オープンソースのネットフローツールの運用

JANOG36 BoF

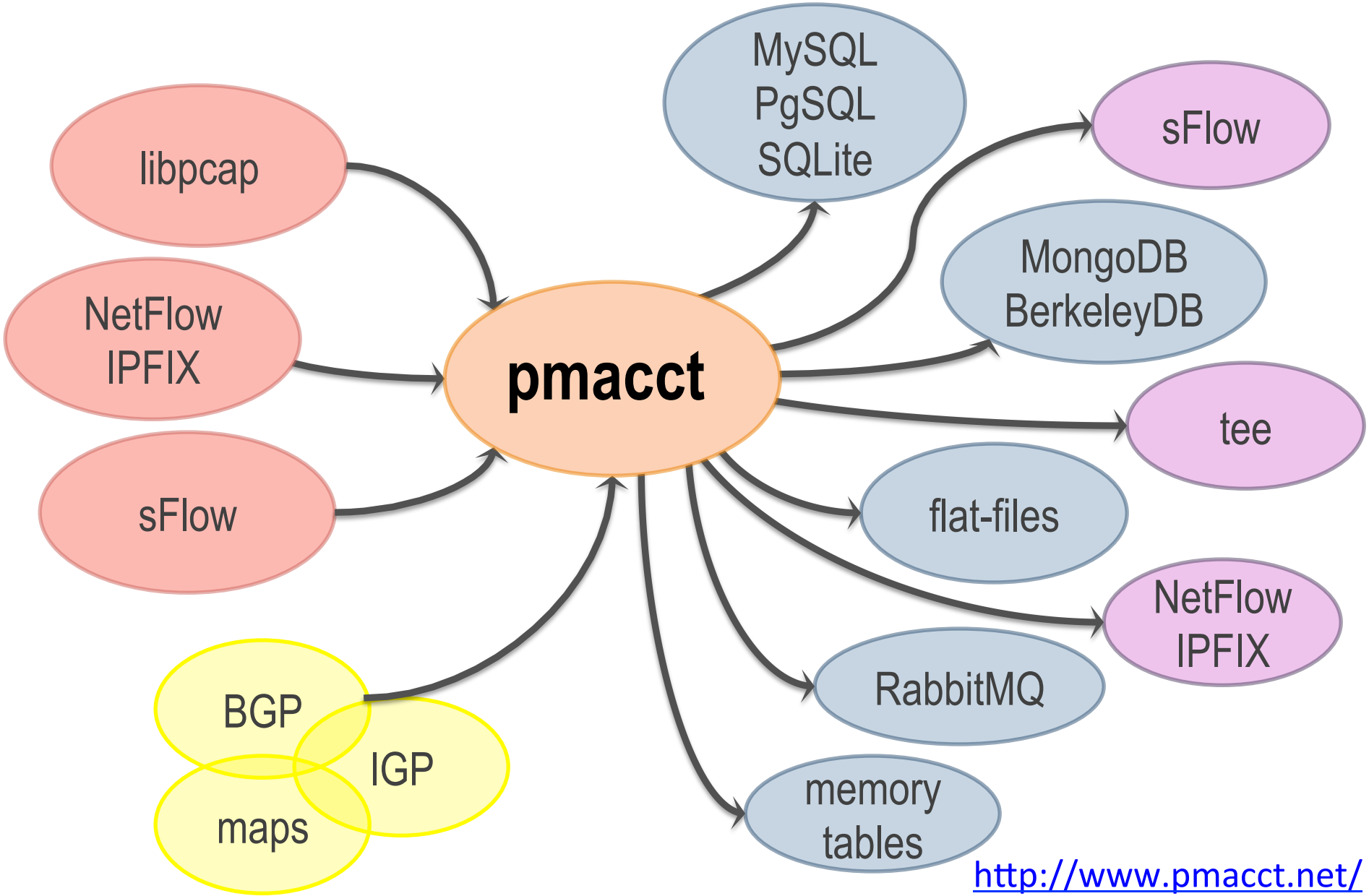
maoke@bbix.net
paolo@pmacct.net

JANOG36 meeting, Kitakyushu – Jul 2015

Introduction

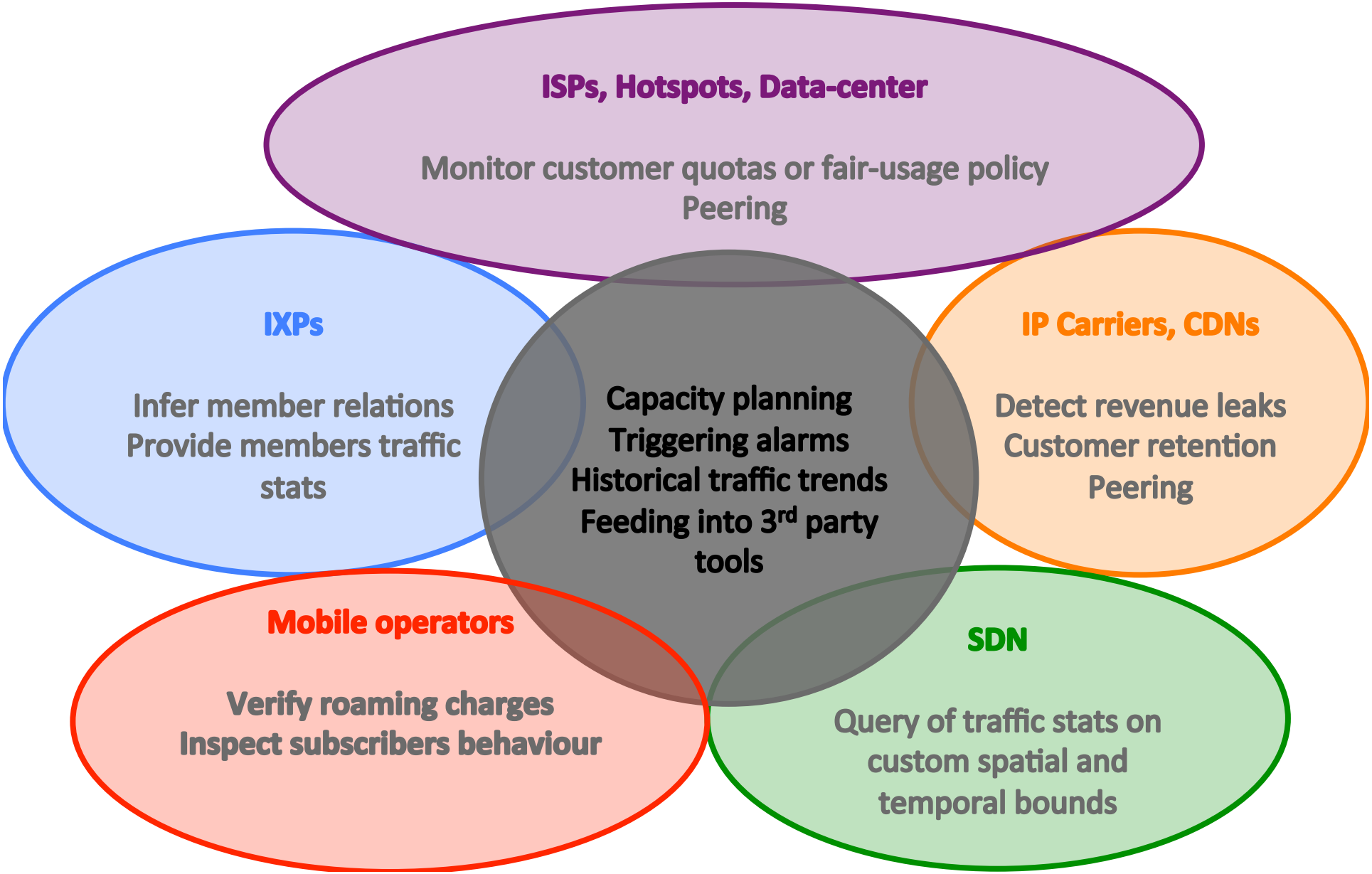
JANOG36 meeting, Kitakyushu – Jul 2015

pmacct is open-source, free, GPL'ed software



http://www.pmacct.net/

Usage scenarios



