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

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?

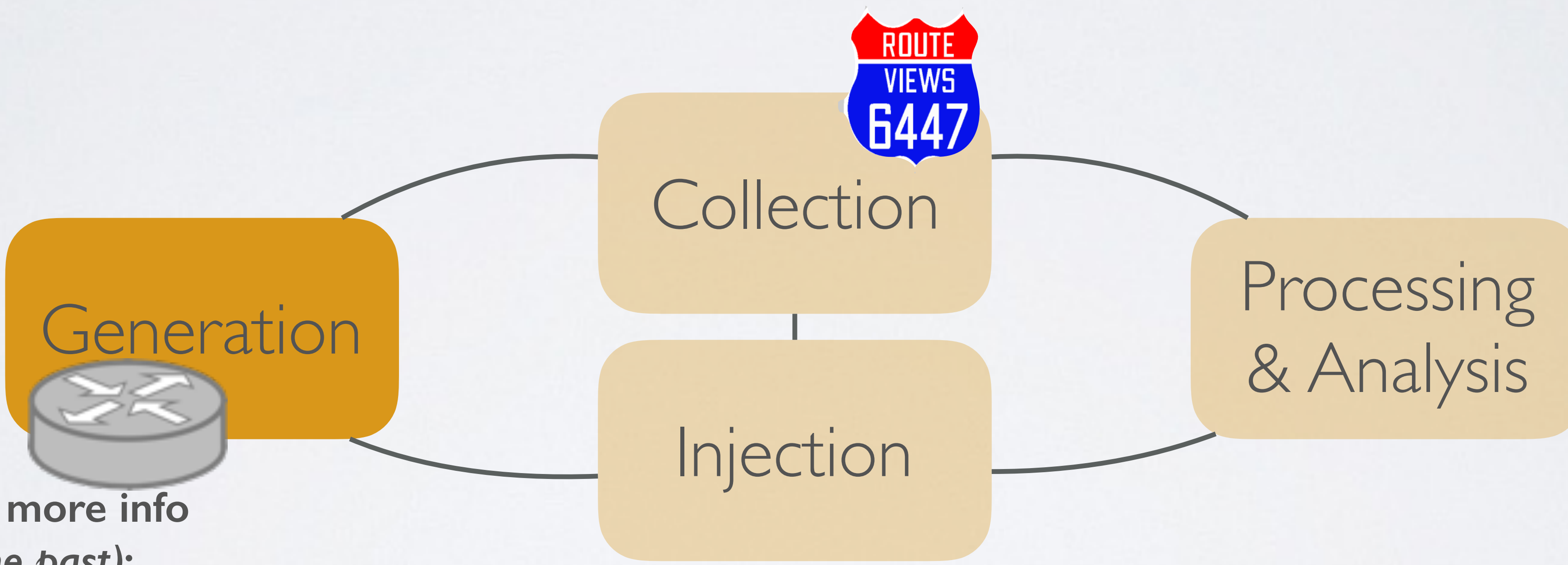
Defining problems and make ***protocol engineering*** decisions through realistic evaluations is difficult also because **we know little about the structure and dynamics of the BGP ecosystem!**

- 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



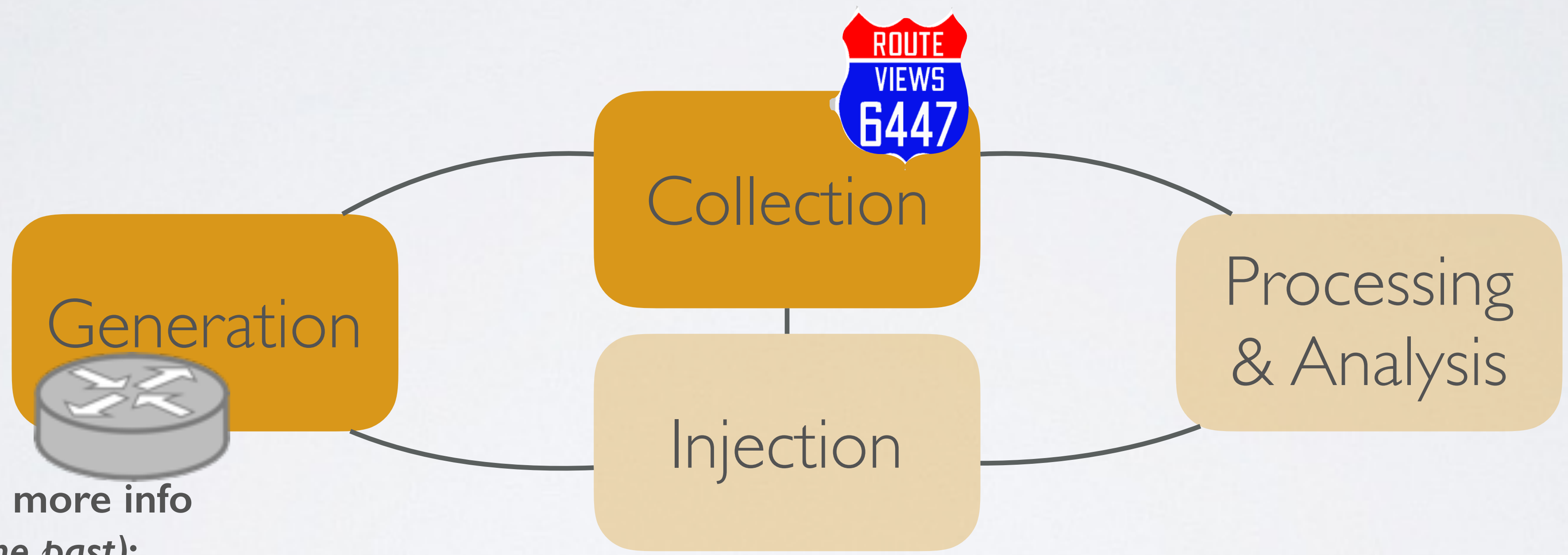
Attempts to generate more info
***(not much traction in the past)*:**

- **`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,



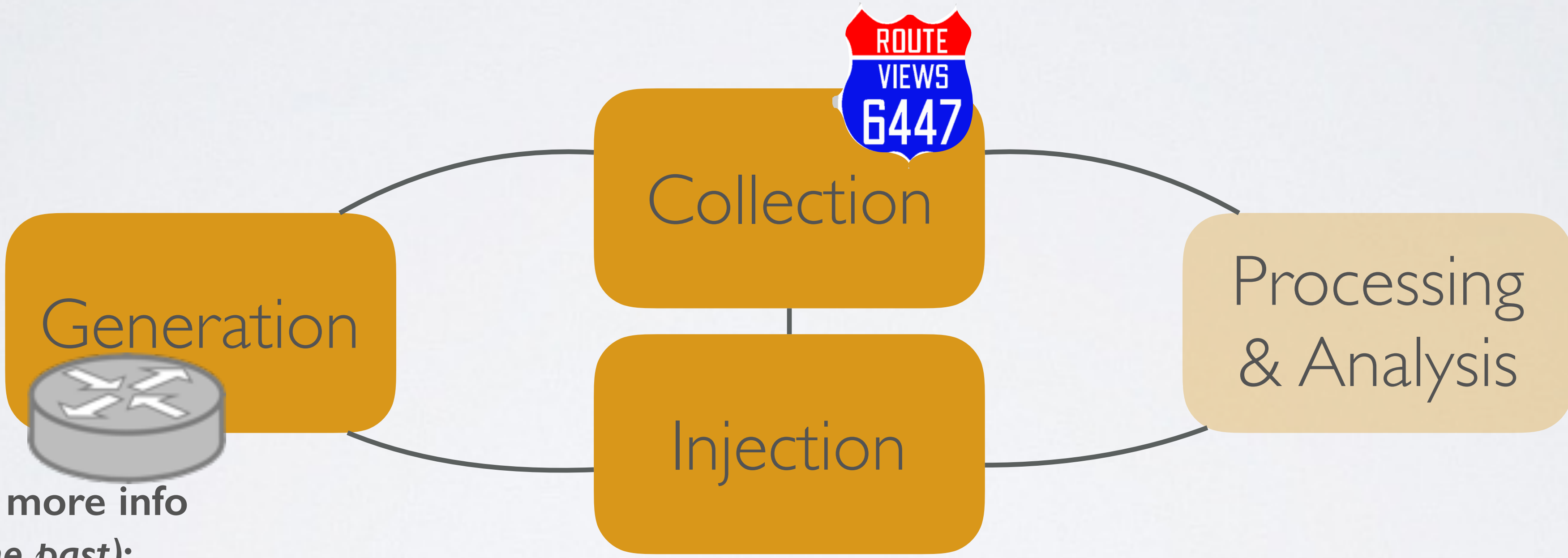
Attempts to generate more info
***(not much traction in the past)*:**

- **`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`**

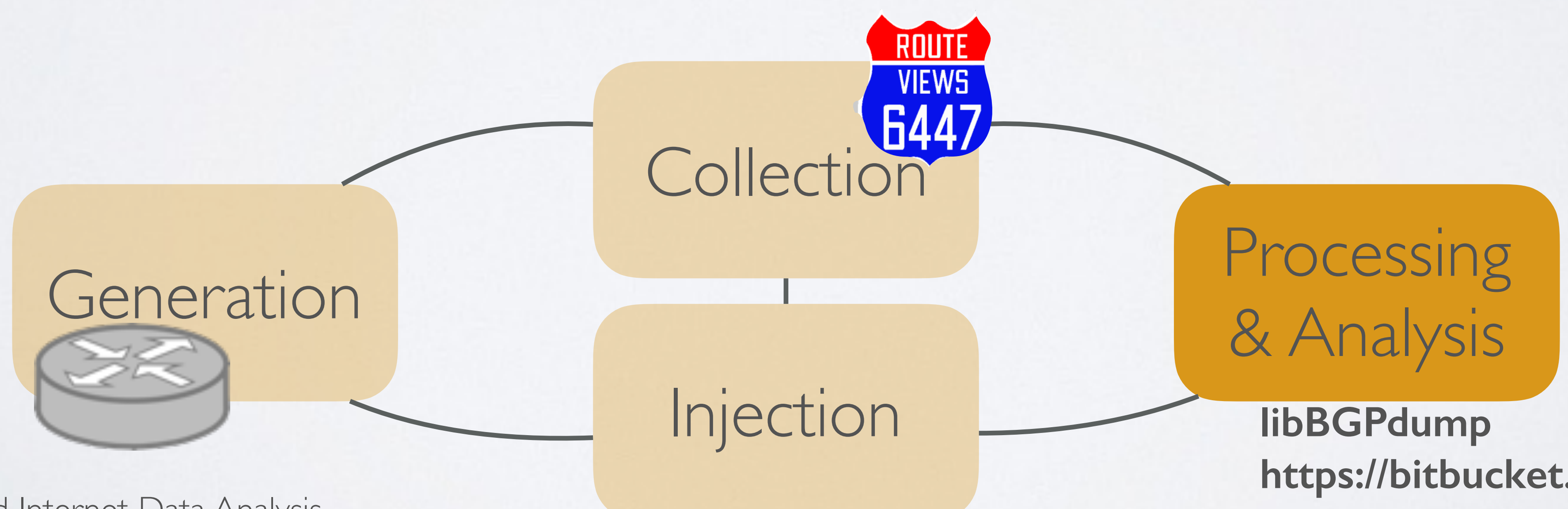
Inject/Receive Routes & Traffic.
PEERING - http://peering.usc.edu
Schlinker et al. *"PEERING: An AS for Us"*, HotNets 2014

