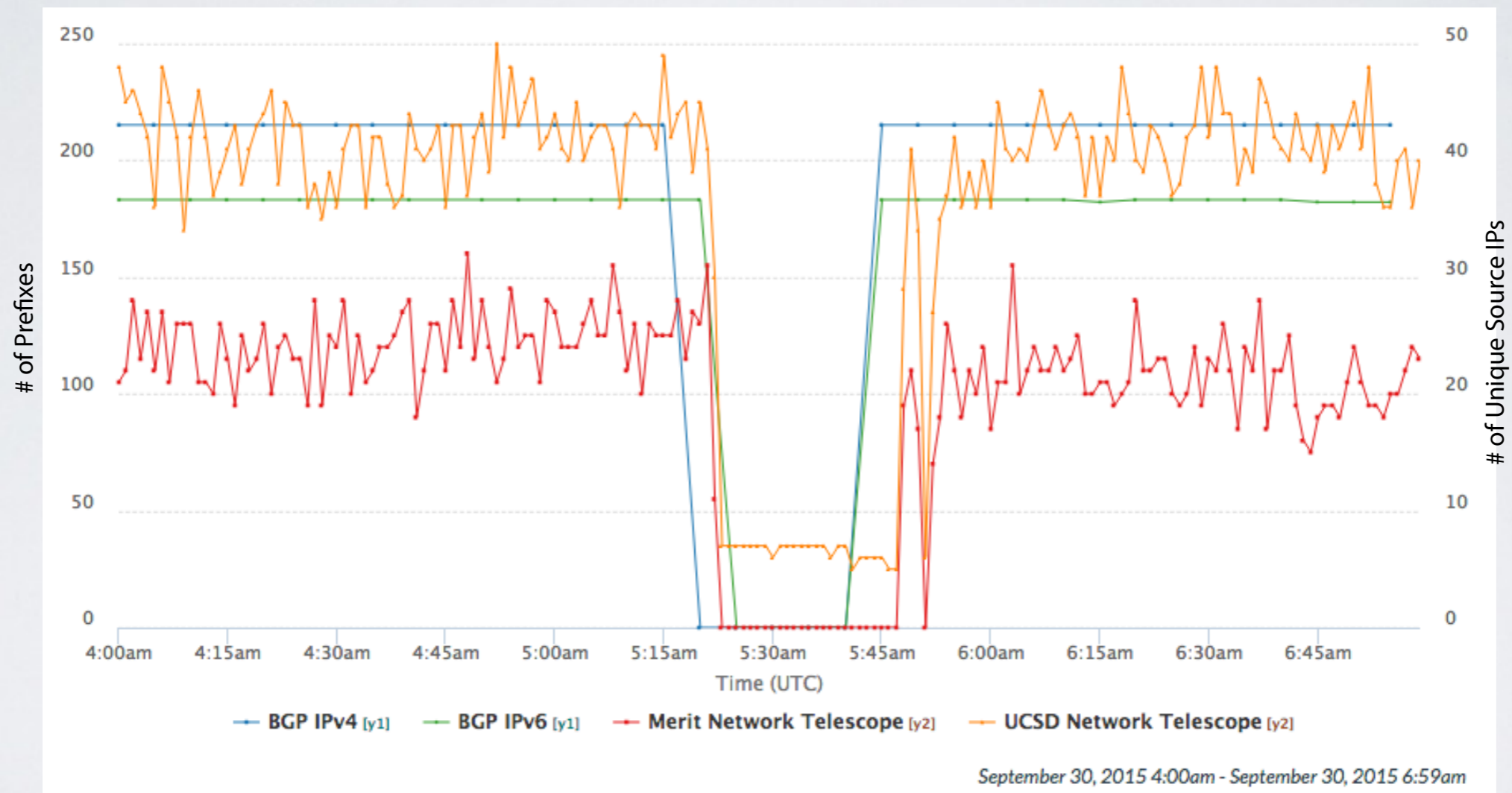
COMCAST



INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

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



Center for Applied Internet Data Analysis
University of California San Diego

www.caida.org/funding/iodal/

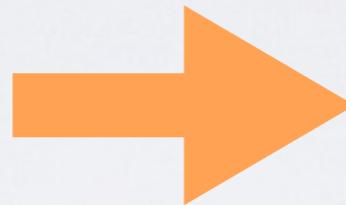


BEFORE IODA

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

ABSTRACT

In the first months of 2011, several North African countries and the states of the Middle East experienced a series of disruptions in their Internet connectivity. We analyze these events by studying network data, monitoring data, and traffic analysis to determine which network components were affected and to identify the attack vectors. We then analyze policies and ASes that control flow and data to narrow down which is the most likely cause. Our main findings are that Internet outages are often caused by a combination of geographic, strategic,

Categories and Settings

General Terms

Measurement, Security

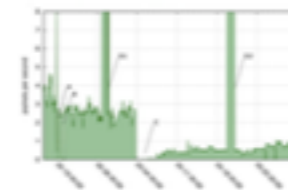


Figure 12: UCSD dataset's traffic coming from Libya. Labels A, B, C indicate the three outages. Spikes labeled D1 and D2 are due to backbone business denial of service attacks.

related to protests in the country. The web site of the Ministry of Communications (www.mcom.gov.ly) was attacked with a randomly-specified DoS attack just before the outage started, on January 26 at different times: 15:47 GMT (for 16 minutes), 16:55 GMT (17 minutes), and 21:09 GMT (53 minutes). Analysis of the backbone traffic to the Internet allows estimation of the intensity of the attack in terms of packet rate, indicating average packet rates between 30k and 50k packets per second.

5.2 Libya

5.2.1 Overview

Libya's Internet infrastructure is even more prone to manipulation than Egypt's, judging from its physical structure. International connectivity is provided by only two submarine cables, both ending in Tripoli [36], and the Internet infrastructure is dominated by a single, state-owned, AS. We only found two other ASes having a small presence in Libya, as described in Section 5.2.2.

In Libya three different outages in early 2011 were identified and publicly discussed (Figure 1). Figure 12 shows the traffic observed by the UCSD network telescope from Libya throughout an interval encompassing the outages. The points labeled A, B and C indicate three different blackout episodes; points D1 and D2 refer to two denial-of-service attacks discussed in Section 5.2.3. Toward the right of the graph it is difficult to interpret what is really happening in Libya because of the civil war.

5.2.2 Outages in detail

The first two outages happened during two consecutive nights. Figure 13(a) shows a more detailed view of these two outages as observed by the UCSD telescope. Figure 13(b) shows BGP data over the same interval: in both cases, within a few minutes, 12 out of the 15 IPv4 prefixes associated with IP address ranges officially delegated to Libya were withdrawn. These twelve IPv4 prefixes were announced by LyStataS, the local telecom operator, while the remaining IPv4 prefixes were managed by IBAAS2. As of May 2011, there were no IPv4 prefixes in AfriNIC's delegated file for Libya. The MaxMind IP geolocation database further puts 12 non-contiguous IP ranges in Libya, all part of an encompassing IPv4

prefix announced by SatAS, which provides satellite services in the Middle East, Asia and Africa. The covering IPv4 prefix also contained 180 IP ranges in several other countries, predominantly in the Middle East. We considered this additional AS because the UCSD dataset generally observed a significant amount of truncated traffic coming from IPs in these 12 ranges before the first outage (about 5k packets each day). This level of background traffic indicates a population of customers using PCs likely infected by Conficker or other malware, allowing inference of network conditions. Traffic from this network also provided evidence of what happened to Libyan Internet connections based on satellite systems not managed by the local telecom provider.

Comparing Figures 13(a) and 13(b) reveals a different behavior than conflicts with previous reports [17]: the second outage was not entirely caused by BGP withdrawals. The BGP shutdown began on February 19 around 23:58:55 UTC, exactly matching the sharp decrease of distinct traffic from Libya (and in accordance with reports on Libyan traffic seen by Aben Networks [34]); but it ended approximately one hour later, at 23:02:52. In contrast, the Internet outage as shown by the telescope data and reported by the news [17] lasted until approximately February 20 at 6:12 UTC. This finding suggests that a different disruption technique – a packet-blocking strategy apparently adopted subsequently in the third outage and recognized by the rest of the world – was already being used during

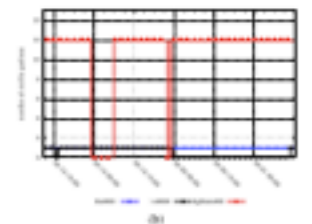
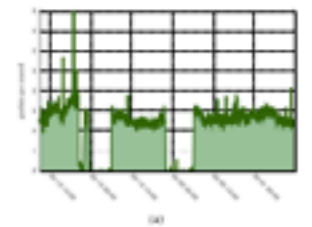
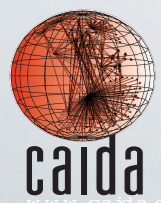


Figure 13: The first two Libyan outages: (a) unaltered traffic to UCSD dataset coming from Libya; (b) visibility of Libyan IPv4 prefixes in BGP data from Rome/Aben and RIPE NCC/RD collectors. Note that the corresponding satellite plane observations of connectivity do not match, suggesting that different techniques for censorship were being used during different outages.