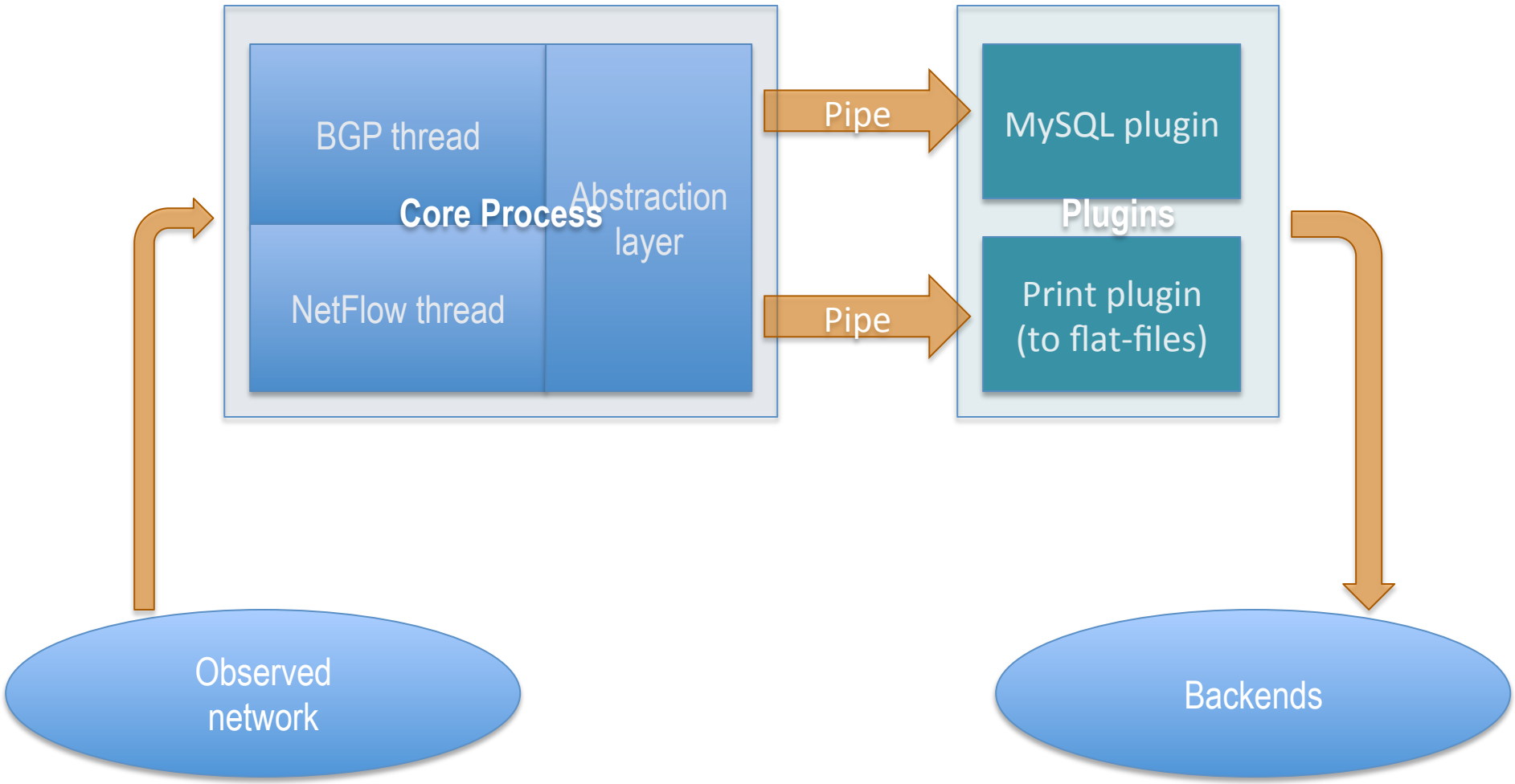
Key pmacct non-technical facts

- 10+ years old project
- Can’t spell the name after the second drink
- Free, open-source, independent
- Under active development
- Innovation being introduced
- Well deployed around, also large SPs
- Aims to be the traffic accounting tool closer to the SP community needs