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, …



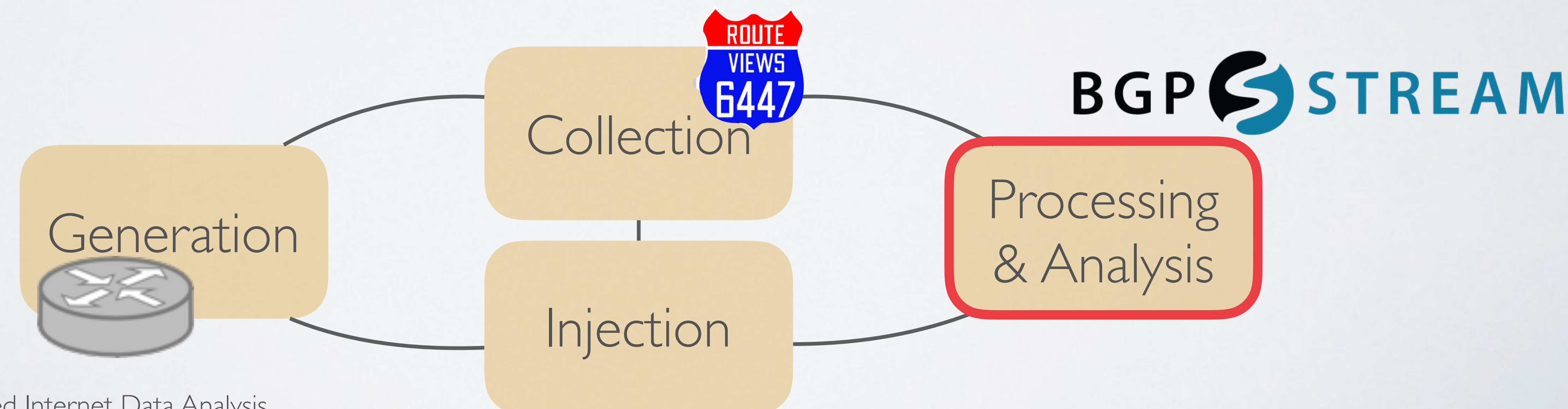
libBGPdump
https://bitbucket.org/ripencc/bgpdump

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

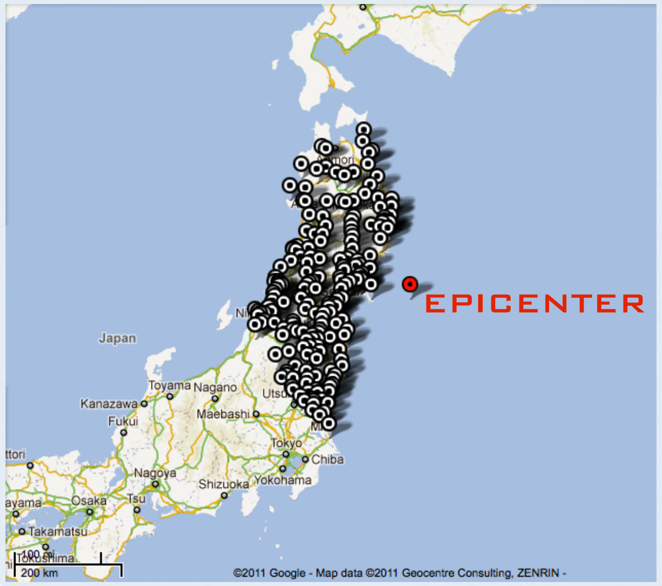
- Country-level Internet Blackouts during the Arab Spring

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

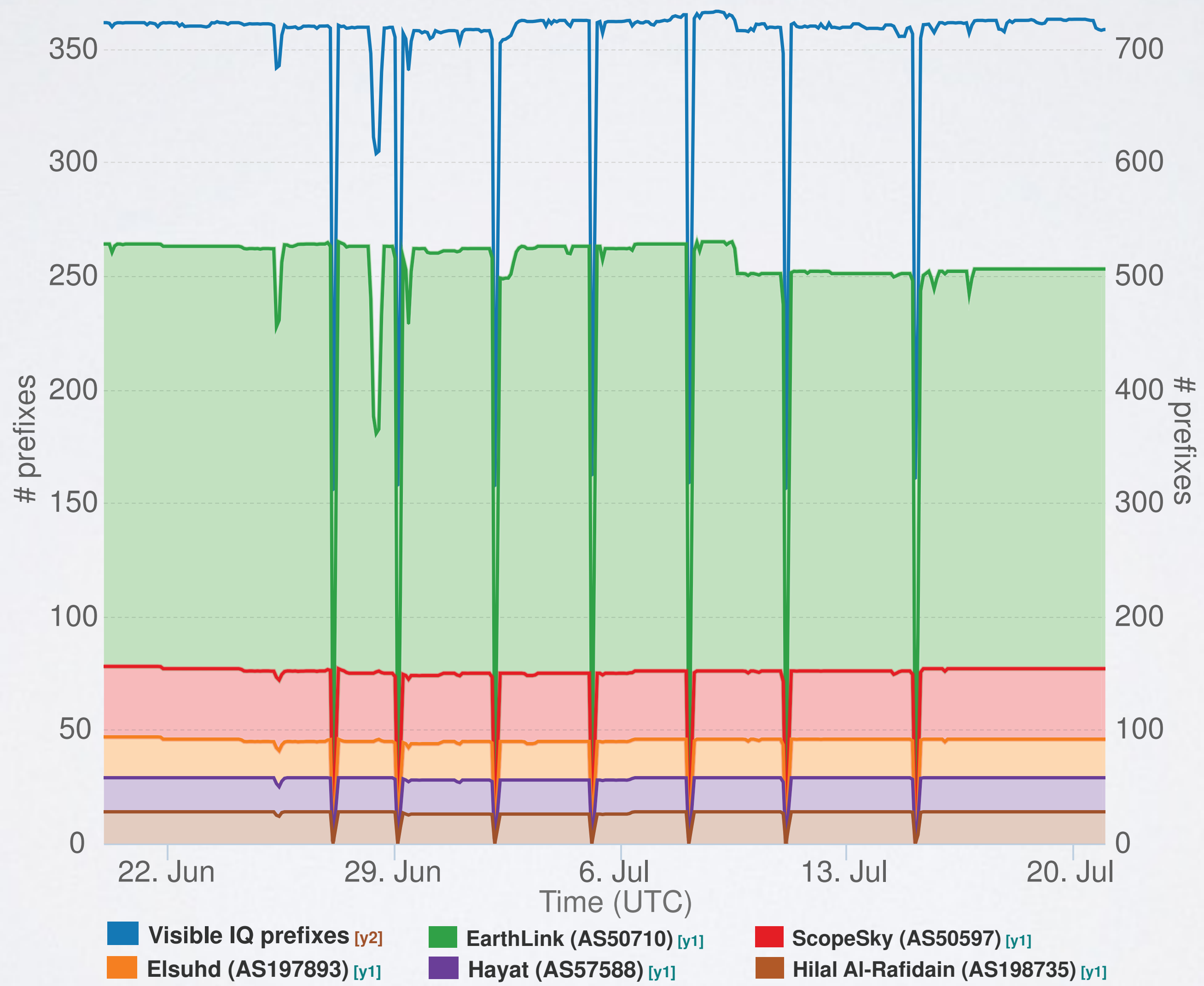
www.caida.org/funding/ioda/

COMCAST

INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

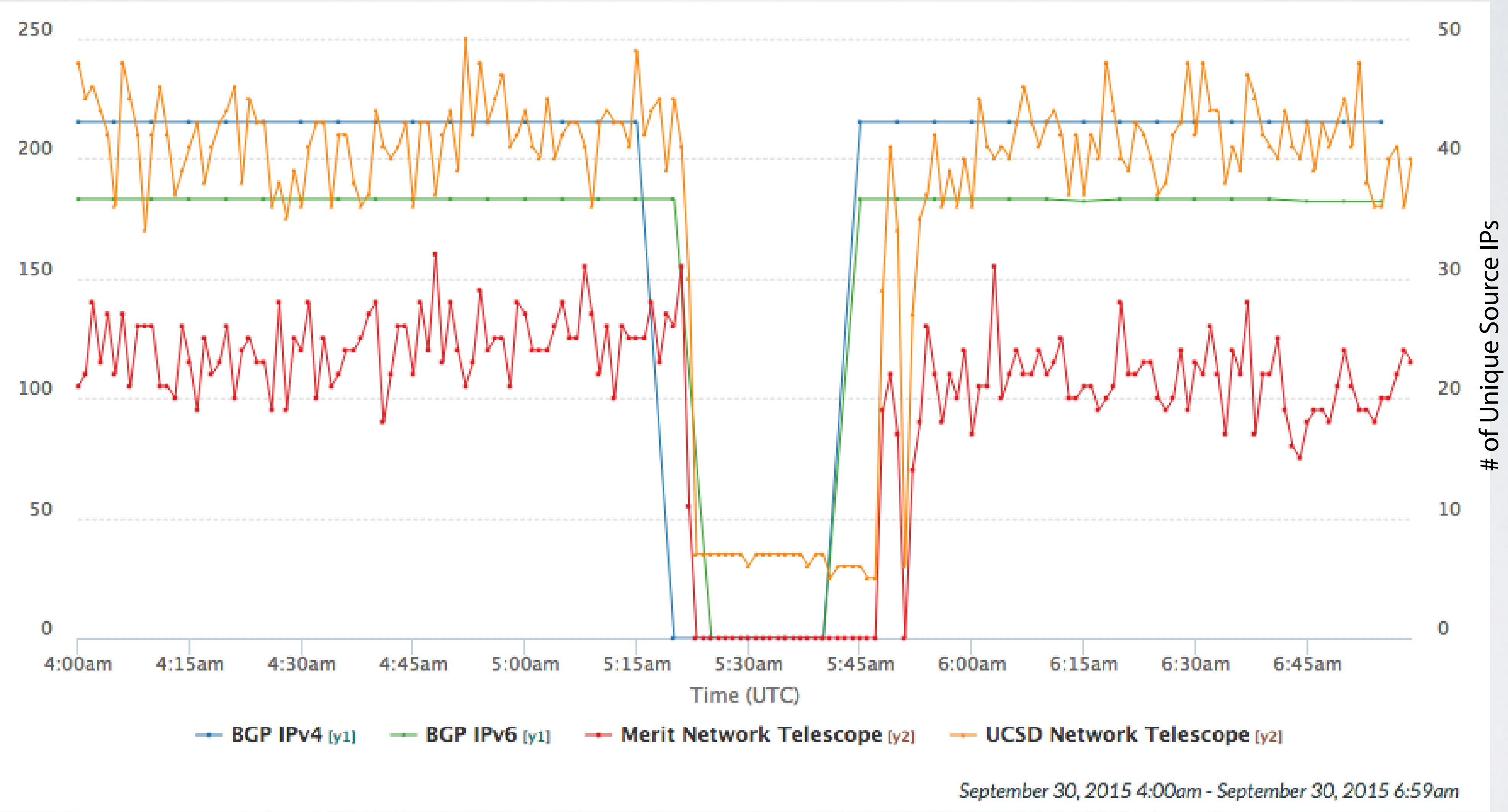
Country-wide Internet outages in Iraq that the government ordered in conjunction with the ministerial preparatory exams - Jul 2015