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 TODAY

live Internet monitoring



Last Christmas we made it possible for anybody to follow the North Korean disconnection almost live

CAIDA @caidaorg · Dec 23

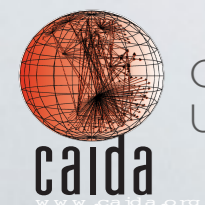
Follow outages in #NorthKoreaInternet in almost real-time (30min delay) at [charthouse.caida.org/public/kp-outa...](https://charthouse.caida.org/public/kp-outage)

Dec 21 2014 → Now
Visible BGP Prefixes

caida

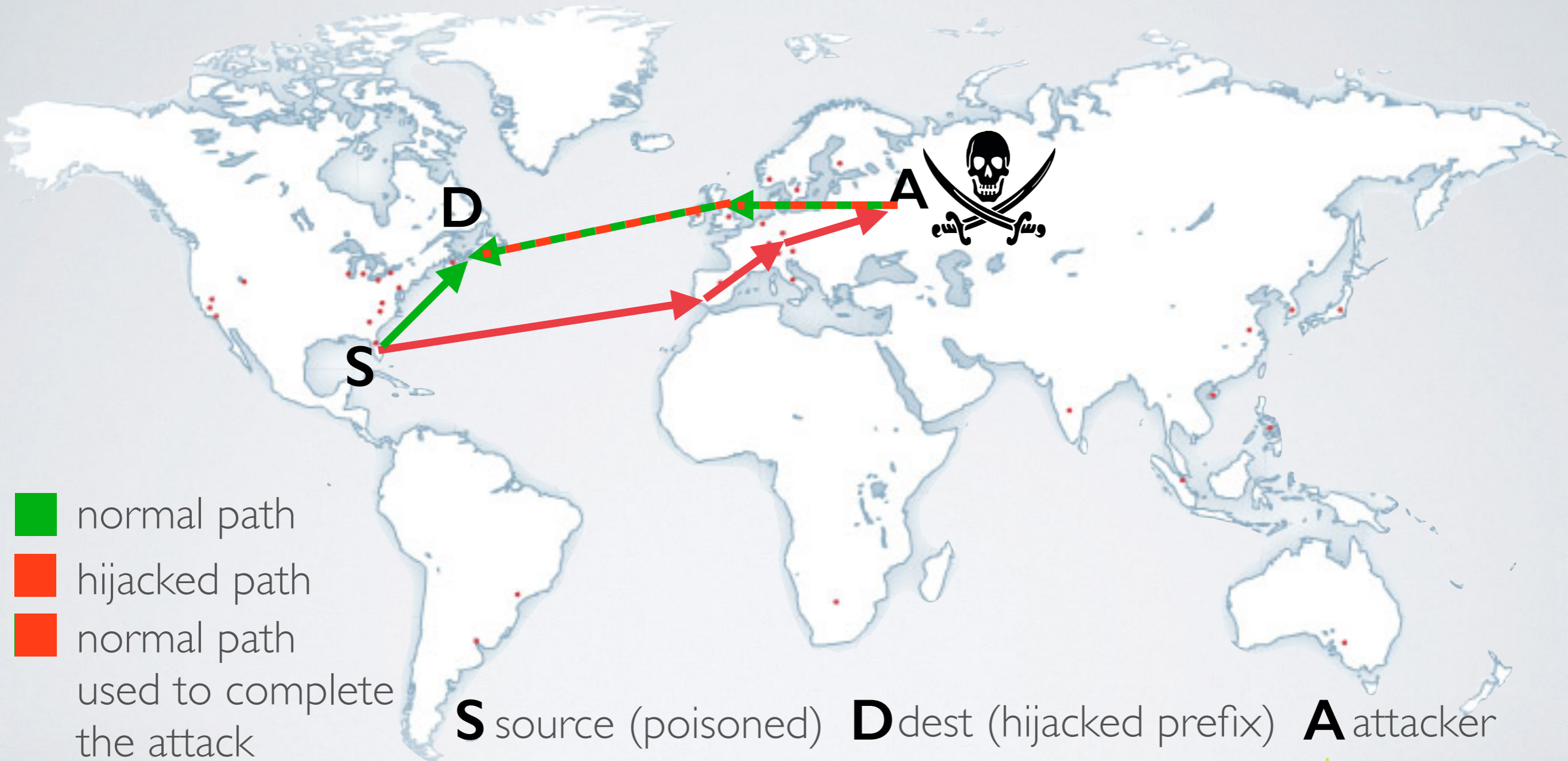
4pm 22. Dec 8am 4pm 23. Dec 8am 4pm

← ↻ 3 ★ 4 ... View more photos and videos



INSPIRING PROJECTS (2/2)

Hijacks: detection of MITM BGP attacks



BGP STREAM

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

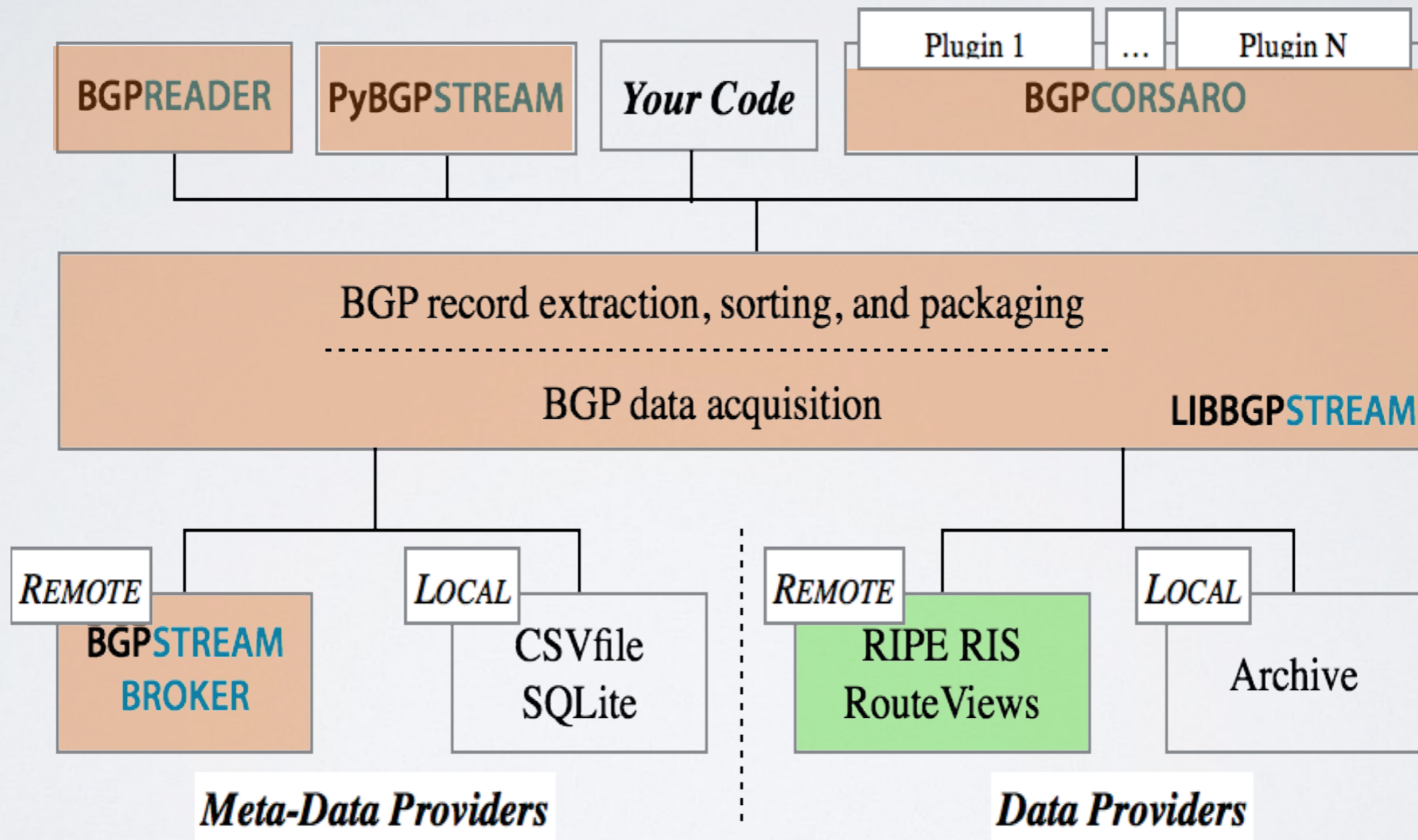
BGP STREAM

it's real!

