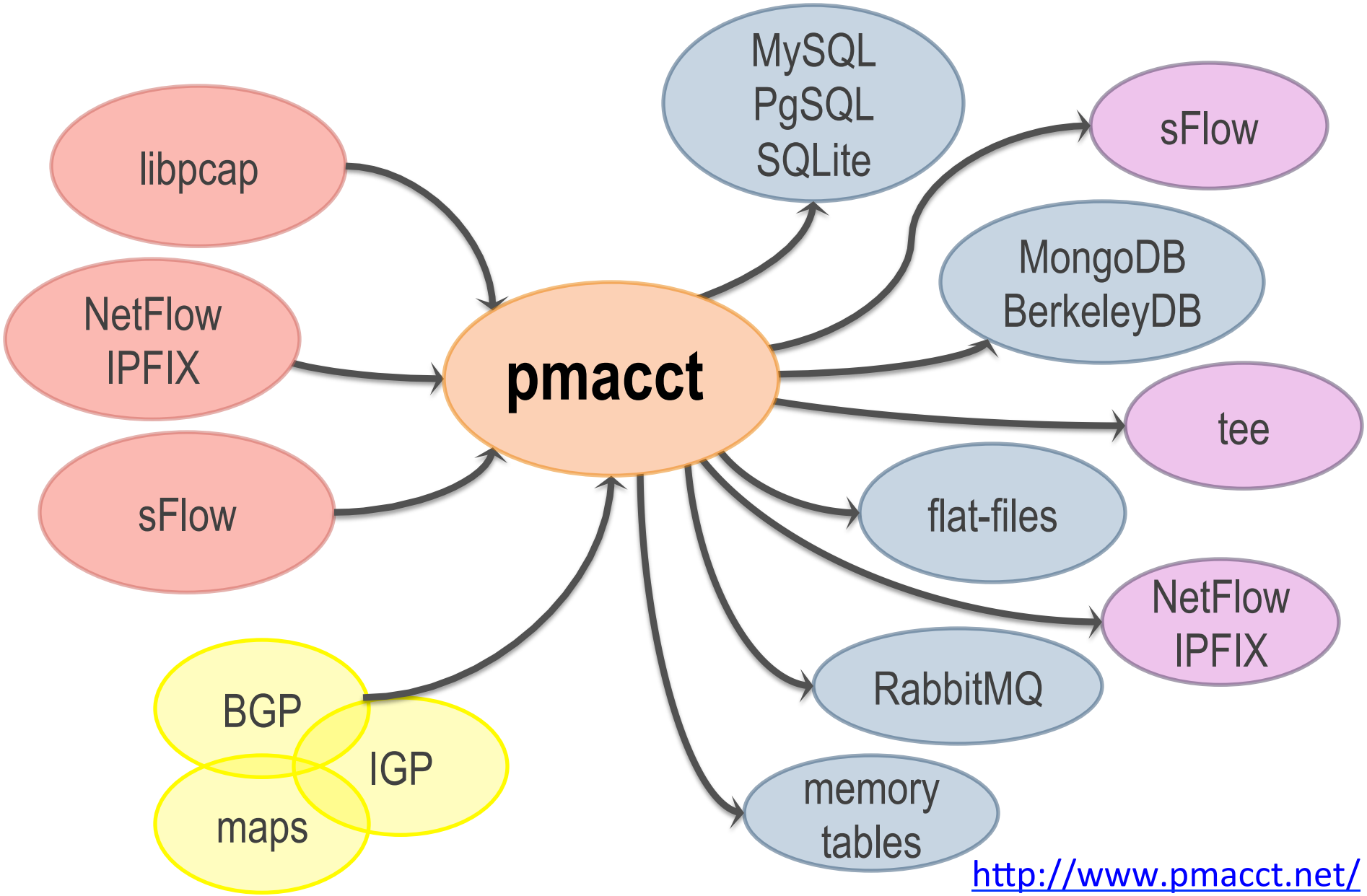
Agenda

- About pmacct
- Spotify use-case
- Netflix use-case

About pmacct

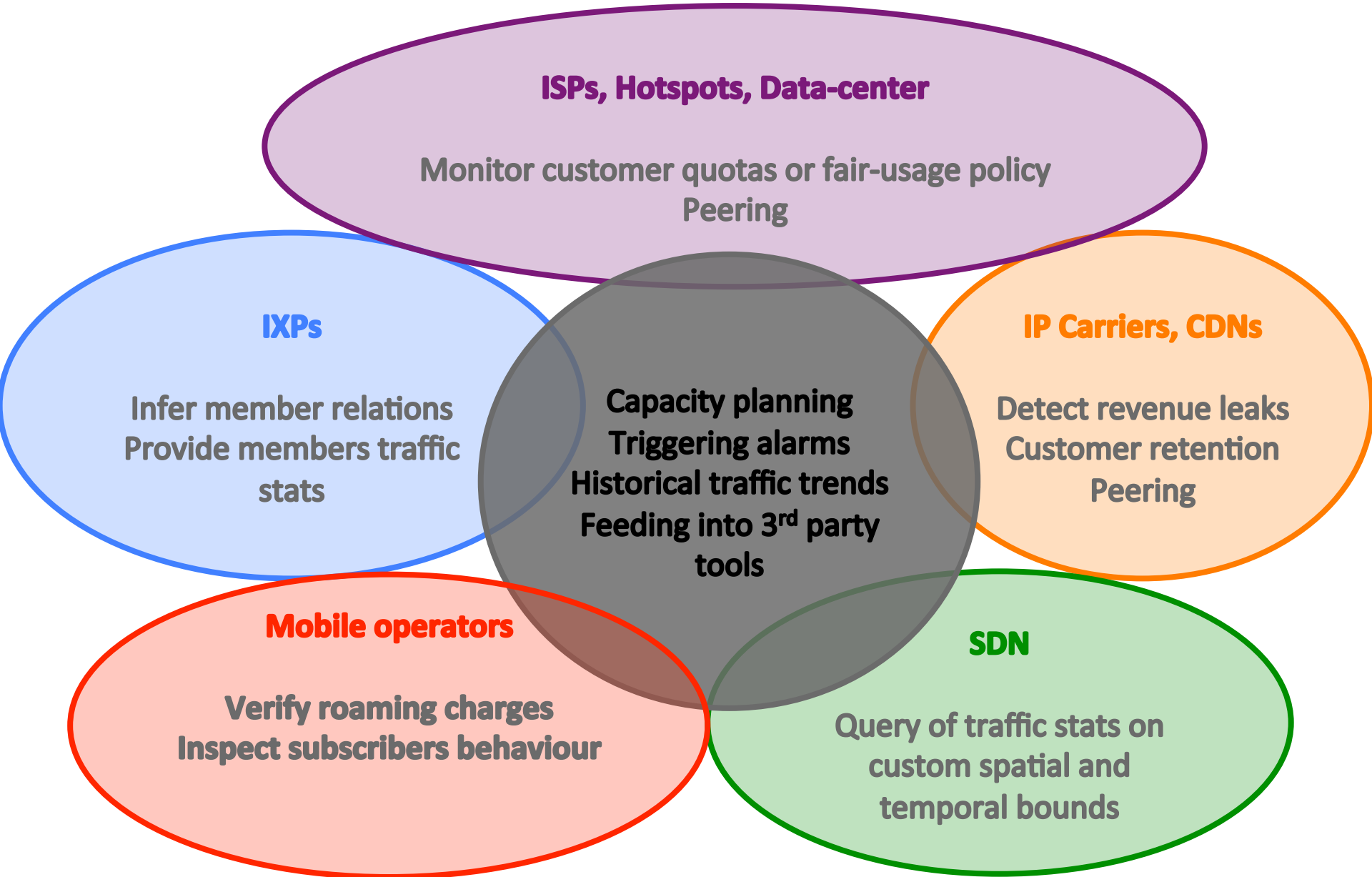
JANOG36 meeting, Kitakyushu – Jul 2015

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



http://www.pmacct.net/

Usage scenarios



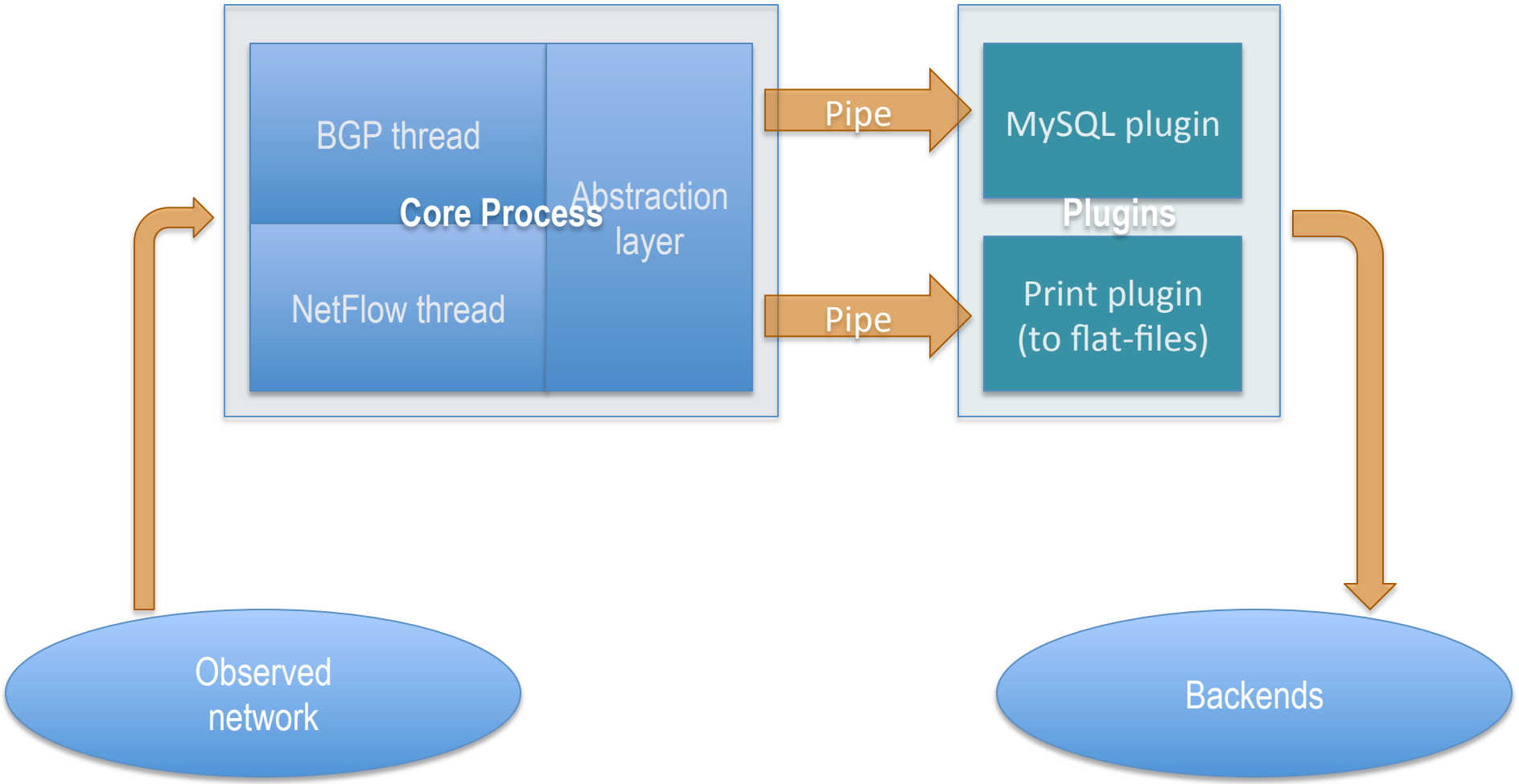
Key pmacct non-technical facts

- 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

BGP integration

- pmacct introduced a Quagga-based BGP daemon
  - Implemented as a parallel thread within the collector
  - Doesn't send UPDATEs and WITHDRAWs whatsoever
  - Behaves as a passive BGP neighbor
  - Maintains per-peer BGP RIBs
  - Supports 32-bit ASNs; IPv4, IPv6 and VPN families
  - Supports ADD-PATH: draft-ietf-idr-add-paths
- Why BGP at the collector?
  - Telemetry reports on forwarding-plane, and a bit more