INSPIRING PROJECTS (1/2)

IODA: Detection and Analysis of Internet Outages

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

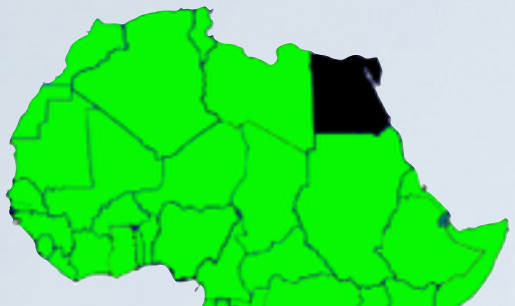


Center for Applied Internet Data Analysis
University of California San Diego

www.caida.org/funding/ioda/

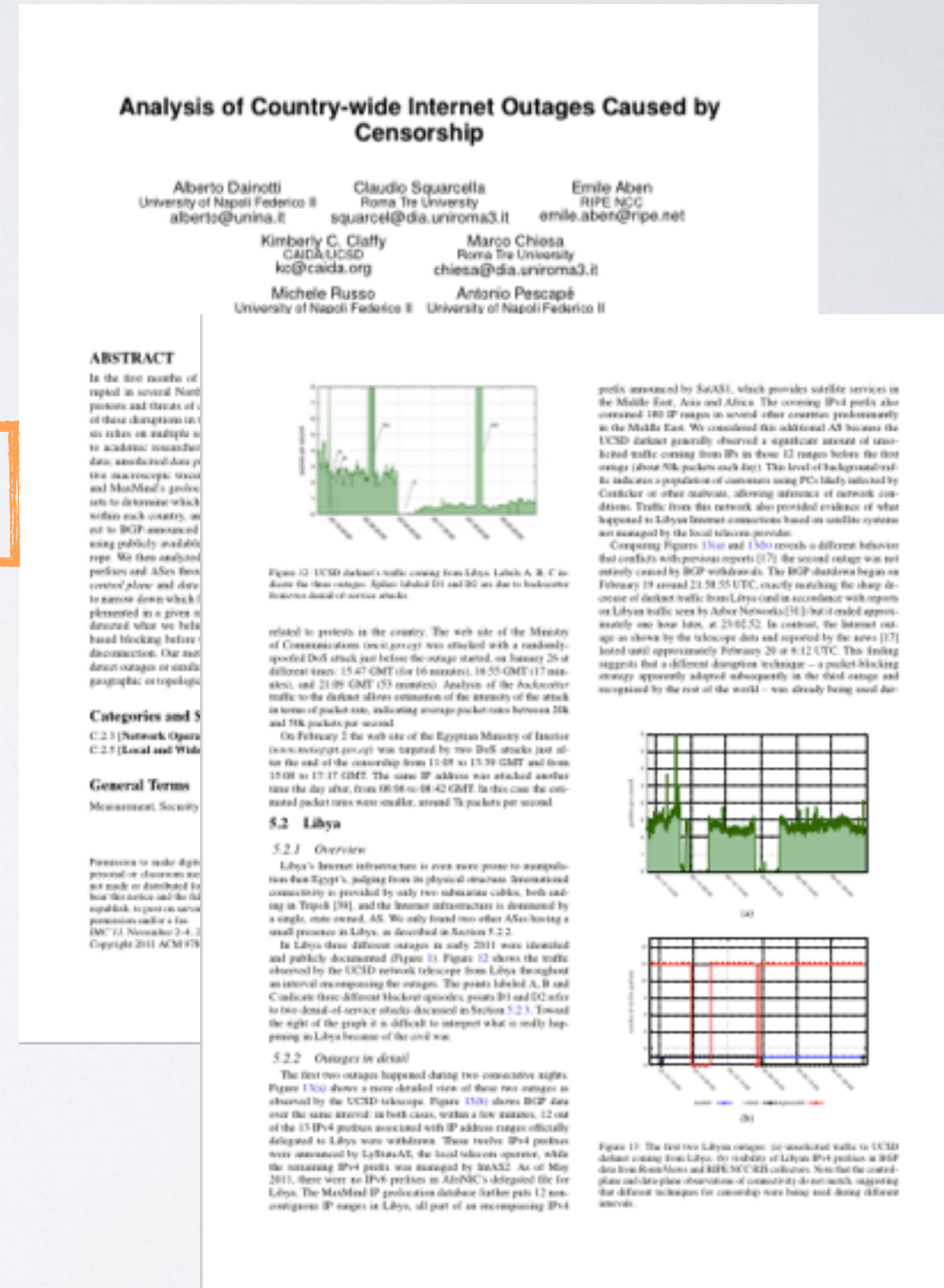
BEFORE IODA

post-event manual analysis



Egypt, Jan 2011
Government orders to shut down the Internet

4 months of work

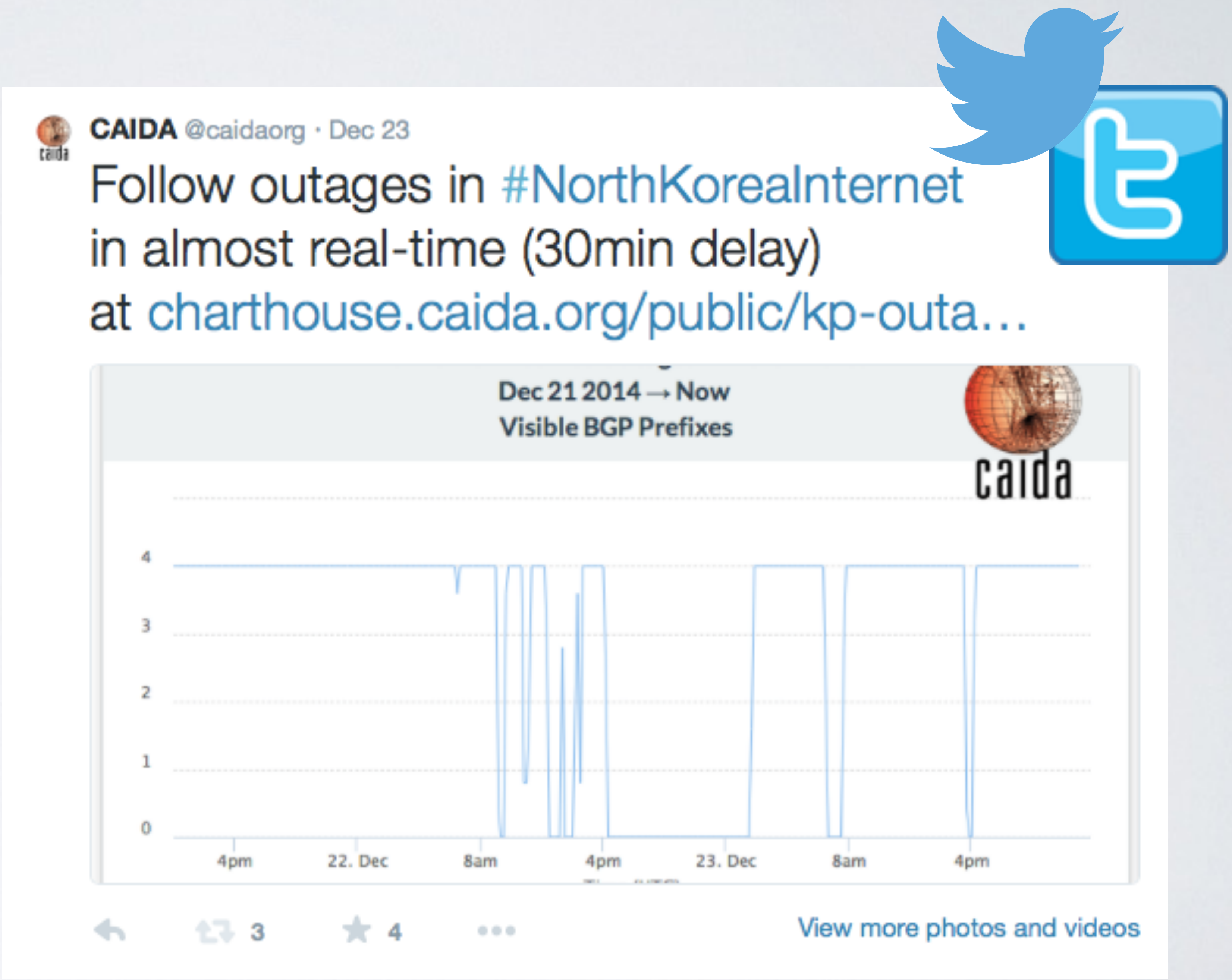


Dainotti et al. "Analysis of Country-wide Internet Outages Caused by Censorship" IMC 2011