- bgpstream.caida.org
 - download it! (version 1.0)
 - active development - github.com/caida/bgpstream
 - Docs & Tutorials
- paper under submission at NSDI '16 (tech report on web site)
- people are using it!
- coordination with RouteViews, Colorado State BGPMon, RIPE NCC
- BGP Hackathon in February

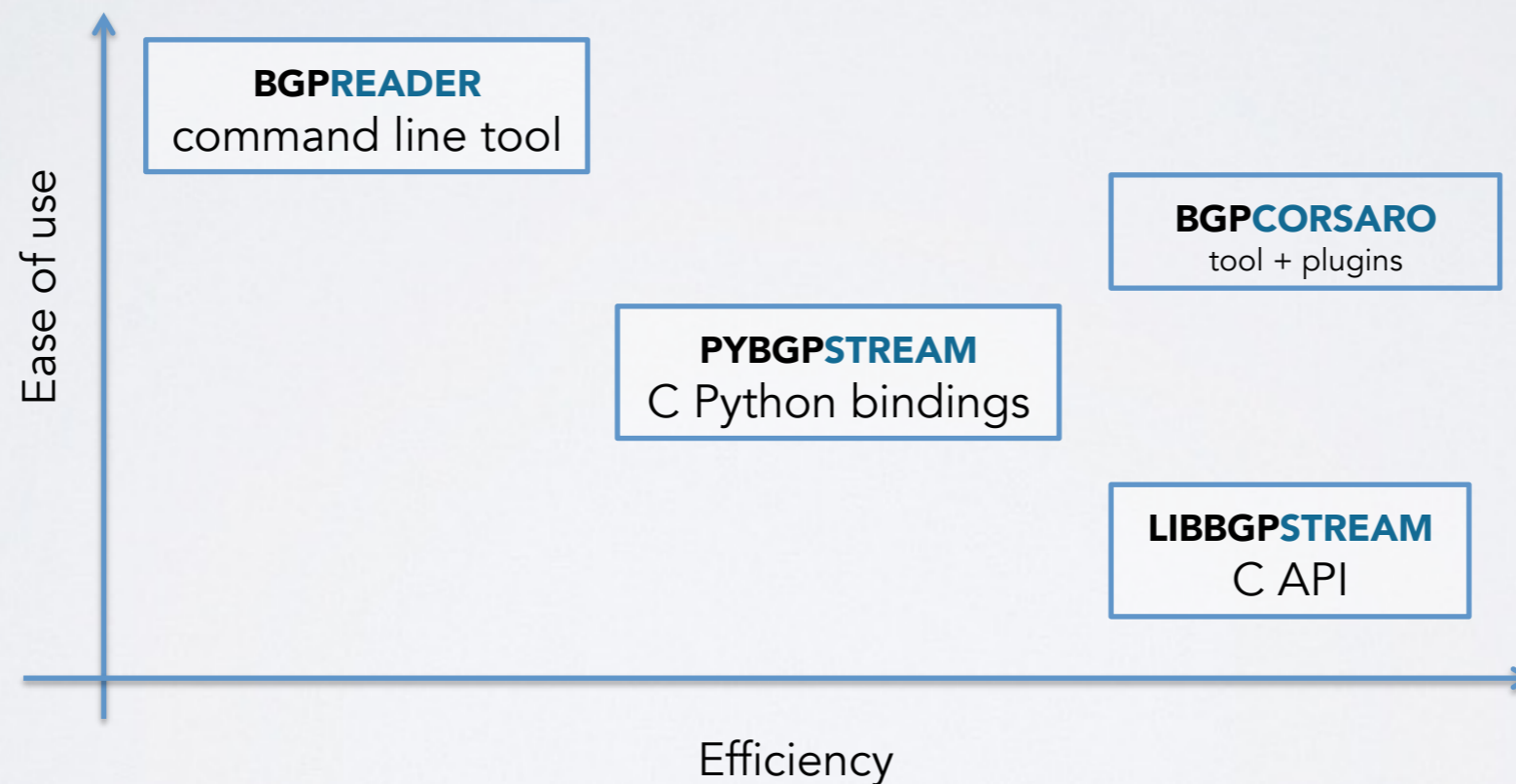
BGP STREAM

overview



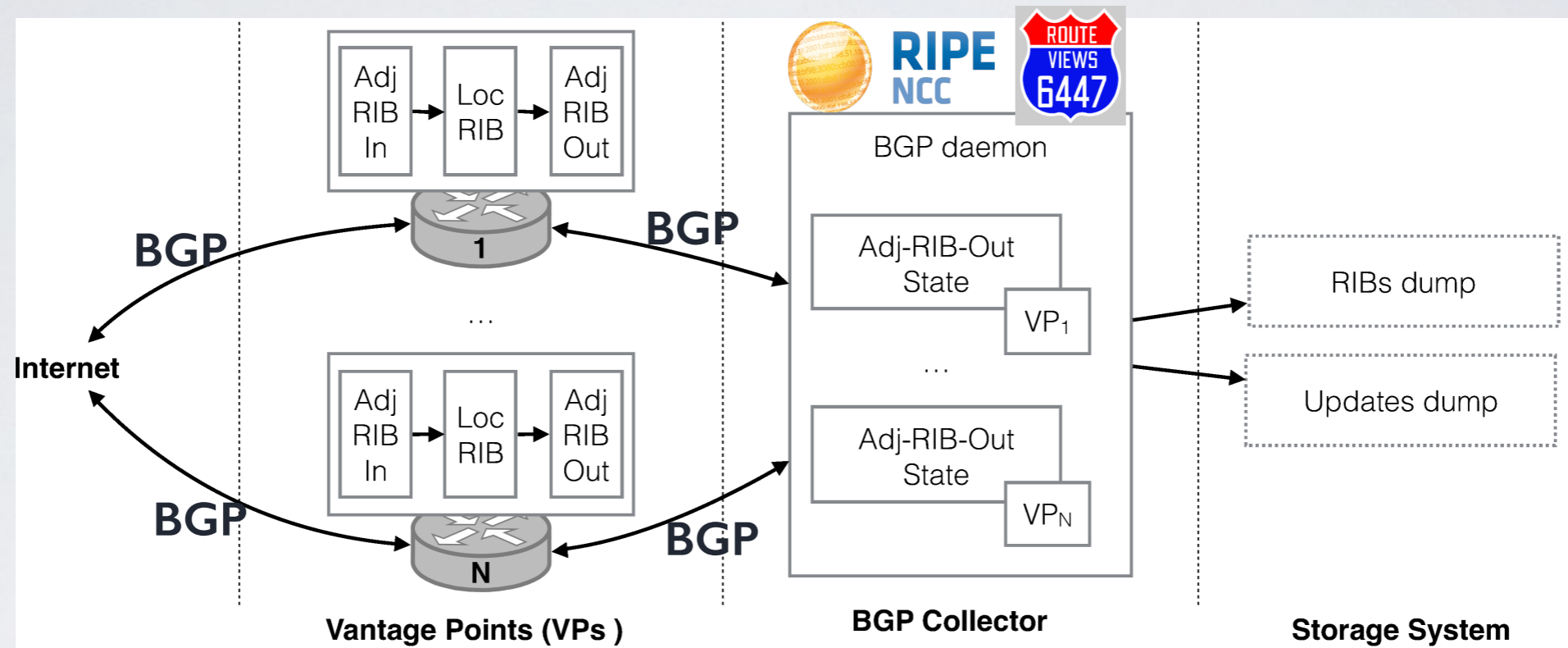
BGP STREAM

different applications and development paradigms



TERMINOLOGY

background and naming conventions



- Adj-RIB-Out etc. [RFC 4271]
- Collectors: RIB and Updates dumps
- VPs
- Partial vs Full-feed VPs
- ...

FOR COLLECTED DATA

overview

- RouteViews and RIPE RIS collectors (~31) save:
 - RIB dumps every 2 and 8 hours
 - Updates dumps every 15 and 5 minutes
- a full-feed VP (in 2015)
 - has a an Adj-RIB-Out with ~550k routes
 - generates ~1.5K updates every 5 minutes
- RIB and Updates dumps are saved in the Multi-Threaded Routing Toolkit (MRT) binary format [RFC6396]
 - 10KB - 100MB for RIB dumps (compressed)
 - 1KB - 10MB for Updates dumps (compressed)
- RouteViews and RIPE RIS archives date back to 2001 and 1999 respectively
- The full archives of compressed files are about 8.9TB and 3.7TB, currently growing at the rate of **2TB per year**

LIBBGPSTREAM API

BGP data stream

- BGP data stream: *<collector projects (e.g., Route Views, RIPE RIS), list of collectors, dump types (RIB/Updates), time interval start and either time interval end or live mode>*.
- A stream can include dumps of different type and from different collector projects.
- A stream is made of *BGP records*, which can be decomposed in *BGP elems*

LIBBGPSTREAM PULL MODEL

based on the Broker

- the library implements a “*client pull*” model
 - efficient data retrieval without potential input buffer overflow (i.e., data is only retrieved when the user is ready to process it)
 - supports live mode
- iteratively alternates between:
 - meta-data queries to the Broker
 - and opening and processing the returned data
- *historical mode*: the stream ends when the Broker returns an empty set
- *live mode*: the query mechanism is blocking. If the Broker has no data available, a polling cycle will begin, periodically re-issuing the request to the Broker

C API

specifying a 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

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*)

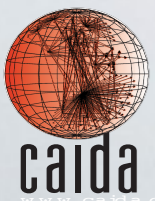
Field	Type	Function
type	enum	route from a RIB dump, announcement, withdrawal, or state message
time	long	timestamp of MRT record
peer address	struct	IP address of the VP
peer ASN	long	AS number of the VP
prefix*	struct	IP prefix
next hop*	struct	IP address of the next hop
AS path*	struct	AS path
old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