Some technical facts (1/3)

- Pluggable architecture
  - Straightforward to add support for new collection methods or backends
- An abstraction layer allows out-of-the-box any collection method to interact with any backend
- Both multi-process and (coarse) multi-threading
  - Multiple plugins (of same or different type) can be instantiated at runtime, each with own config

Some technical facts (2/3)

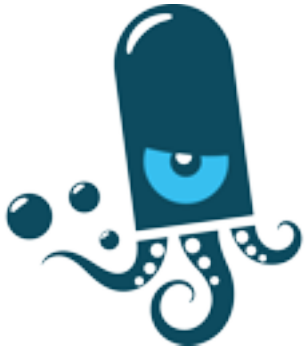


Some technical facts (3/3)

- Pervasive data-reduction techniques, ie.:
  - Data aggregation
  - Tagging and filtering
  - Sampling
- Ability to build multiple views out of the very same collected network traffic dataset , ie.:
  - Unaggregated to flat-files for security and forensic purposes
  - Aggregated as [<ingress router>, <ingress interface>, <BGP next-hop>, <peer destination ASN>] to build an internal traffic matrix for capacity planning purposes

Netflix use-case
(peering analysis, traffic visibility)

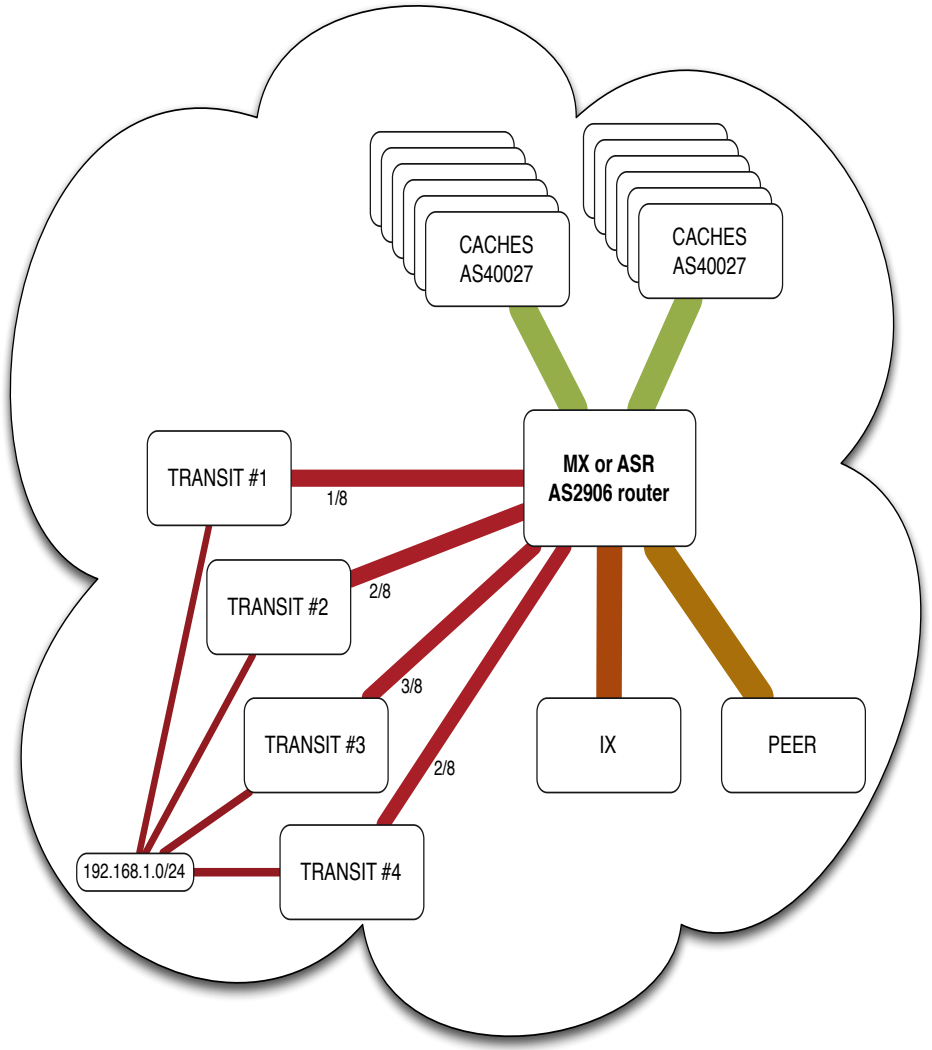
NETFLIX



JANOG36 meeting, Kitakyushu – Jul 2015

Egress BGP hacks

- In many cases too much traffic to handle for 1, 2 or even 4 egress partners
- Use of BGP multi-path via different ASN's



On BGP add-path

- A BGP extension that allows the advertisement of multiple paths for the same address prefix without the new paths implicitly replacing any previous ones
- Draft at IETF: draft-ietf-idr-add-paths-09

The problem

- BGP multi-path, traffic not only sent to a single best path
- pmacct was only aware of the best from its BGP feed

BGP Multi-path

```
192.168.1.0/24      [BGP/170] 3w0d 01:19:58, MED 100, localpref 200
                      AS path: 789 I, validation-state: unverified
                    > to 10.0.0.1 via ae12.0
                    [BGP/170] 3w0d 01:15:44, MED 100, localpref 100
                      AS path: 123 456 789 I, validation-state: unverified
                    > to 10.0.0.2 via ae8.0
                    [BGP/170] 3w0d 01:13:48, MED 100, localpref 100
                      AS path: 321 654 789 I, validation-state: unverified
                    > to 10.0.0.3 via ae10.0
                    [BGP/170] 3w0d 01:18:24, MED 100, localpref 100
                      AS path: 213 546 789 I, validation-state: unverified
                    > to 10.0.0.4 via ae1.0
```