IODA TODAY

live Internet monitoring

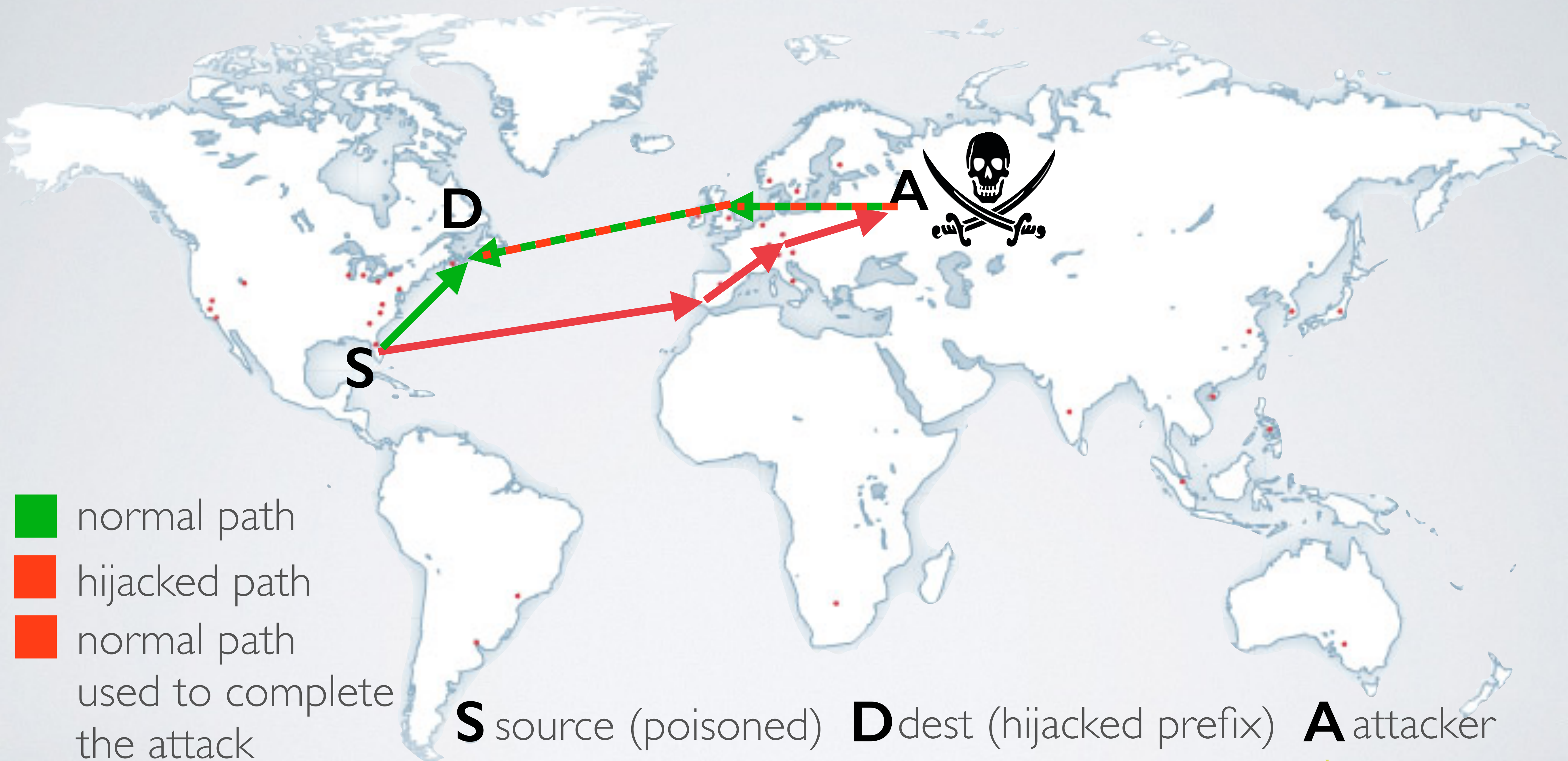
Last Christmas we made it possible for anybody to follow the North Korean disconnection almost live



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

INSPIRING PROJECTS (2/2)