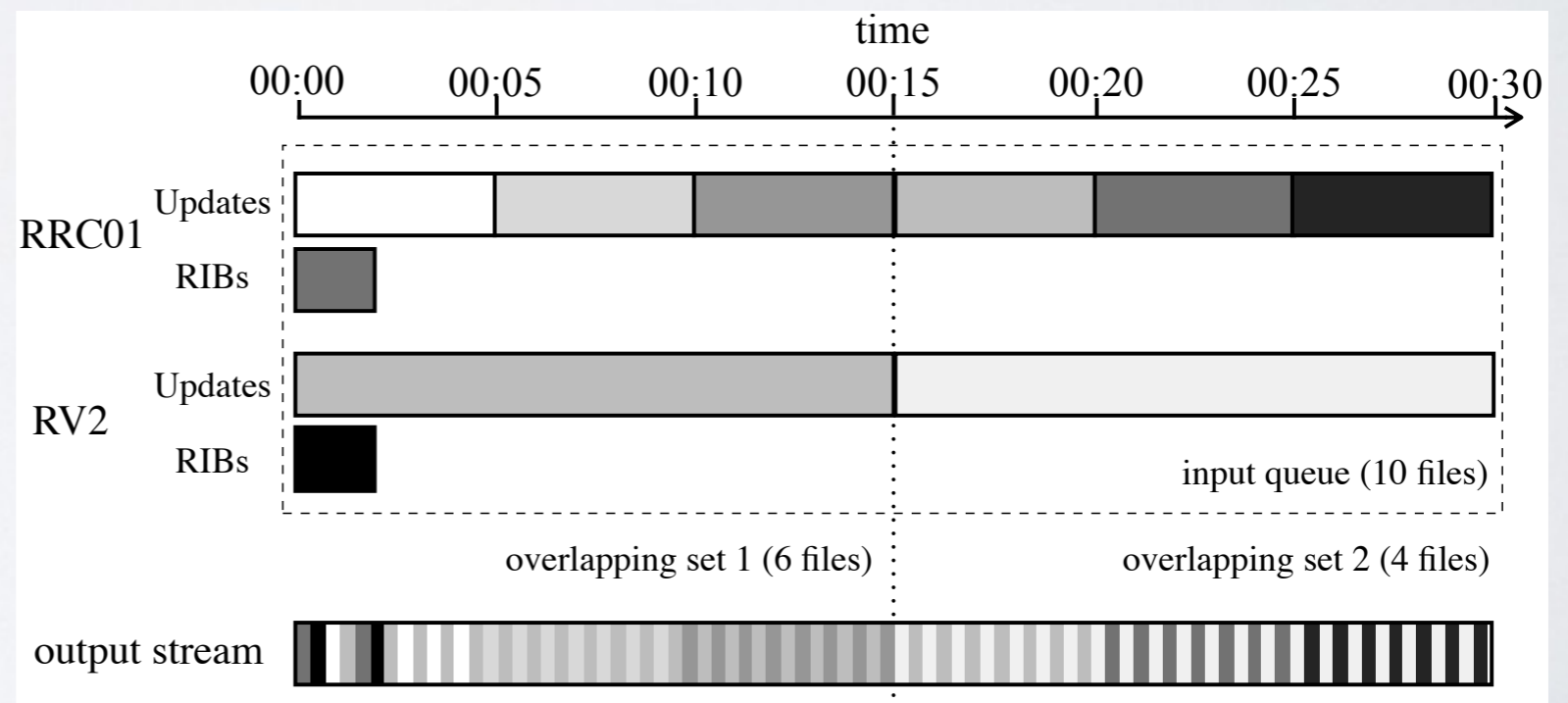
```
/* Start the stream */ 21
bgpstream_start(bs); 22
23
/* Read the stream of records */ 24
while (bgpstream_get_next_record(bs, record) > 0) { 25
    /* Ignore invalid records */ 26
    if (record->status != BGPSTREAM_RECORD_STATUS_VALID_RECORD) { 27
        continue; 28
    } 29
    /* Extract elems 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
```



RECORD-LEVEL SORTING

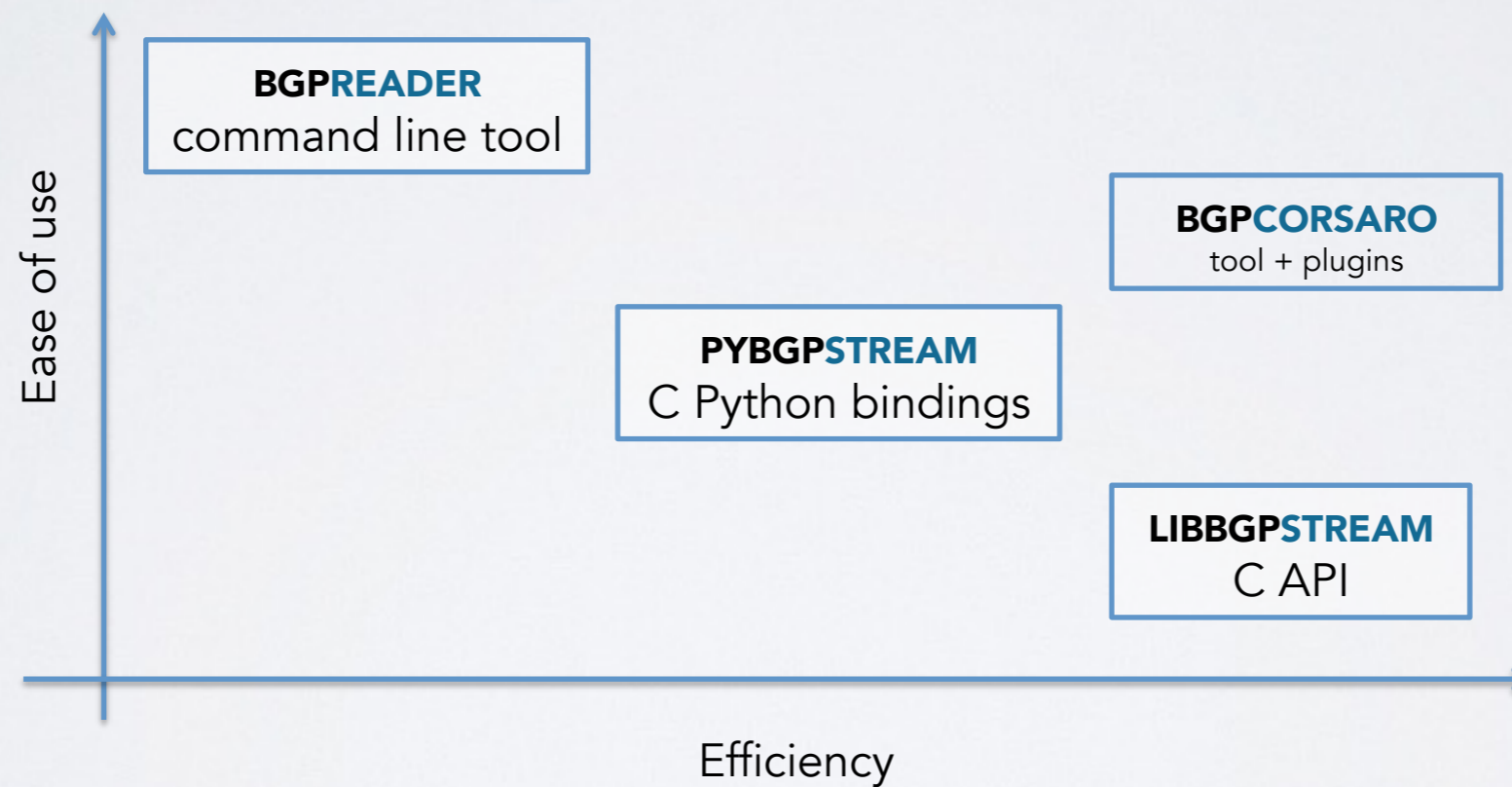
When

- When:
 - when reading dumps from more than one collector (*inter-collector sorting*)
 - when a stream is configured to include both RIB and Updates dumps (*intra-collector sorting*)