Traditional BGP to pmacct

```
* 192.168.1.0/24             10.0.0.1       100 200     789 I
```

BGP add-path in action

- BGP add-path gives visibility into the N BGP multi-path best-paths

BGP Multi-path

```
192.168.1.0/24      [BGP/170] 3w0d 01:19:58, MED 100, localpref 200
                      AS path: 789 I, validation-state: unverified
                    > to 10.0.0.1 via ae12.0
                    [BGP/170] 3w0d 01:15:44, MED 100, localpref 100
                      AS path: 123 456 789 I, validation-state: unverified
                    > to 10.0.0.2 via ae8.0
                    [BGP/170] 3w0d 01:13:48, MED 100, localpref 100
                      AS path: 321 654 789 I, validation-state: unverified
                    > to 10.0.0.3 via ae10.0
                    [BGP/170] 3w0d 01:18:24, MED 100, localpref 100
                      AS path: 213 546 789 I, validation-state: unverified
                    > to 10.0.0.4 via ae1.0
```

BGP ADD-PATH to pmacct

```
* 192.168.1.0/24          10.0.0.1       100 200     789 I
                          10.0.0.2       100 100     123 456 789 I
                          10.0.0.3       100 100     321 654 789 I
                          10.0.0.4       100 100     213 546 789 I
```

NetFlow/IPFIX and BGP add-path (1/2)

- OK, so we have visibility in the N best-paths ..
- .. but how to map NetFlow traffic onto them?
  - We don't want to get in the exercise of hashing traffic onto paths ourselves as much as possible
  - NetFlow will tell! BGP next-hop in NetFlow is used as selector to tie the right BGP information to traffic data
  - Initially concerned if the BGP NextHop in NetFlow would be of any use to determine the actual path
    - We verified it accurate and consistent across vendors

NetFlow/IPFIX and BGP add-path (2/2)

NetFlow

```
SrcAddr:         10.0.1.71
DstAddr:         192.168.1.148
NextHop:         10.0.0.3
InputInt:        662
OutputInt:       953
Packets:         2
Octets:          2908
Duration:        5.112000000 sec
SrcPort:         80
DstPort:         33738
TCP Flags:       0x10
Protocol:        6
IP ToS:          0x00
SrcAS:           2906
DstAS:           789
SrcMask:         26 (prefix: 10.0.1.64/26)
DstMask:         24 (prefix: 192.168.1.0/24)
```

BGP ADD-PATH to pmacct

```
* 192.168.1.0/24          10.0.0.1       100 200     789 I
                          10.0.0.2       100 100     123 456 789 I
                          10.0.0.3       100 100     321 654 789 I
                          10.0.0.4       100 100     213 546 789 I
```

Deployment notes

- Multiple pmacct servers in various locations
- BGP ADD-PATHS is being set up between routers and the pmacct servers
  - Sessions configured as iBGP, RR-client
  - Juniper ADD-7 (maximum)
  - Cisco ADD-ALL
- NetFlow is being exported to the pmacct servers:
  - Mix of NetFlow v5, v9 and IPFIX

Spotify use-case (SDN)

JANOG36 meeting, Kitakyushu – Jul 2015

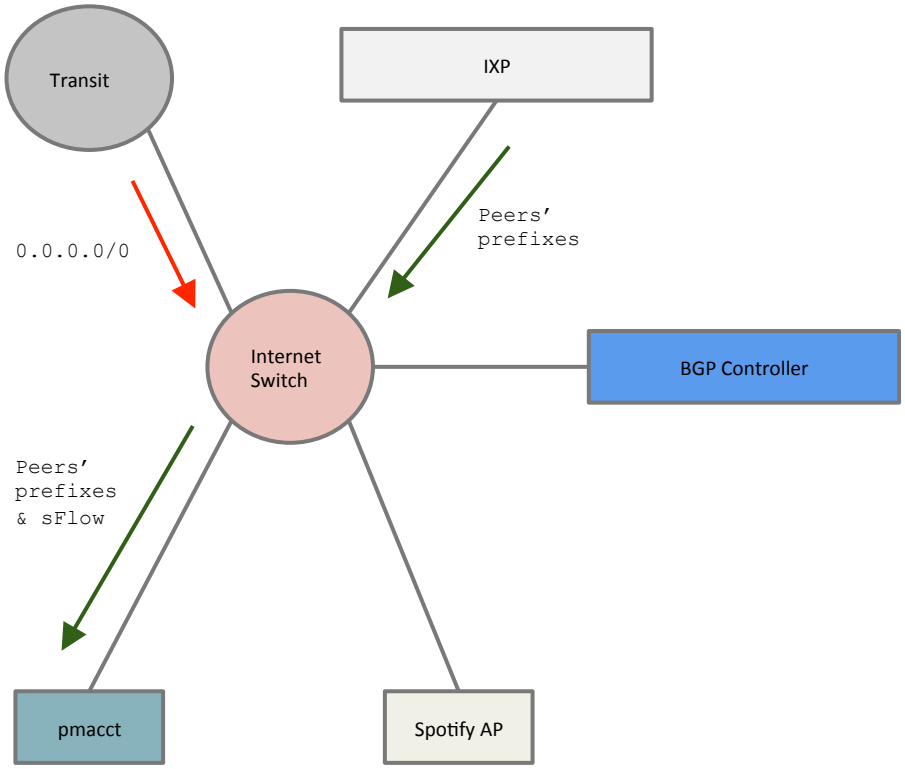
When you travel …

- Example: Spotify datacenter in Stockholm
  - Total prefixes: ~519k
  - Prefixes from peers: ~150k
  - Average # of active prefixes per day: **~16k**
- Example explained:
  - Spotify streams music to users
  - Users are typically served from the closest DC
  - Why would the Spotify DC in San Jose need to specifically know how to reach users in $EU_COUNTRY