Hijacks: detection of MITM BGP attacks



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

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

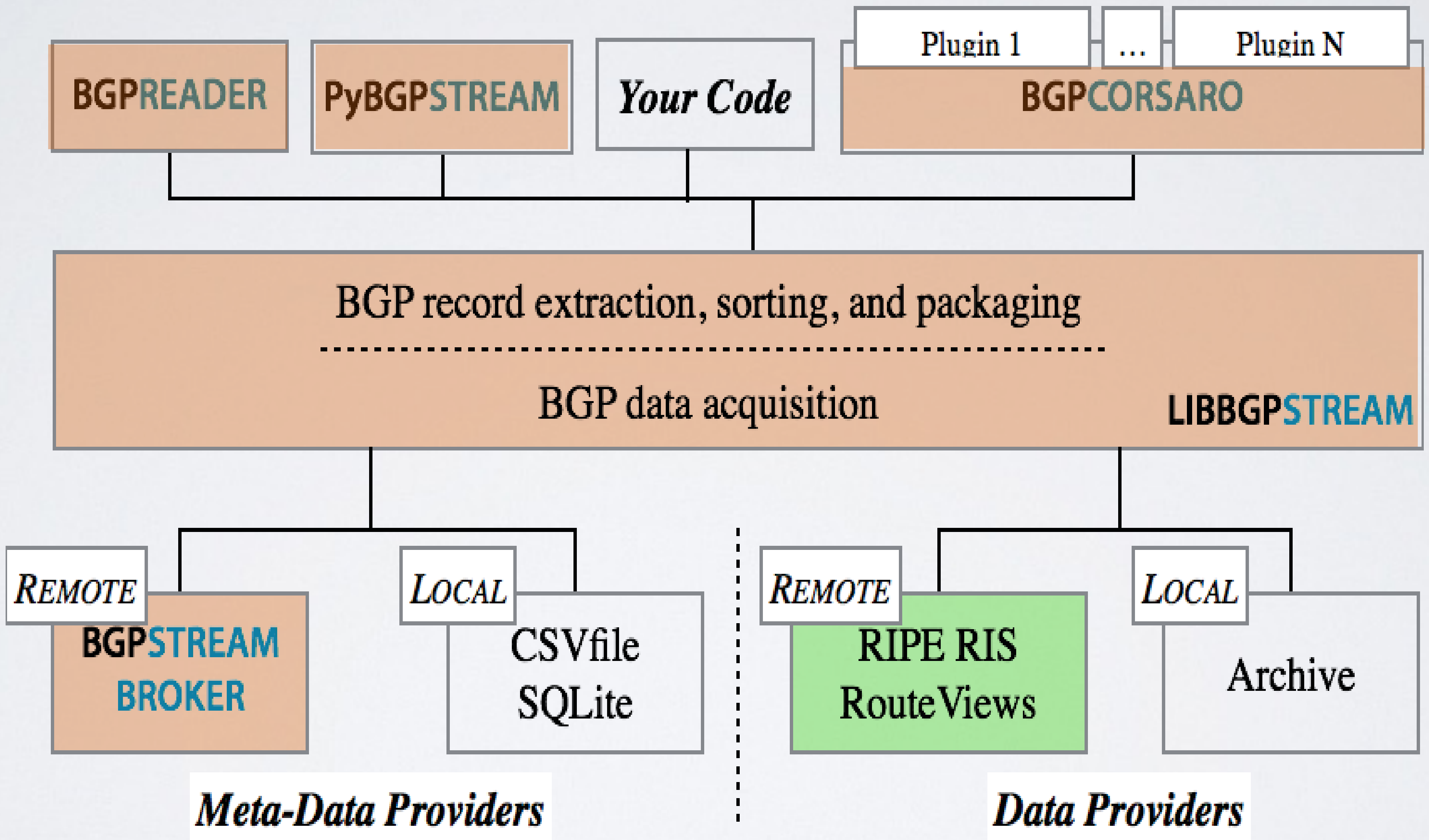
BGPSTREAM

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

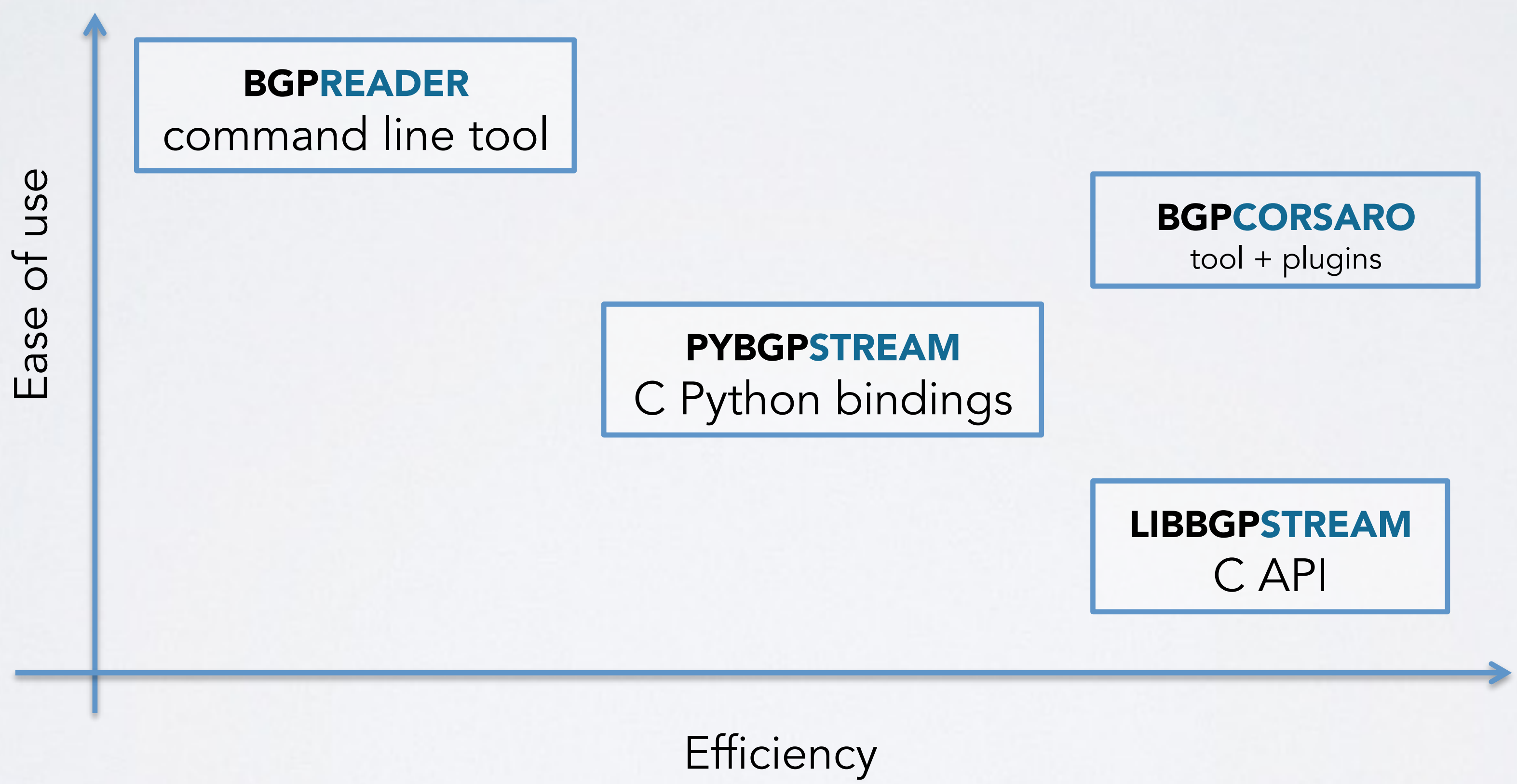
BGPSTREAM

overview



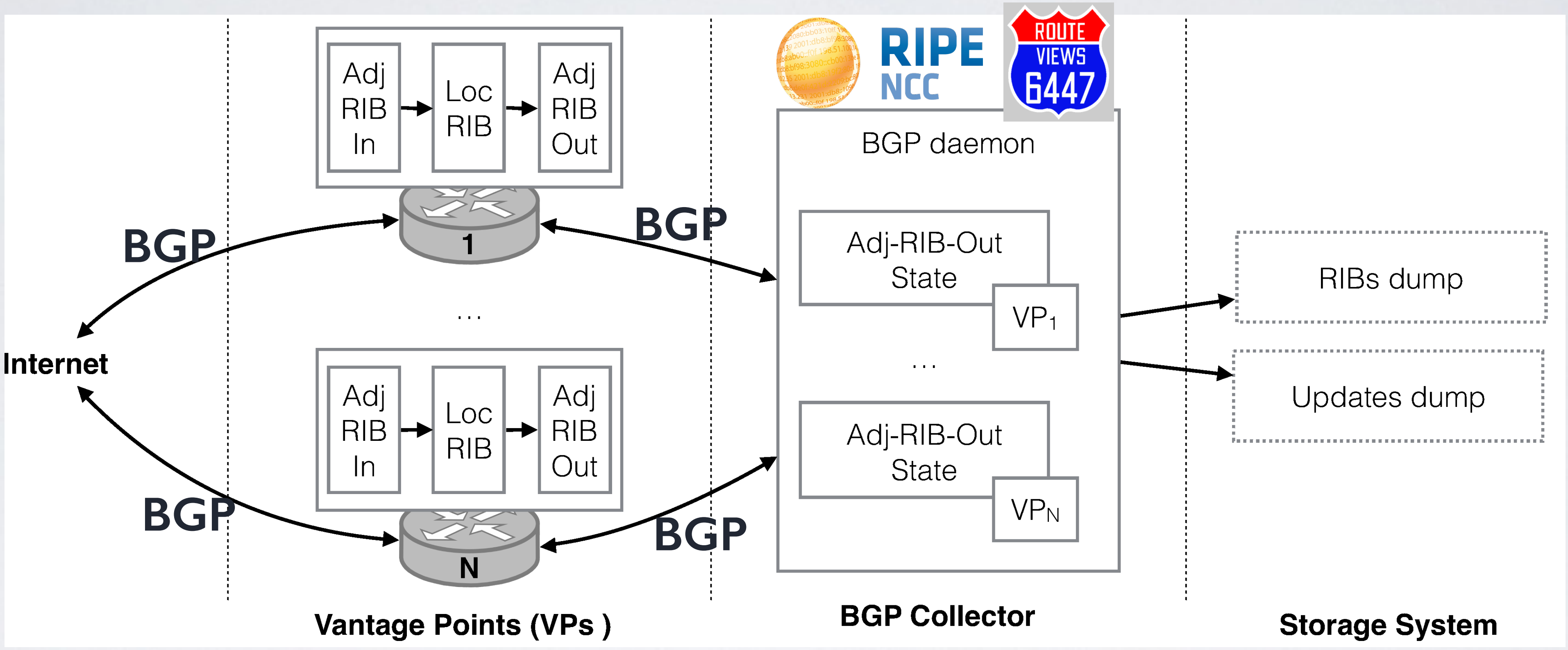
BGPSTREAM

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)
{
  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

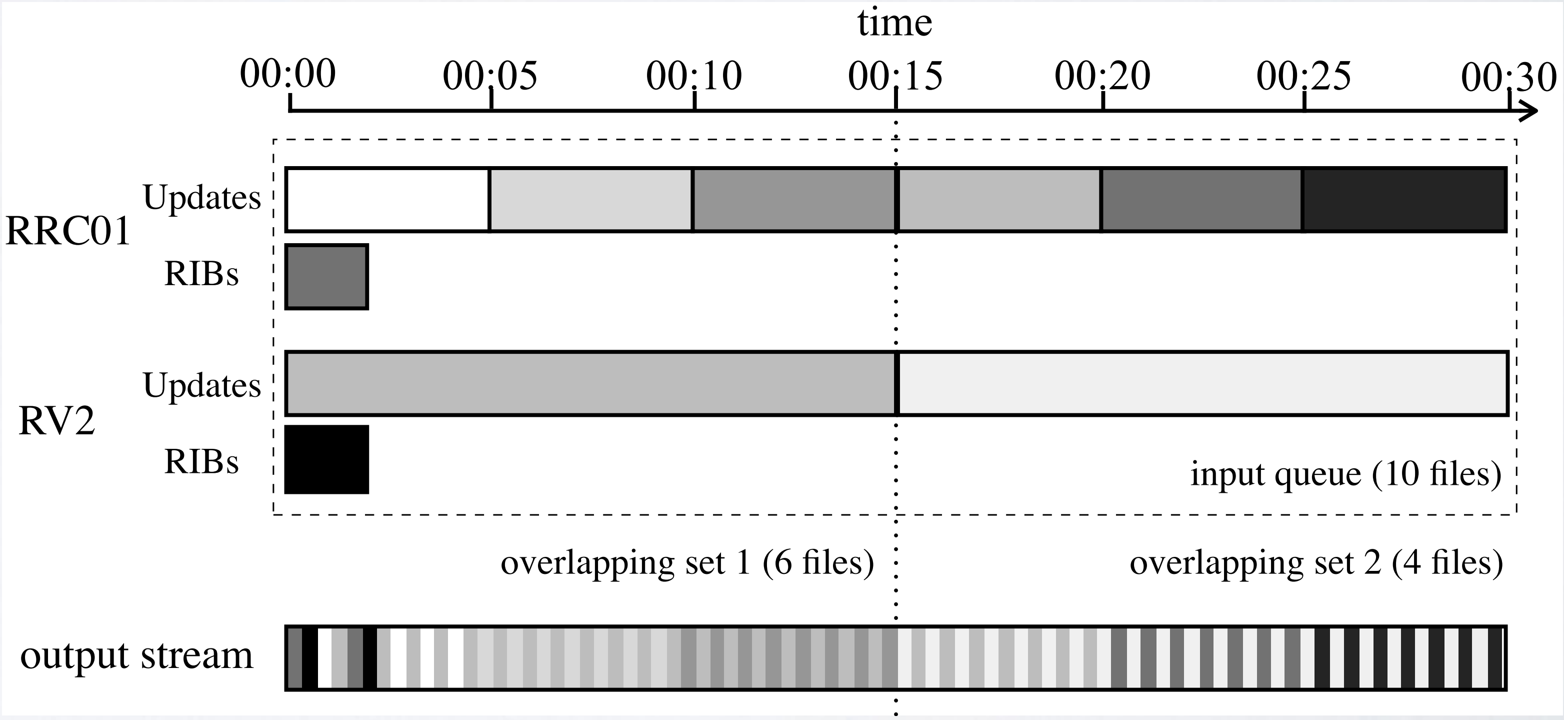
```
/* Start the stream */
bgpstream_start(bs);

/* 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);
    }
  }
}
```