  - Extended visibility into control-plane information

Brokering data around: RabbitMQ message exchanges

- pmacct opening to AMQP protocol
- noSQL landscape difficult to move through, ie. fragmented and lacks of standardization
- Data can be picked up at the message exchange in the preferred programming/scripting language
- Data can be then easily inserted in the preferred backend, ie. not natively supported by pmacct

Spotify use-case

JANOG36 meeting, Kitakyushu – Jul 2015

About the presenters

- **David Barroso**
  - Network Engineer @Spotify
  - 10+ years in the network industry
  - Python enthusiast
  - Automation junkie
- **Paolo Lucente**
  - Principal Software Developer @pmacct
  - 10+ years measuring and correlating traffic flows
  - Service Providers are his DNA

About Spotify (1/2)

Spotify is a commercial music streaming service providing digital rights management-restricted content from record labels [...] Paid "Premium" subscriptions remove advertisements and allow users to download music to listen to offline.

About Spotify (2/2)

- Over 60M active users per month, 15M paying subscribers, 30M+ songs, 28k songs added per day, available in 58 markets
- Four major datacenters:
  - Stockholm, London, Ashburn, San Jose
- Users are directed to the closest datacenter:
  - In case of fault or maintenance users can be redirected to another DC

FIB vs RIB (1/2)

- RIB (Routing Information Base)
  - A representation in memory of all available paths and their attributes
  - This information is fed by routing protocols
- FIB (Forwarding Information Base)
  - A copy of the RIB (usually in hardware) where some attributes are resolved (like next-hop or outgoing interface)

FIB vs RIB (2/2)

- RIB (Routing Information Base)
  - Virtually unlimited (limited only by the memory of the device)
- FIB (Forwarding Information Base)
  - Limited by the underlying hardware
  - Between 64k-128k LPM prefixes in modern switches with commodity ASIC
  - Between 500k-1000k LPM prefixes in expensive routers/switches with customized ASICs

The Internet

- +500k prefixes
- Too many to fit them in commodity ASICs, ie.:
  - Trident 2 supports 32k prefixes
  - ARAD supports 64k prefixes

When you travel … (1/2)

- Do you carry an atlas?
- Or do you carry a local map?

So .. (granted I'm close to content or eyeballs, ie. I'm not in the business of routing the internet for 3rd parties):

- Why do I need all the prefixes?
- What if I only install the prefixes I really need?

When you travel … (2/2)

- 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 Serbia?