Goal of our work

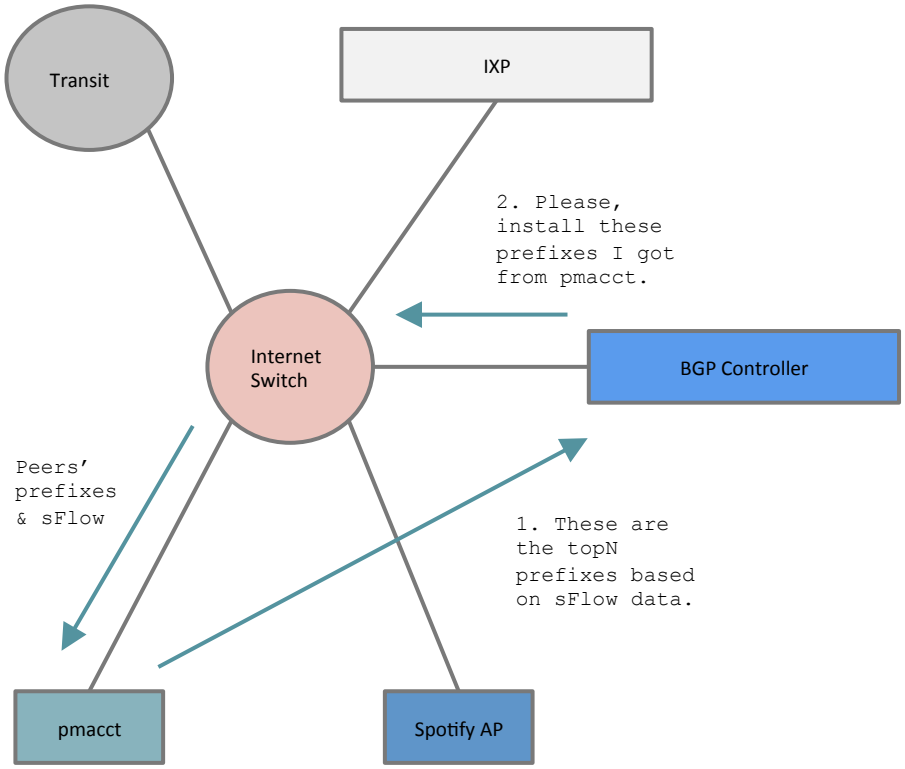
- Make a selection of “needed” routes from the RIB so to be able to fit them on the FIB of a switch with commodity ASICs
- In simplest term this can be reduced to a TopN problem, where N is the amount of routes the commodity ASIC can fit

Overview



- Transit will send the default route to the Internet Switch. The route is installed by default in the FIB
- We receive from the IXP all the peers´ prefixes. Those are not installed, they are forwarded to pmacct
- pmacct will receive in addition sFlow data