RECORD-LEVEL SORTING

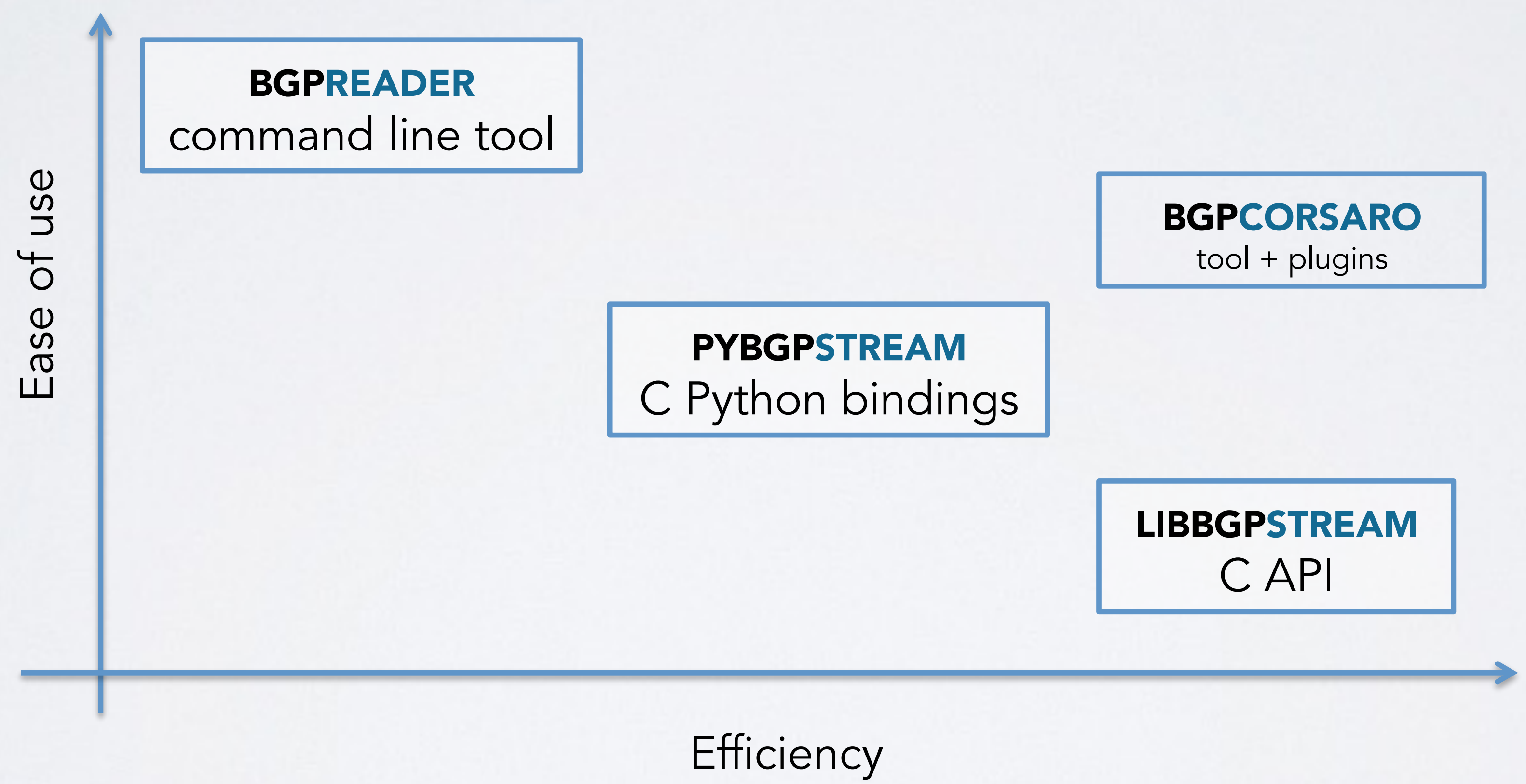
When

- 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

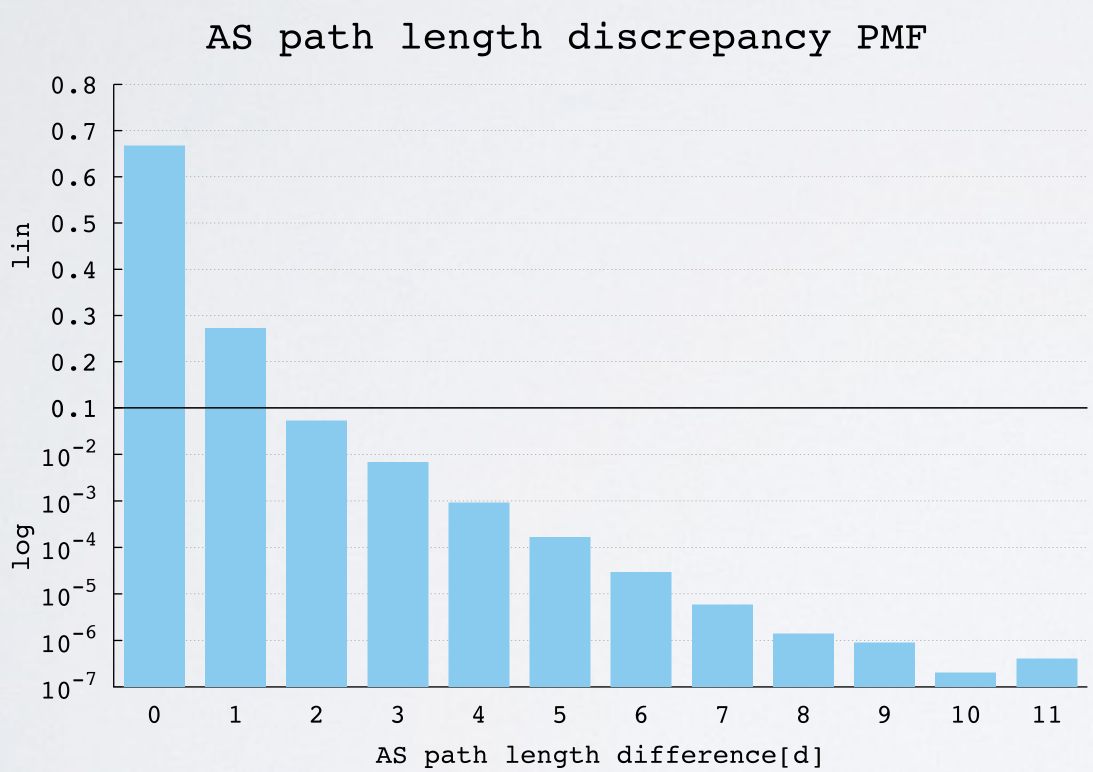
continued..



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)




```
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
```

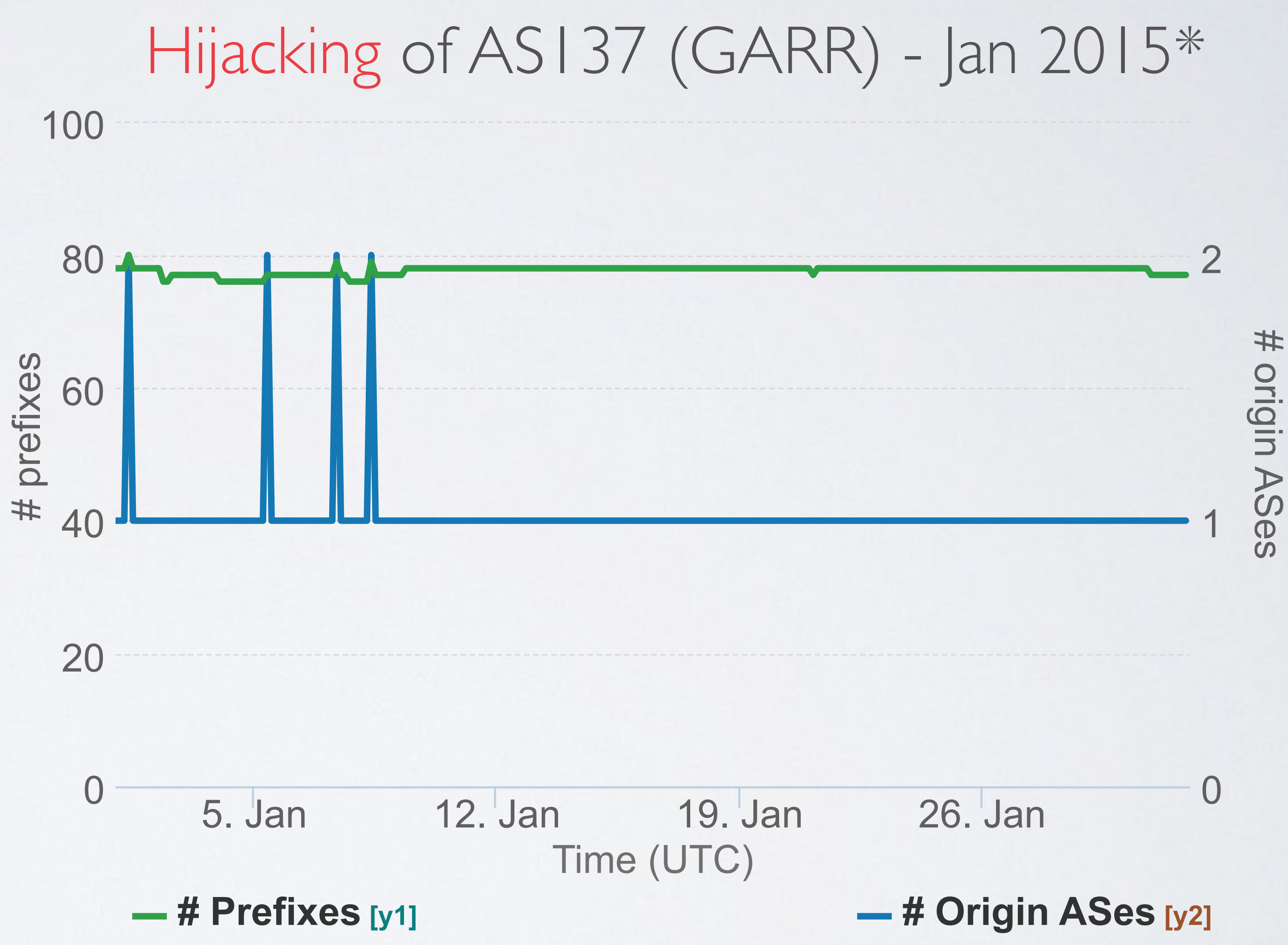
30 LINES OF PYTHON CODE

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*



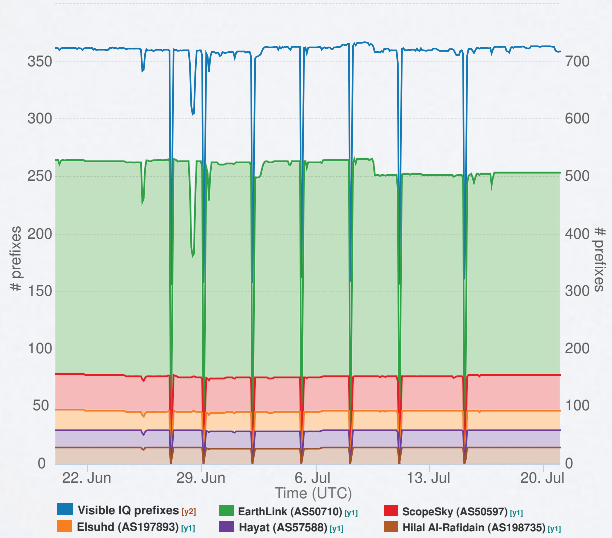
**Originally discovered by Dyn: http://research.dyn.com/2015/01/vast-world-of-fraudulent-routing/*

GLOBAL MONITORING

IODA, HIJACKS, etc.

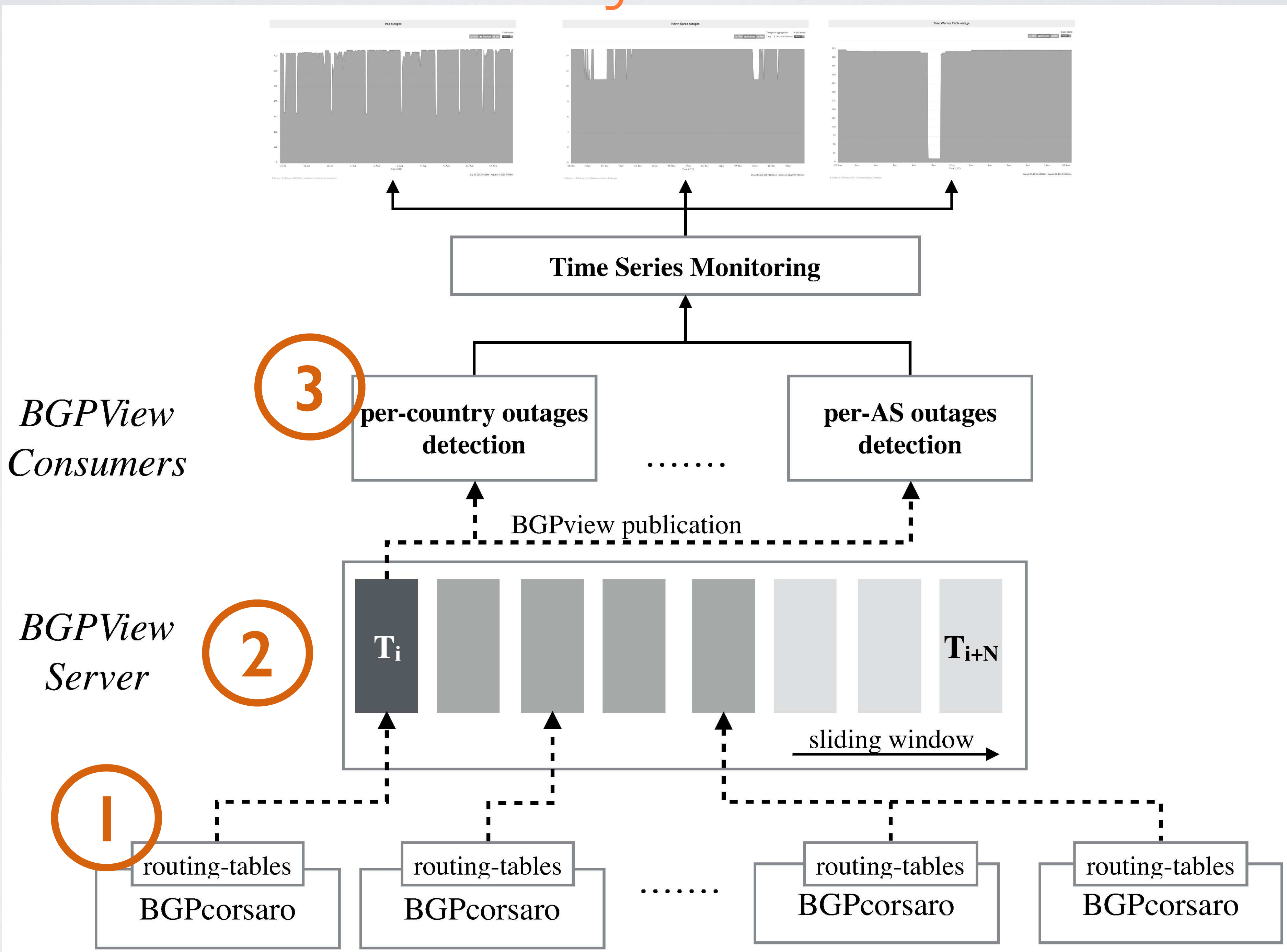
- need to maintain a live **global view** (i.e., for each and every VP) of BGP reachability information updated with **fine time granularity** (e.g., few minutes)