TOOLS/APIs

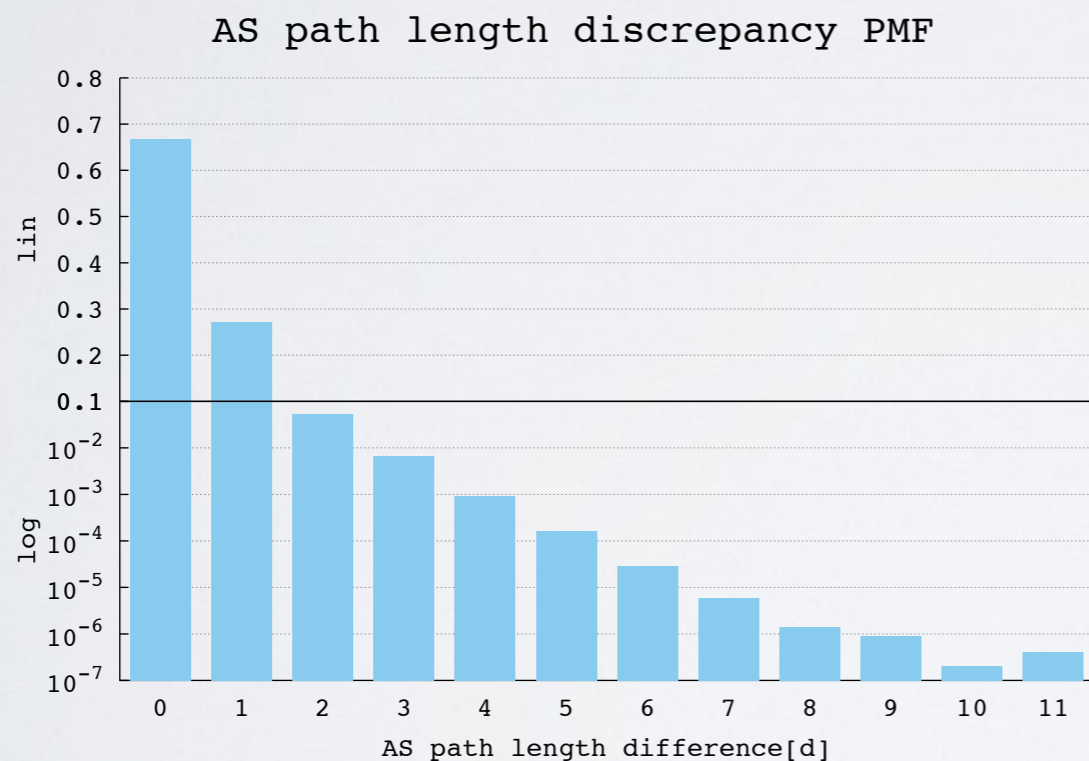
continued..



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*)



```
1 from _pybgpstream import BGPStream, BGPRecord, BGPElem
2 from collections import defaultdict
3 from itertools import groupby
4 import networkx as nx
5
6 stream = BGPStream()
7 as_graph = nx.Graph()
8 rec = BGPRecord()
9 bgp_lens = defaultdict(lambda: defaultdict(lambda: None))
10 stream.add_filter('record-type', 'ribs')
11 stream.add_interval_filter(1438415400, 1438416600)
12 stream.start()
13
14 while(stream.get_next_record(rec)):
15     elem = rec.get_next_elem()
16     while elem:
17         monitor = str(elem.peer_asn)
18         hops = [k for k, g in groupby(elem.fields['as-path'].split(" "))
19                 if len(hops) > 1 and hops[0] == monitor)
20         origin = hops[-1]
21         for i in range(0, len(hops)-1):
22             as_graph.add_edge(hops[i], hops[i+1])
23             bgp_lens[monitor][origin] = \
24                 min(filter(bool, [bgp_lens[monitor][origin], len(hops)]))
25         elem = rec.get_next_elem()
26 for monitor in bgp_lens:
27     for origin in bgp_lens[monitor]:
28         nxlen = len(nx.shortest_path(as_graph, monitor, origin))
29         print monitor, origin, bgp_lens[monitor][origin], nxlen
```

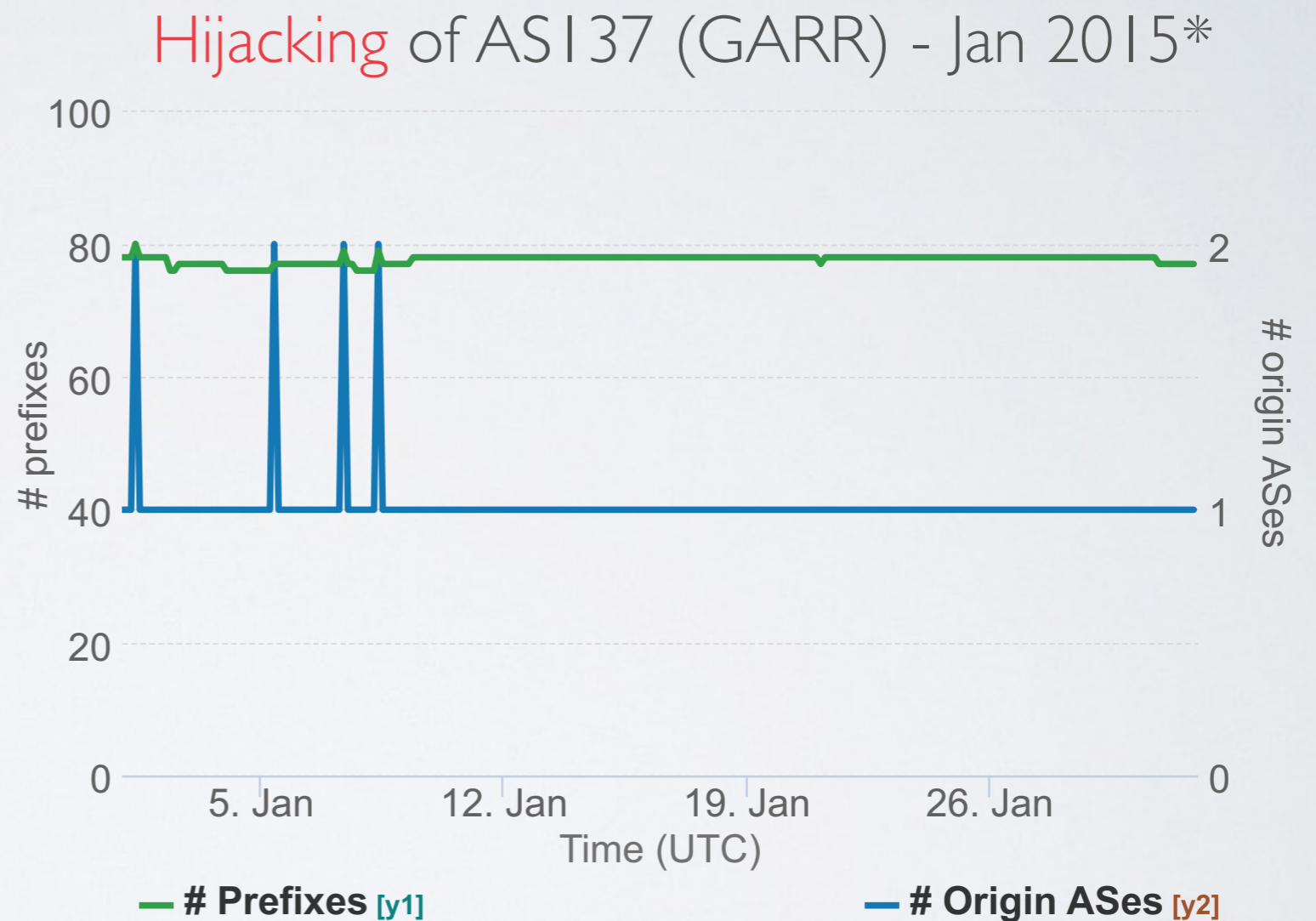
30 LINES OF
PYTHON CODE

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



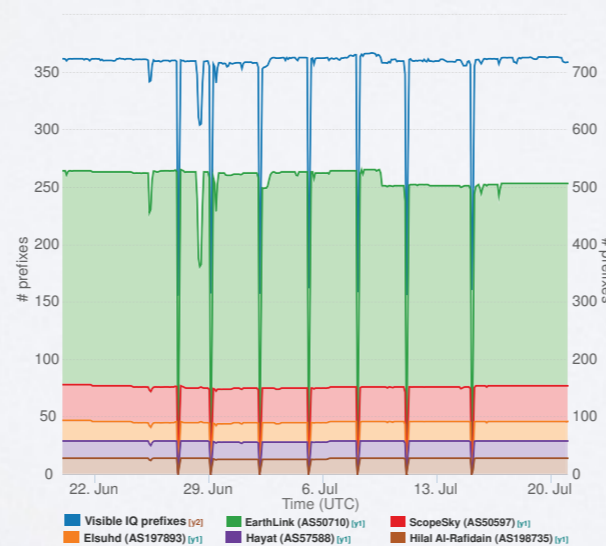
*Originally discovered by Dyn:

<http://research.dyn.com/2015/01/vast-world-of-fraudulent-routing/>

GLOBAL MONITORING

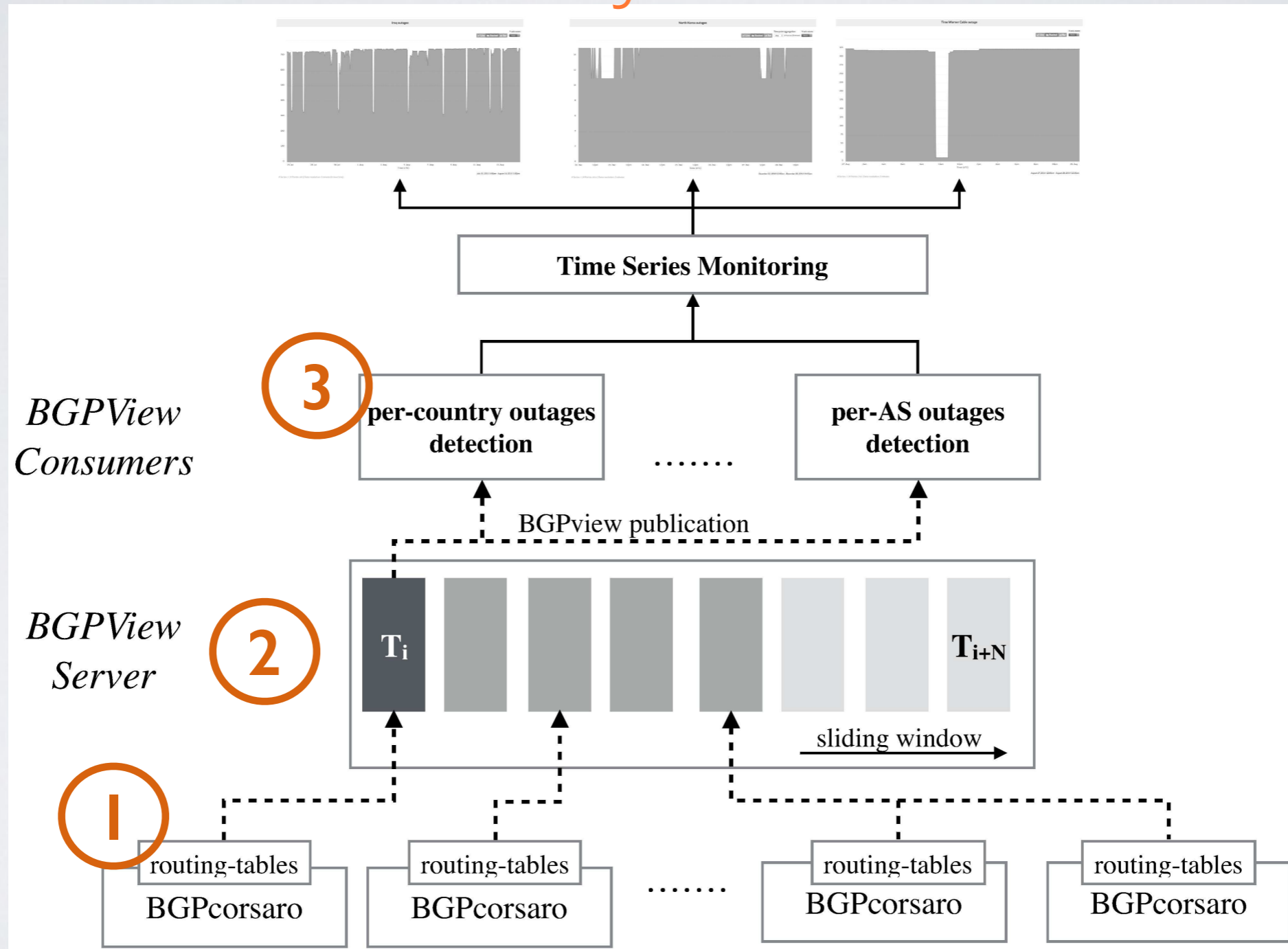
IODA, HIJACKS, etc.

- need to maintain a live **global view** (i.e., for each and every VP) of BGP reachability information updated with **fine time granularity** (e.g., few minutes)
- We implement 3 mechanisms:
 1. A solution to accurately reconstruct the observable LocRIB of each VP
 2. A synchronization mechanism
 3. Analysis modules to manipulate data from a BGP view



GLOBAL MONITORING

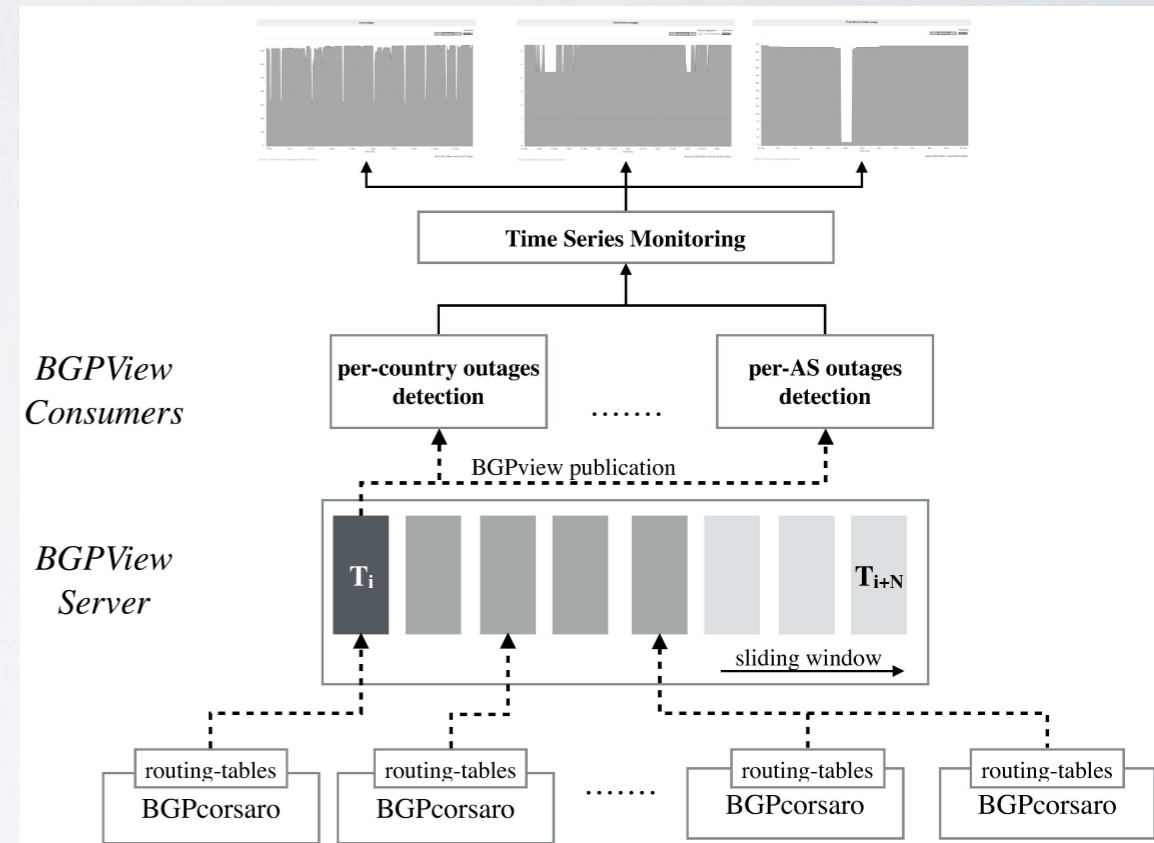
IODA, HIJACKS, etc.



BGPVIEWSERVER

buffering partial/complete BGP views

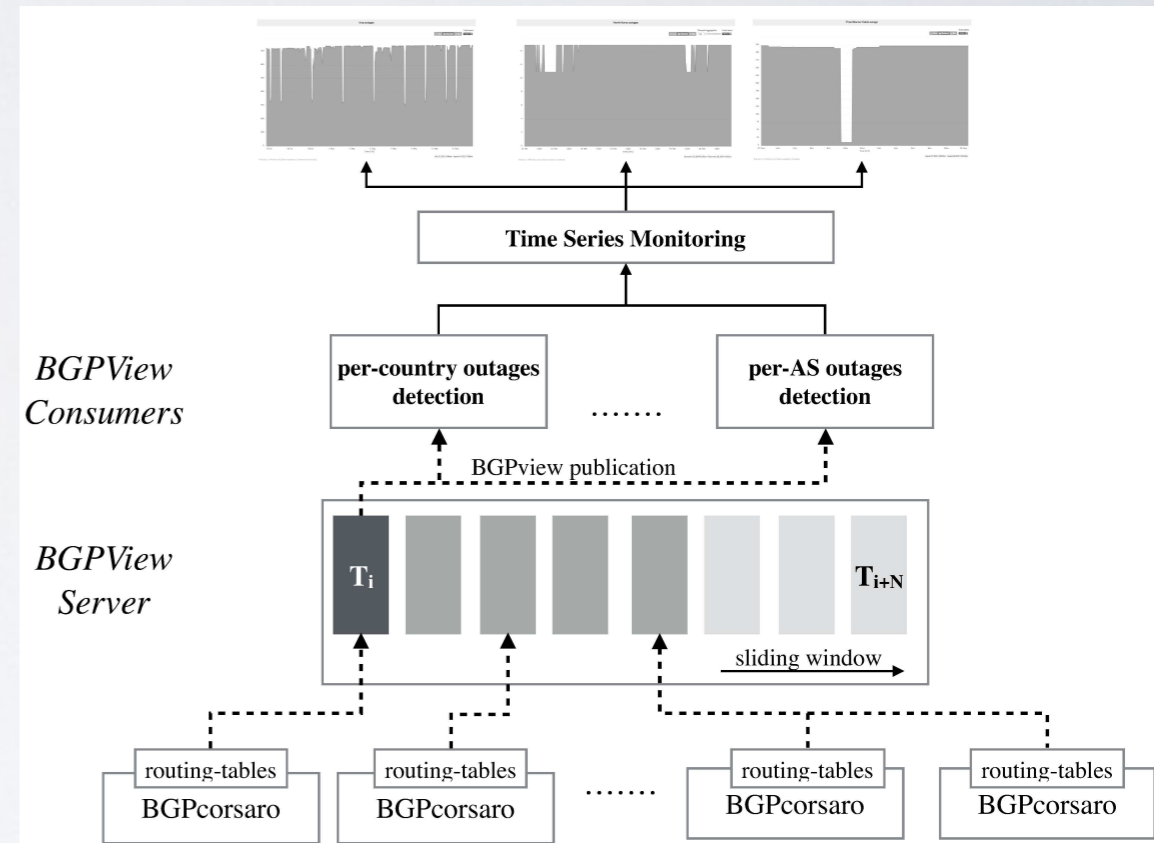
- At the end of a 1-minute time bin, each BGP Corsaro instance pushes data (the reconstructed routing table) to the BGPViewServer
- Such data is merged into a **partial** BGP view corresponding to its time bin
- A BGP view is considered **complete** when all the BGP Corsaro instances have contributed to it