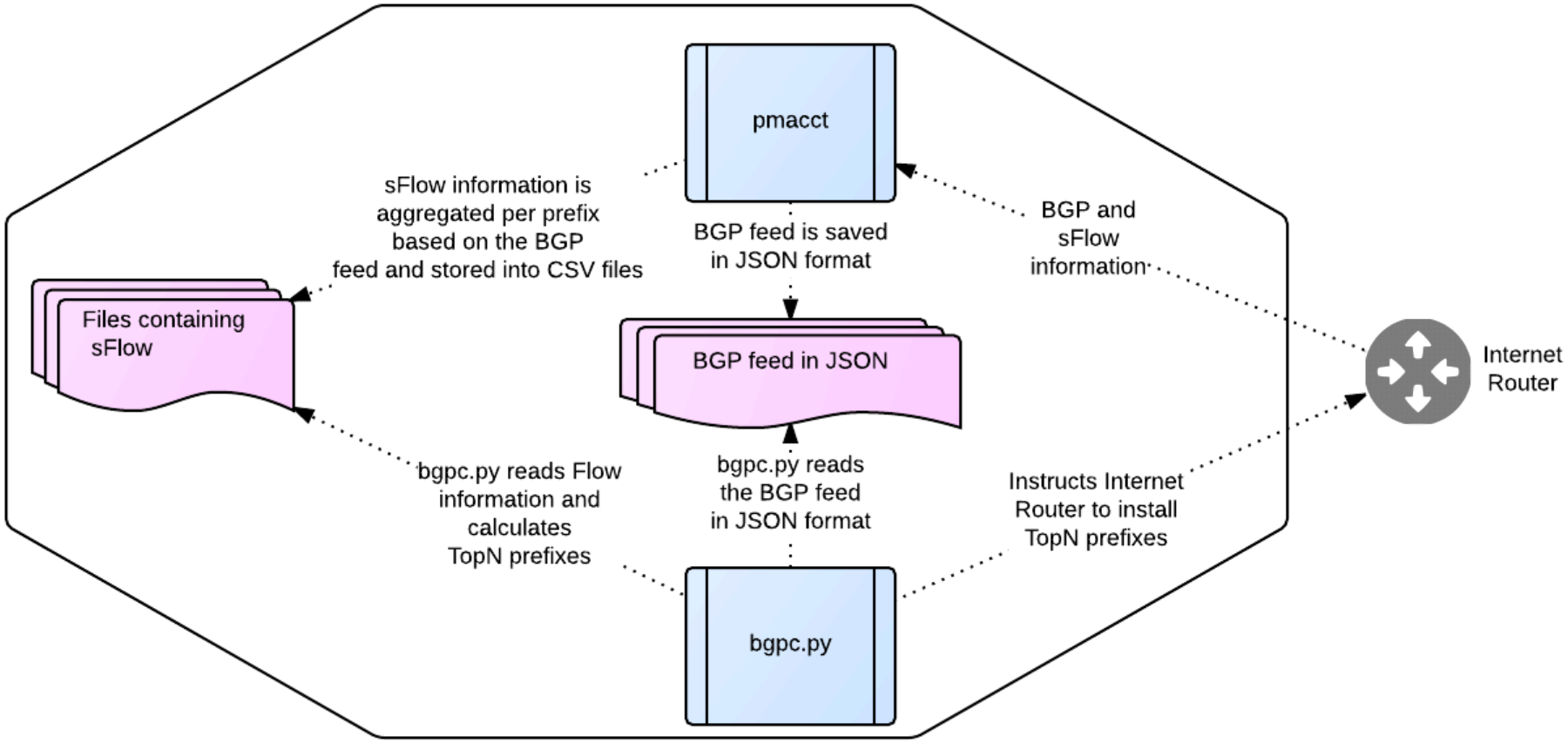
pmacct



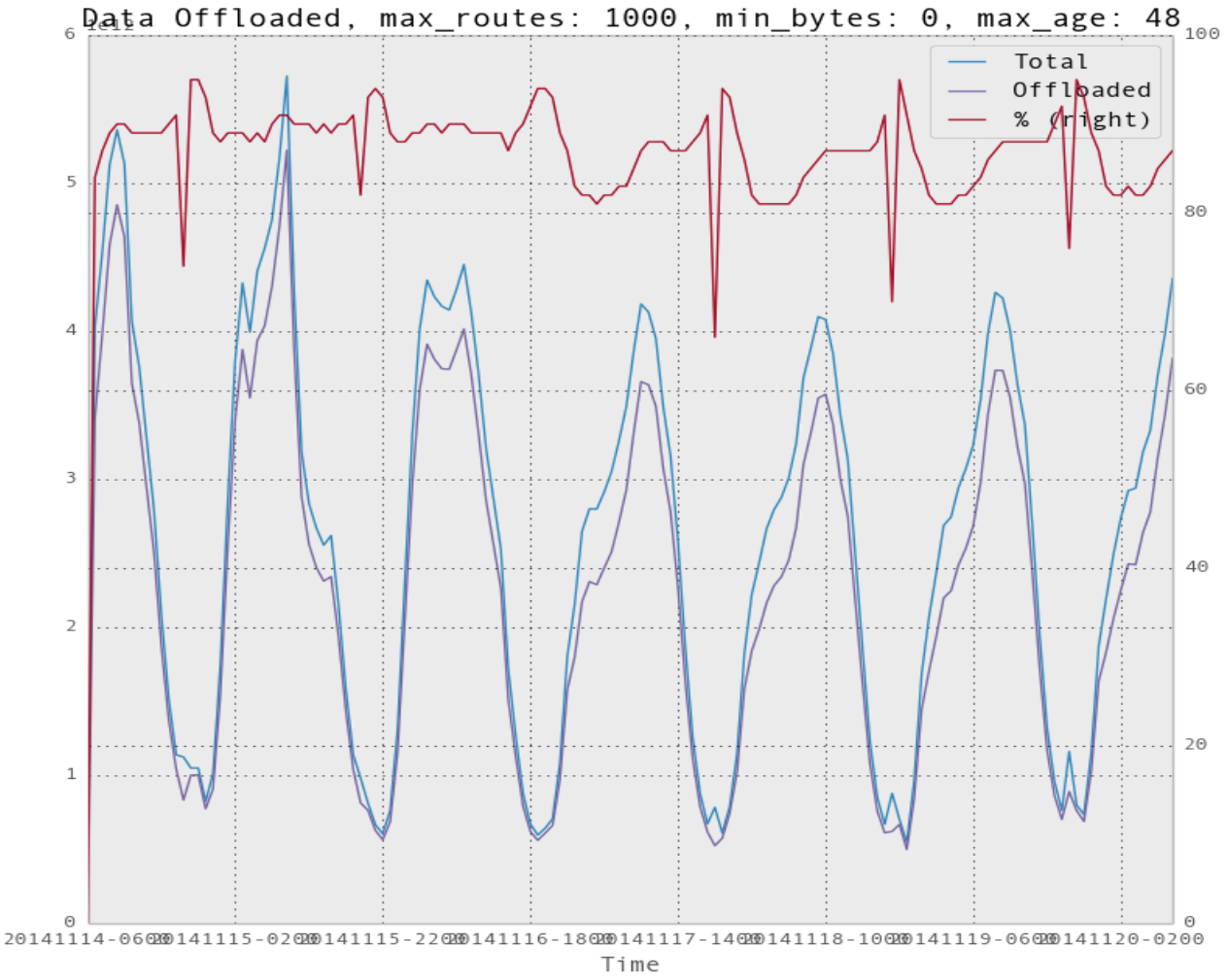
- pmacct aggregates sFlow data using the BGP information previously sent by the Internet Switch
- pmacct reports the flow data to the BGP Controller
- The BGP controller instructs the Internet switch to install those TopN* prefixes

* N is a number close to the maximum number of entries that the FIB of the Internet Switch can support