- 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

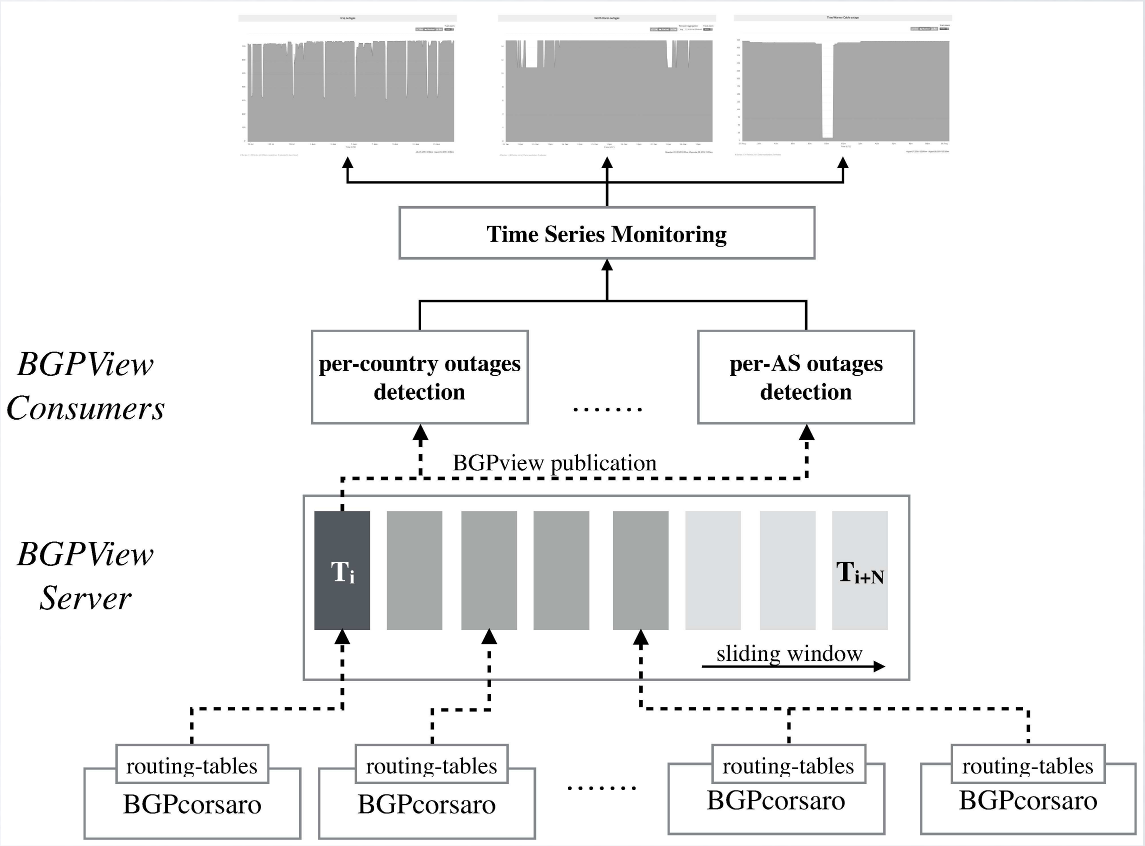
IODA, HIJACKS, etc.



BGPVIEWSERVER

buffering partial/complete BGP views

- At the end of a 1-minute time bin, each BGPCorsaro 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 BGPCorsaro instances have contributed to it



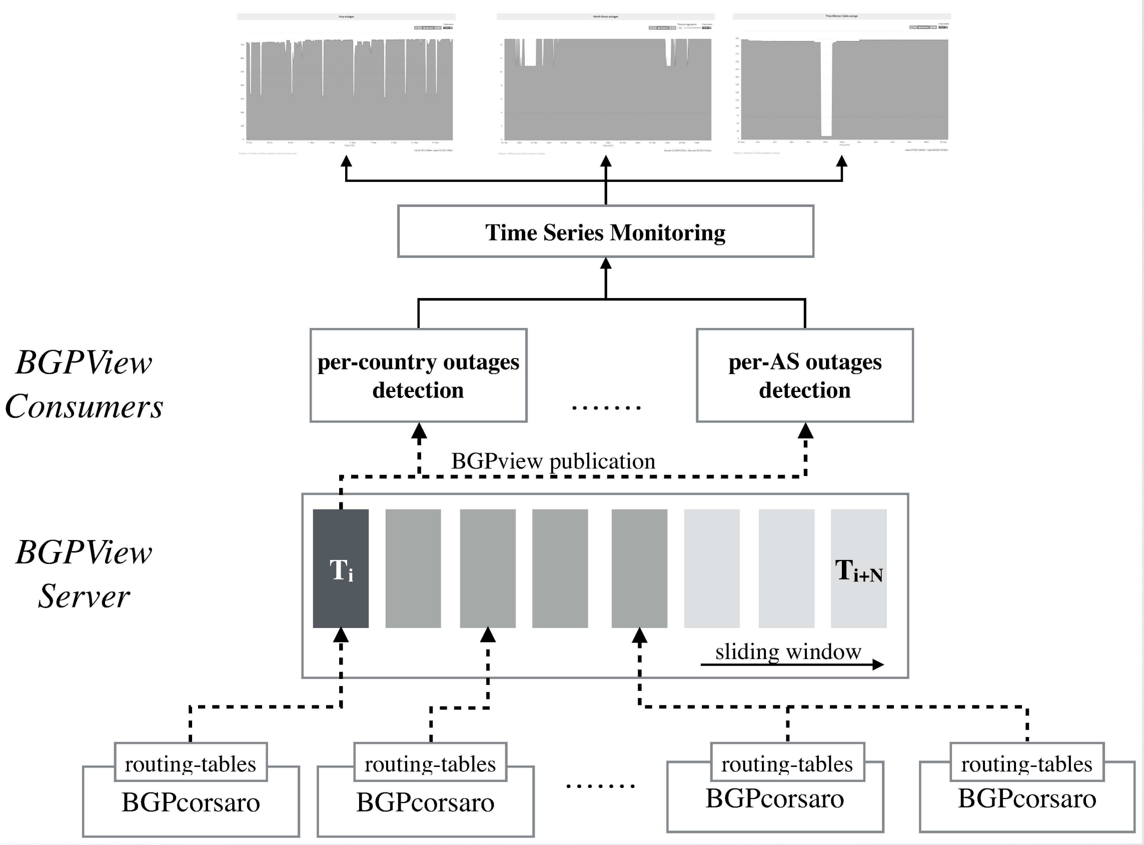
BGPVIEWSERVER

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

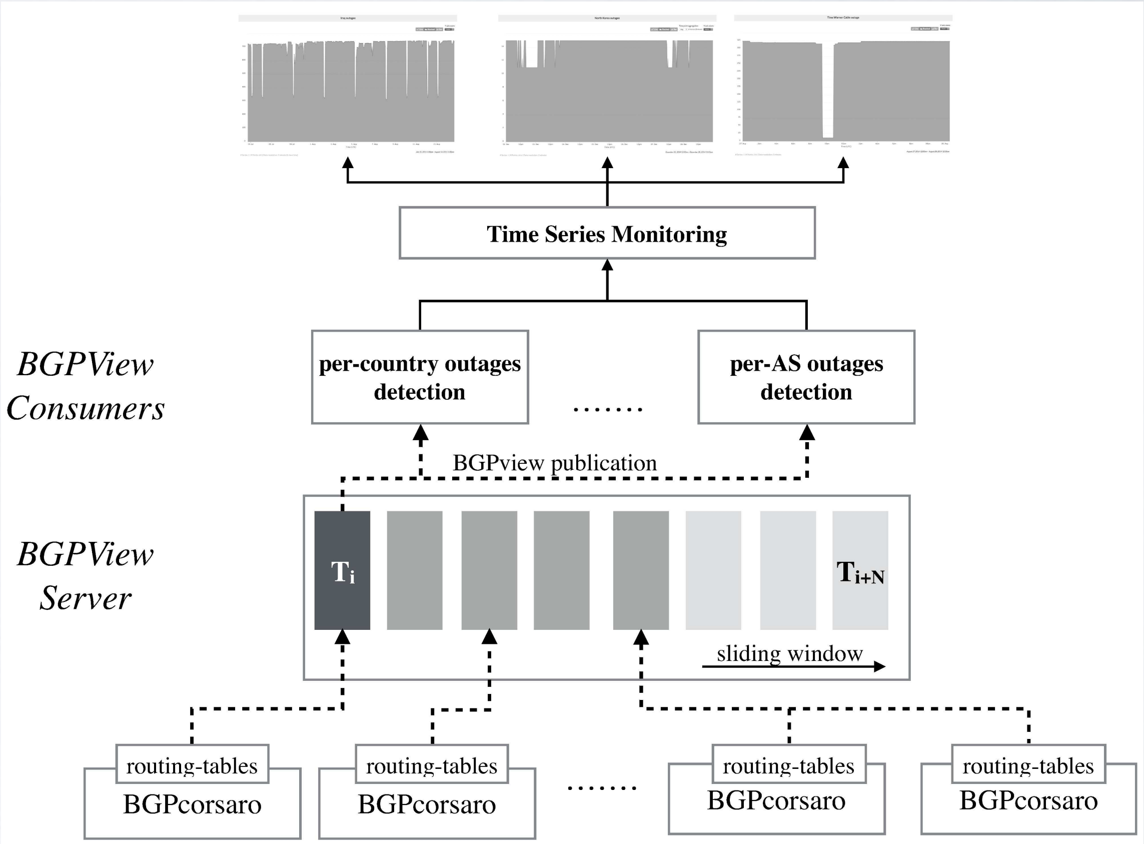
- we publish a BGP view either
  - when all the BGPCorsaro instances have contributed to it (*complete view*)
  - or when it expires, i.e., its time bin is no longer covered by the window (*partial view*)