BGPVIEWSERVER

sliding window

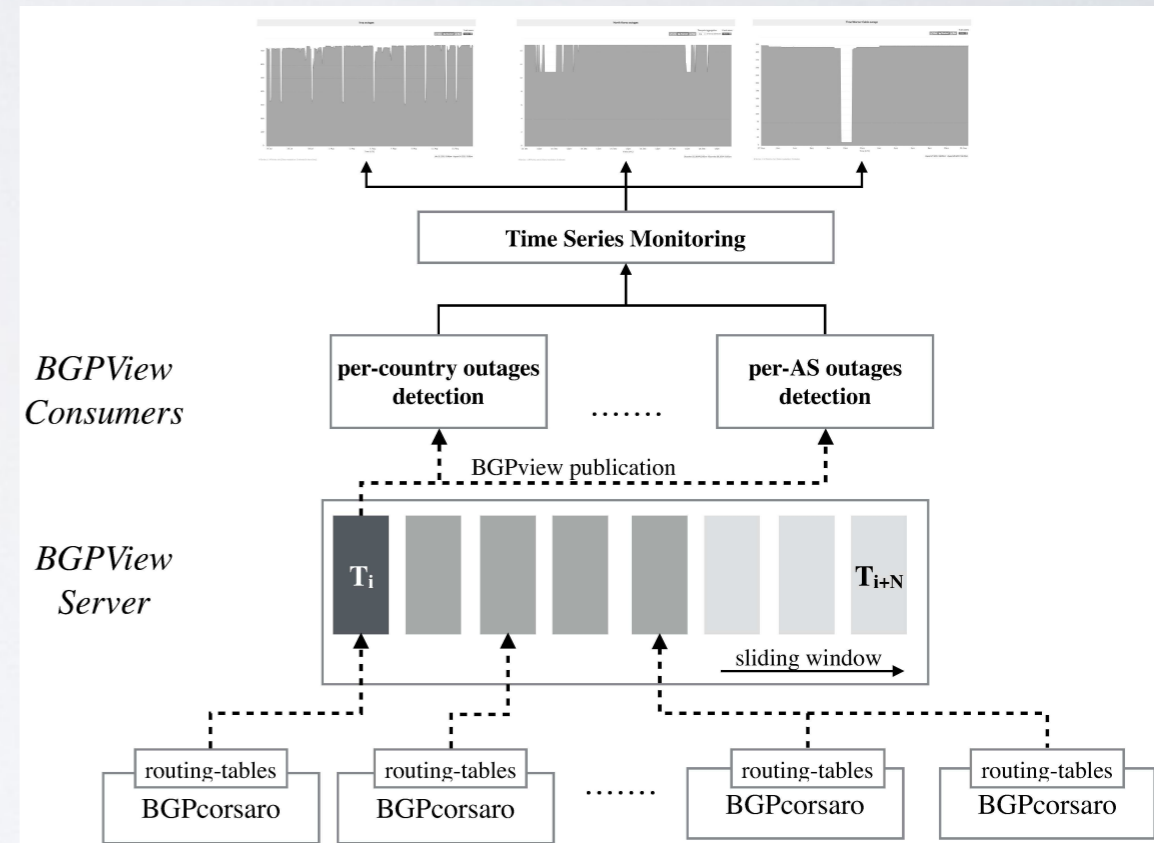
- we buffer partial BGP views in a **sliding window** based on their time bins
- the window slides each time data from a new bin arrives
- we publish a BGP view either
 - when all the BGP Corsaro instances have contributed to it (*complete view*)
 - or when it expires, i.e., its time bin is no longer covered by the window (*partial view*)



BGPVIEWSERVER

dimensioning

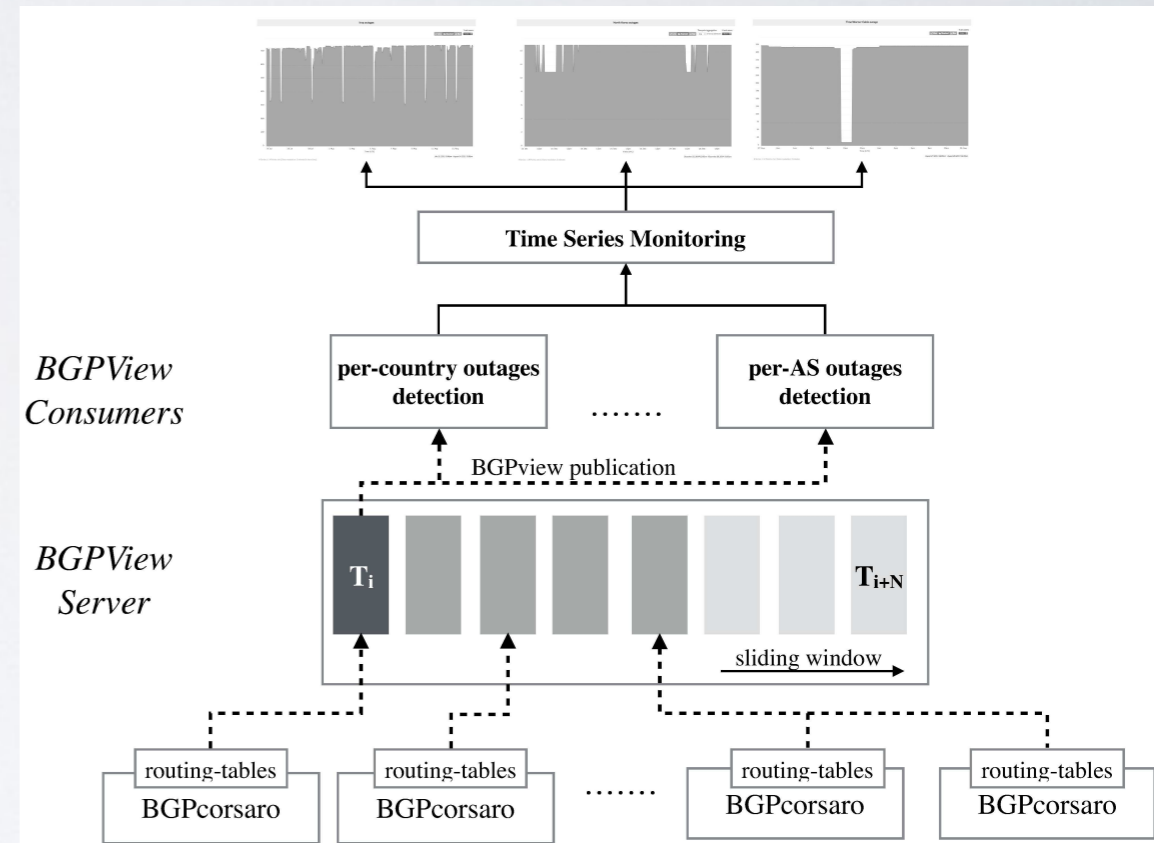
- We **dimension the length of the sliding window** empirically (12 months observation of RV+RIS)
 - the *latency* at which data providers publish dumps
 - the *memory footprint*
 - when processing data from all Route Views and RIPE RIS collectors, a **30 minute** sliding-window buffer requires **≈60GB** of memory and causes **99%** of BGP views to be published because they are complete rather than expired



BGPVIEWSERVER

bottleneck?

- The BGPViewServer is a potential bottleneck
 - # collectors grows \rightarrow increase in the amount of data that the server must receive, process and publish every minute
 - we architected the server to process each time bin independently of others
 - multiple server instances can be run (e.g., on separate hosts), with BGPCorsaro processes distributing data amongst them in a round-robin fashion.