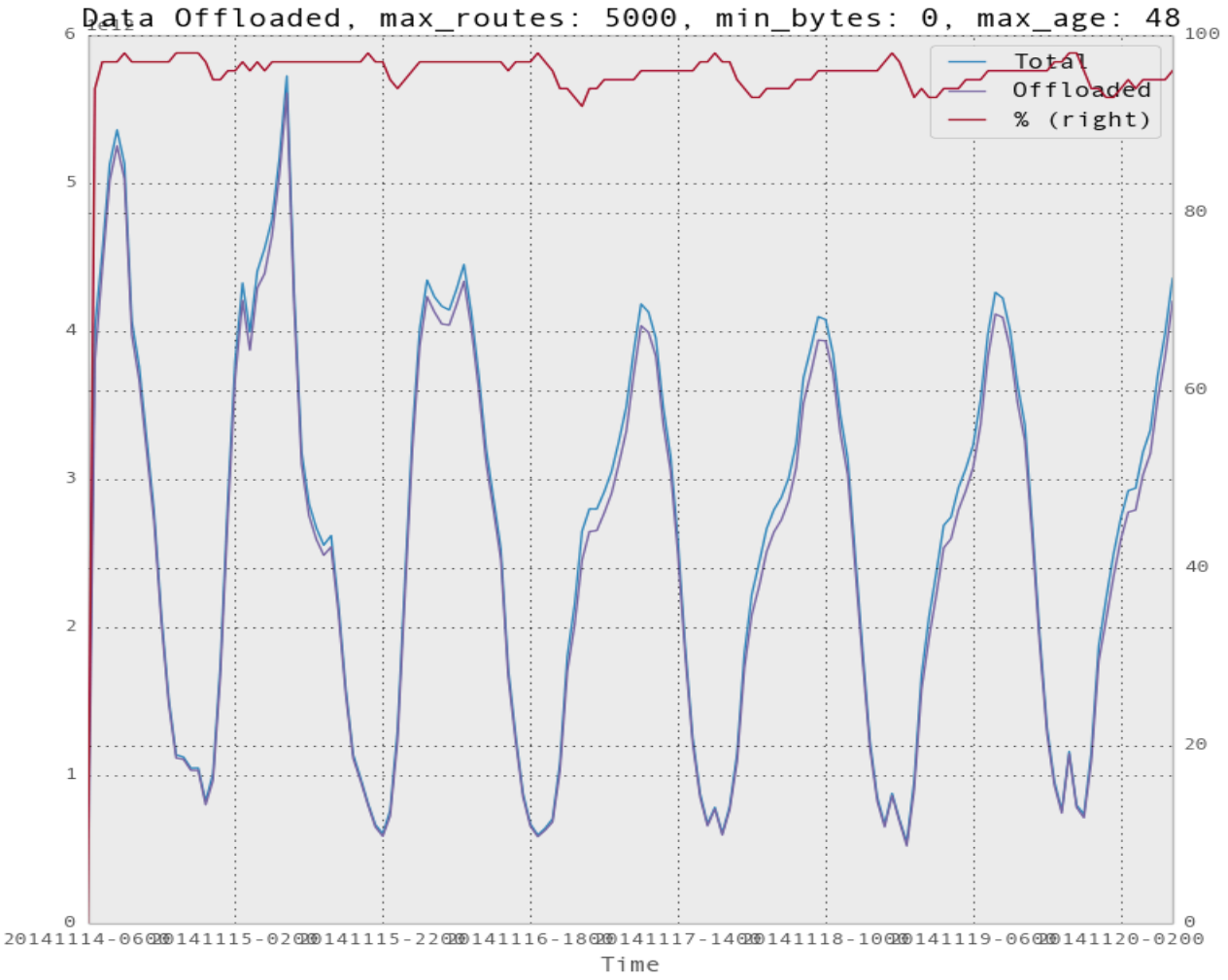
Internals


pmacct
sFlow information is aggregated per prefix based on the BGP feed and stored into CSV files
BGP feed is saved in JSON format
BGP and sFlow information
Files containing sFlow
BGP feed in JSON
Internet Router
bgpc.py reads Flow information and calculates TopN prefixes
bgpc.py reads the BGP feed in JSON format
Instructs Internet Router to install TopN prefixes
bgpc.py

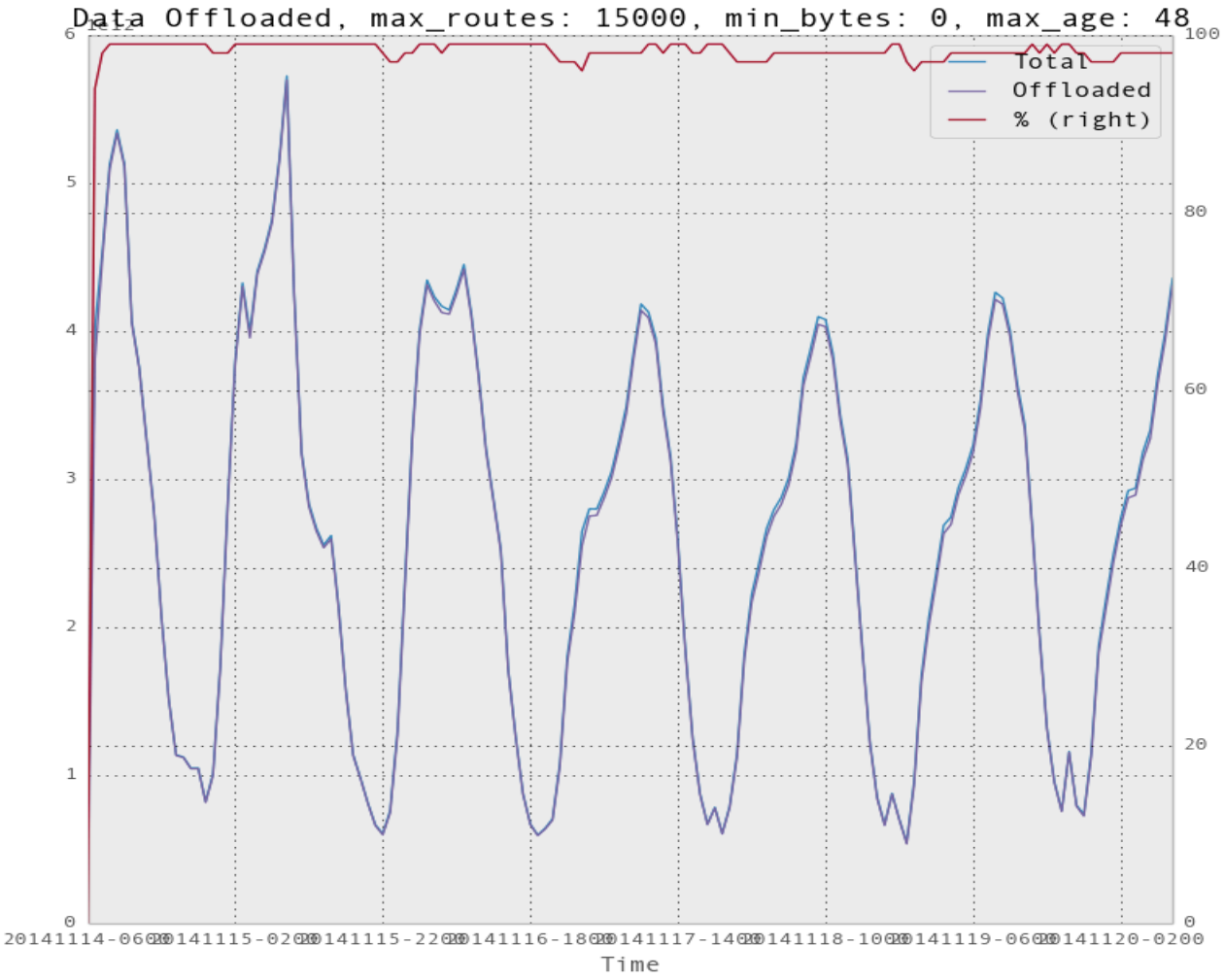
Results: top 1k routes (1/4)