BGPVIEWSERVER

dimensioning

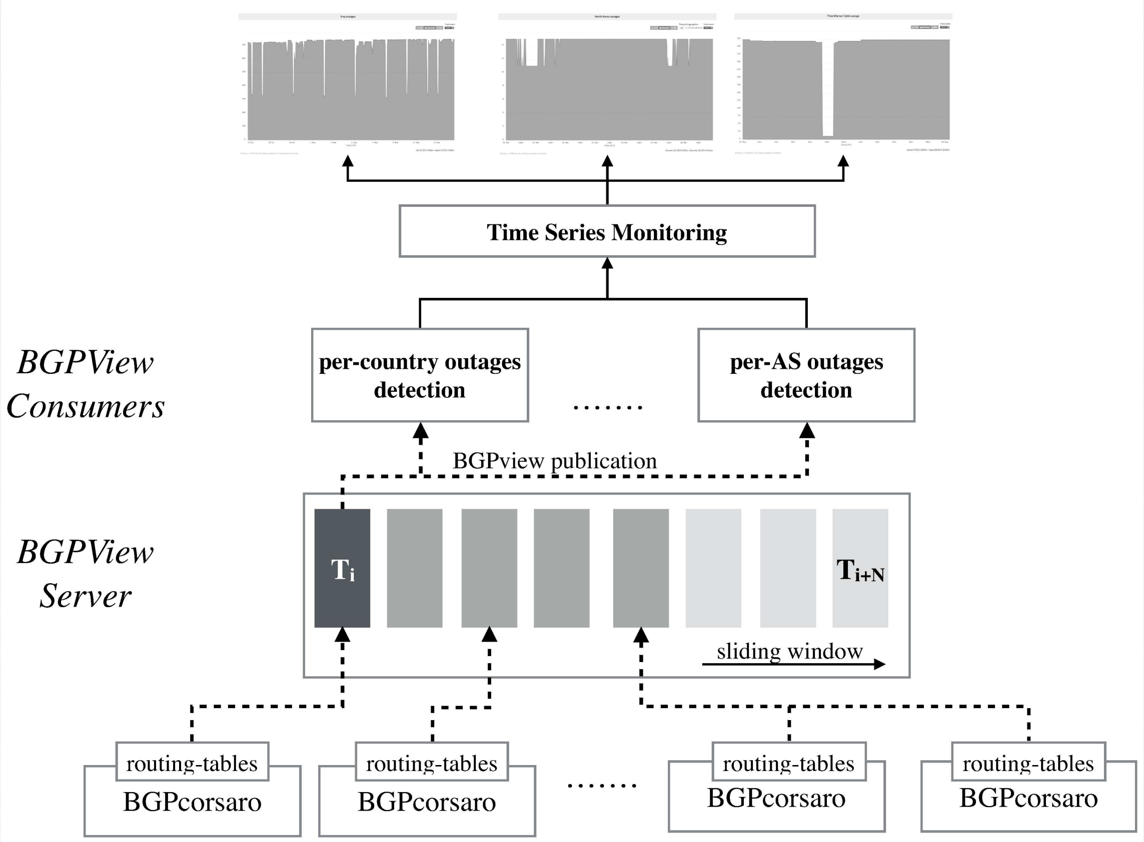
- We **dimension the length of the sliding window** empirically (12 months observation of RV+RIS)
  - the *latency* at which data providers publish dumps
  - the *memory footprint*
    - when processing data from all Route Views and RIPE RIS collectors, a **30 minute** sliding-window buffer requires **≈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 —> 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.



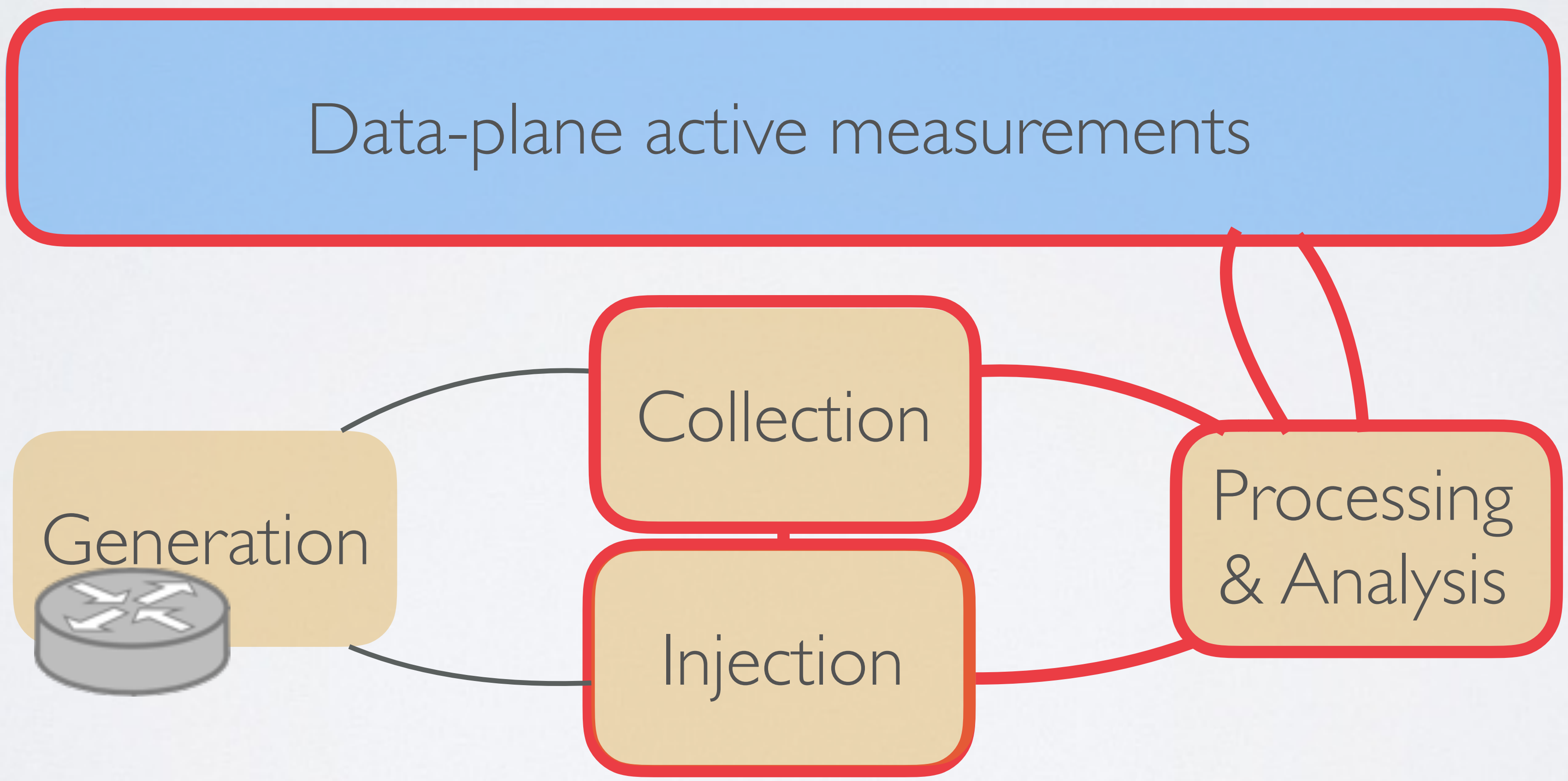
BGPVIEW CONSUMERS

demo on the browser

BGP HACKATHON - FEB 2016

*theme: "**live** BGP measurements & monitoring"*

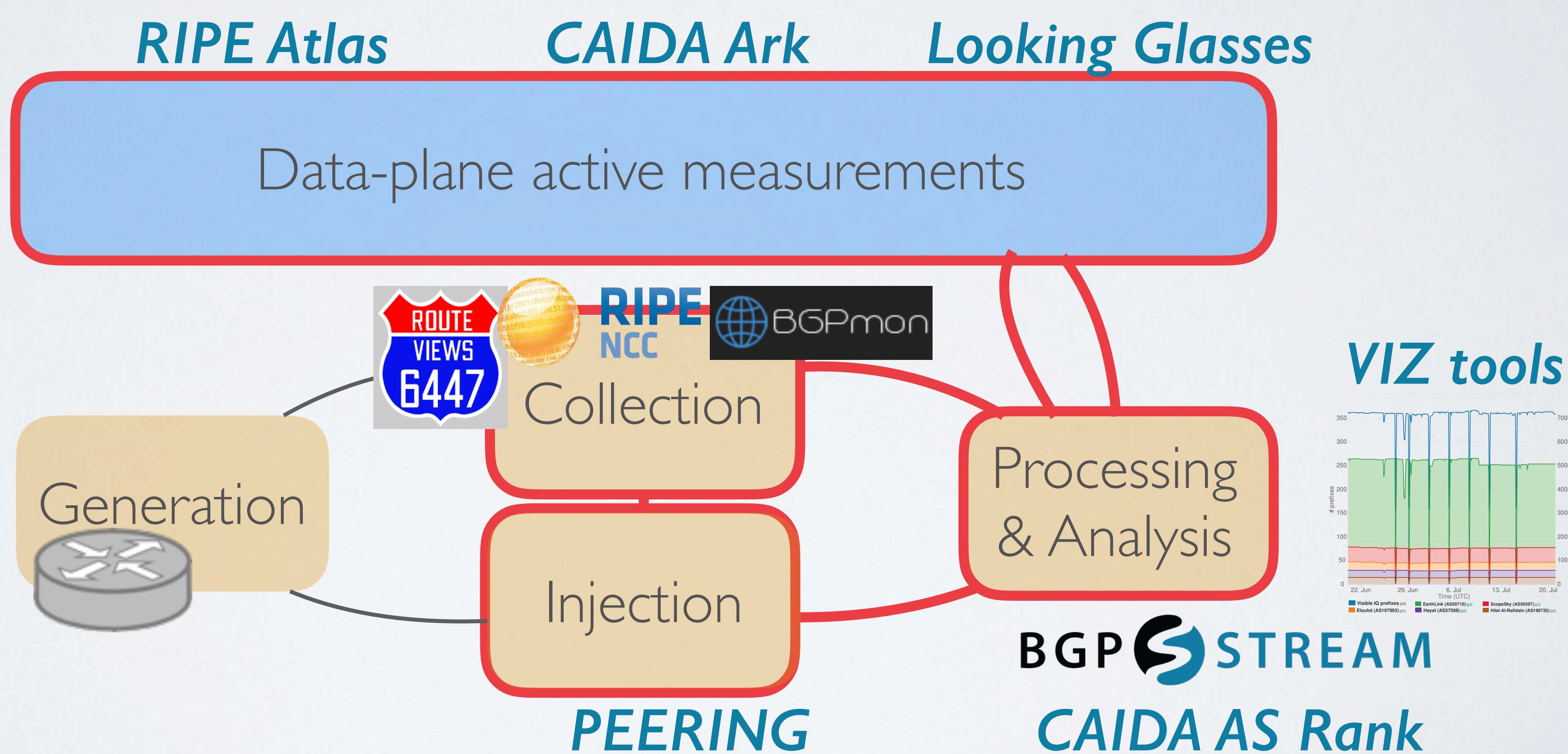
Improve/Integrate tools to study the BGP eco-system. Target practical problems: topology, hijacks, outages, RPKI deployment, path inflation, circuitous paths, policies, relationships, visualize dynamics, …



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 <<<

USC University of Southern California
FORTH Foundation for Research & Technology, Hellas
UC San Diego
BGPmon
UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS
ROUTE VIEWS 6447
RIPE NCC

THANKS

bgpstream.caida.org

github.com/CAIDA/bgp-hackathon/wiki