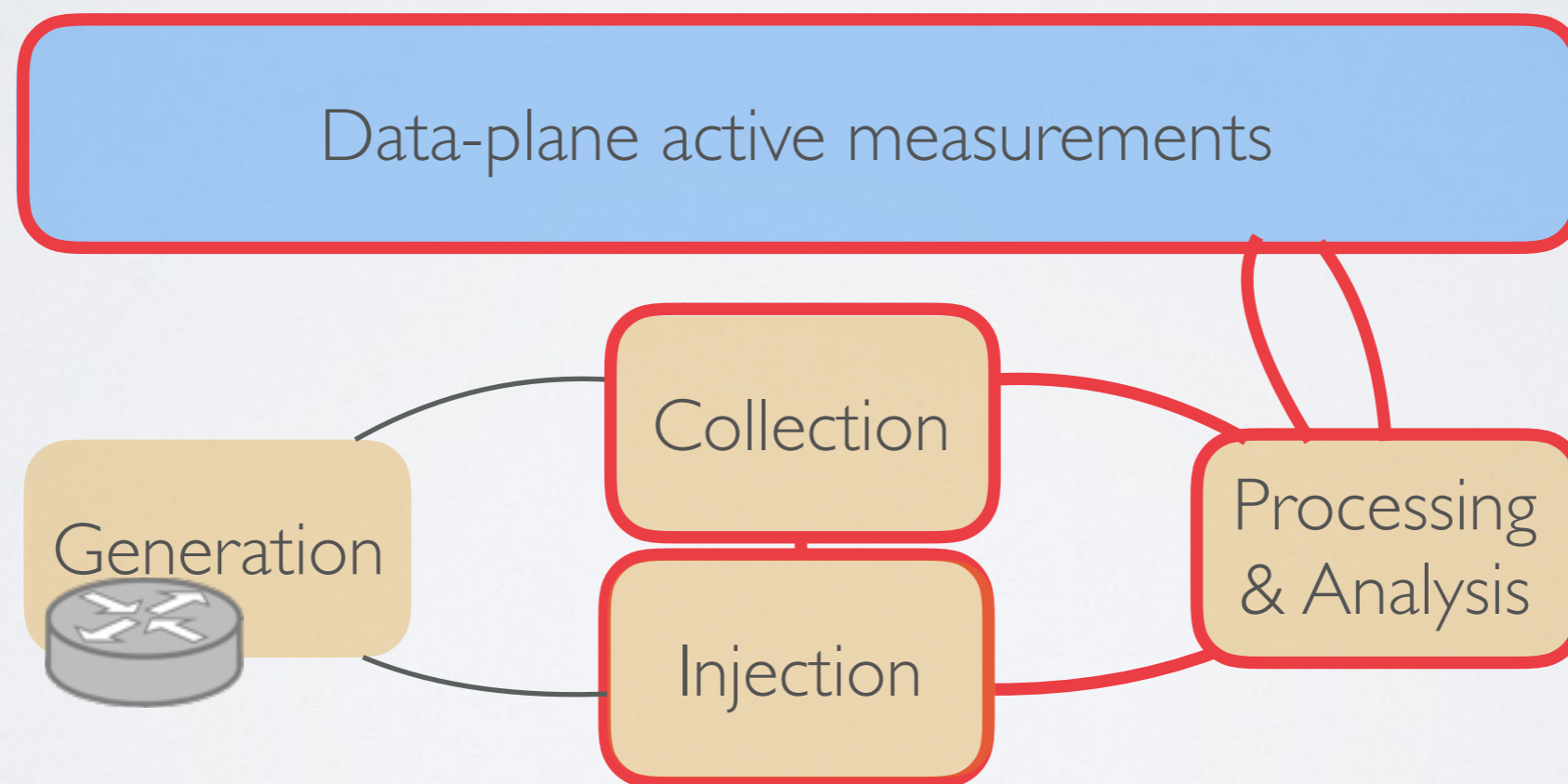
BGPVIEW CONSUMERS

demo on the browser

BGP HACKATHON - FEB 2016

theme: “**live** BGP measurements & monitoring”

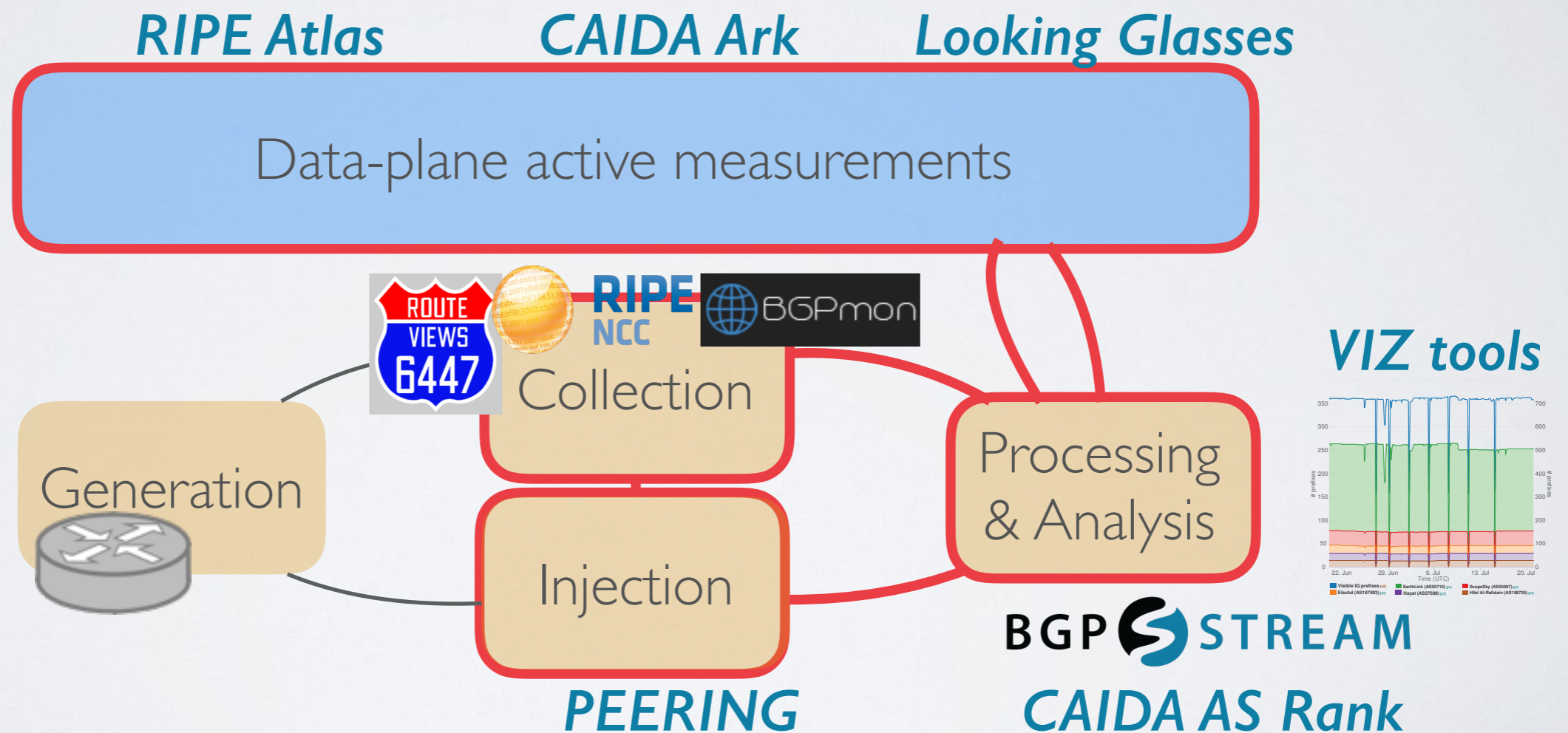
Improve/Integrate tools to study the BGP eco-system. Target practical problems: topology, hijacks, outages, RPKI deployment, path inflation, circuitous paths, policies, relationships, visualize dynamics, ...



BGP HACKATHON - FEB 2016

theme: *“live BGP measurements & monitoring”*

We will provide a rich toolbox and “live” data access:



BGP HACKATHON

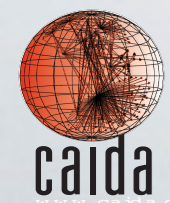
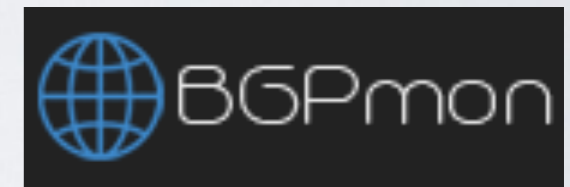
<http://github.com/CAIDA/bgp-hackathon/wiki>

- **6-7 February 2016** (weekend before NANOG 66)
- **San Diego** Supercomputer Center, UC San Diego
- **Theme: live BGP measurements** and monitoring
- Toolbox: *BGPMon, RIPE RIS, PEERING, BGPStream, RIPE Atlas, CAIDA Archipelago, Route Views, looking glasses, AS relationships, AS Rank, Visualization tools, ...*

- How to **contribute:**

- *join us and come over to hack!*
- *help teams as a domain expert*
- *propose projects that hacking teams may pick*
- *offer to join the jury that will assign awards*

>>> bgp-hackathon-info@caida.org <<<



Center for Applied Internet Data Analysis
University of California San Diego

THANKS

bgpstream.caida.org

github.com/CAIDA/bgp-hackathon/wiki