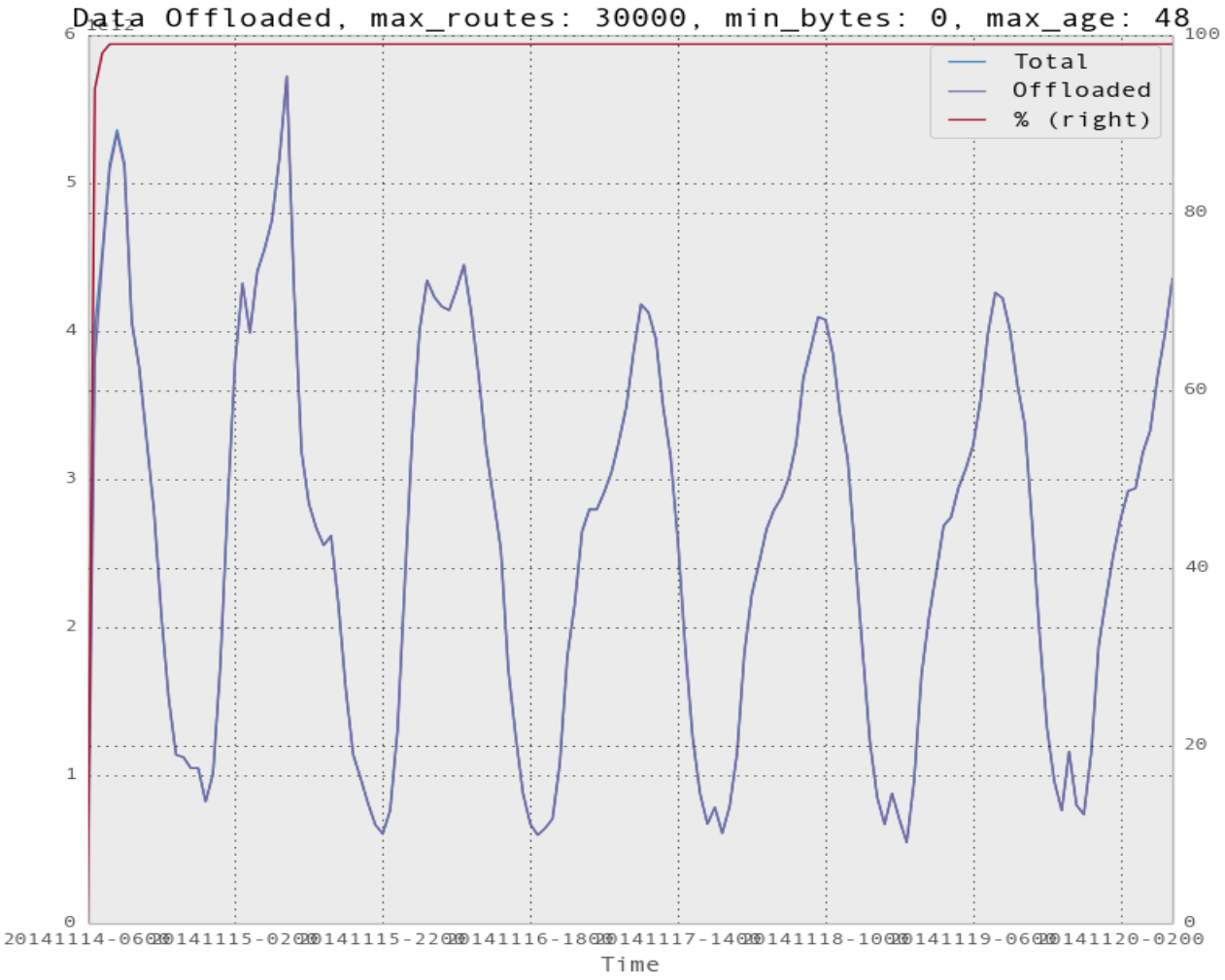
Results: top 5k routes (2/4)



Results: top 15k routes (3/4)



Results: top 30k routes (4/4)



Deployment notes

- Demo run in Spotify Stockholm datacenter, connected to Netnod:
  - Info gathered but no actual changes performed on the Internet Router there
- Pilot to be run very soon by Spotify in cooperation with a major IXP in Europe

Wrap-up

JANOG36 meeting, Kitakyushu – Jul 2015

Acknowledgments

- Elisa Jasinska
  - elisa@bigwaveit.org
- David Barroso
  - dbarroso@spotify.com

Further information (1/2)

- http://www.pmacct.net/dbarroso_plucente_waltzing_v0.5.pdf
  - Full information on the Spotify use-case
- http://www.pmacct.net/nanog61-pmacct-add-path.pdf
  - Full information on the Netflix use-case
- http://www.pmacct.net/Lucente_collecting_netflow_with_pmacct_v1.2.pdf
  - A tutorial on pmacct

Further information (2/2)

- http://www.pmacct.net/lucente_pmacct_uknof14.pdf
  - About coupling telemetry and BGP
- http://ripe61.ripe.net/presentations/156-ripe61-bcp-planning-and-te.pdf
  - About telemetry, traffic matrices, capacity planning & TE
- http://wiki.pmacct.net/OfficialExamples
  - Compiling instructions for pmacct and quick-start guides
- http://wiki.pmacct.net/ImplementationNotes
  - pmacct implementation notes (RDBMS, maintenance, etc.)

オープンソースのネットフローツールの運用

JANOG36 BoF

maoke@bbix.net
paolo@pmacct.net

JANOG36 meeting, Kitakyushu – Jul 2015