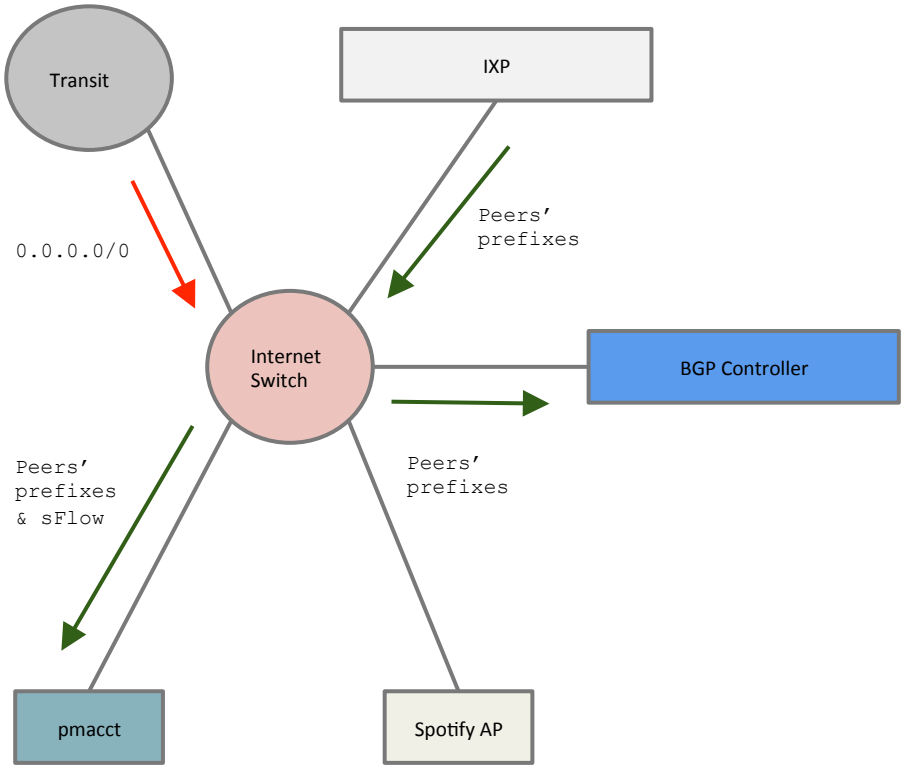
Goal of our work

- 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

Two key components of our work

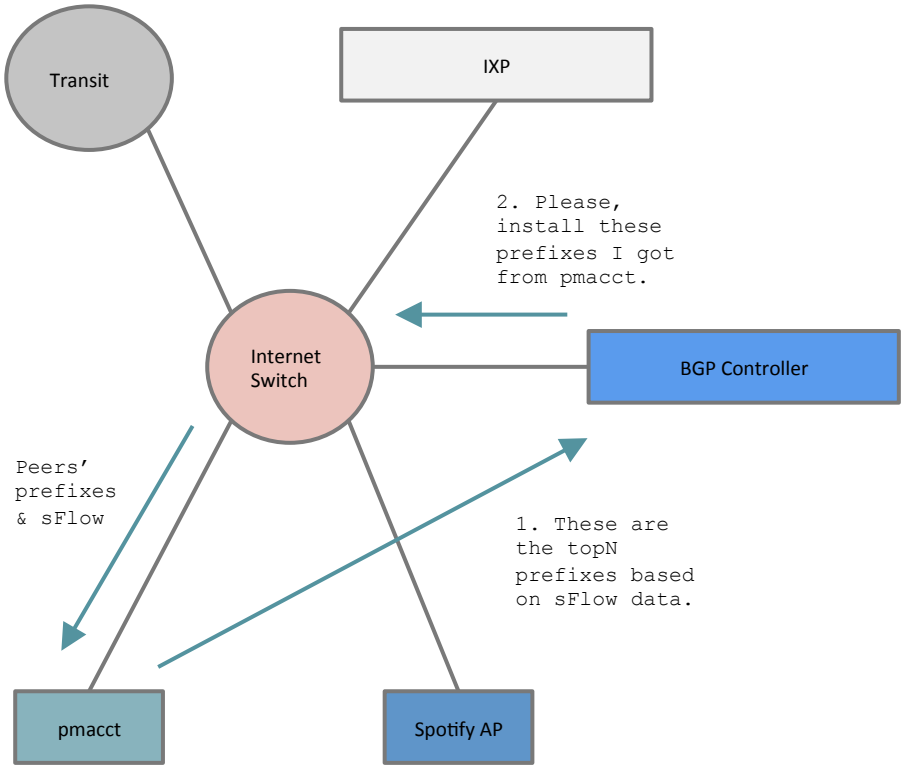
- **pmacct** - Collector that can aggregate traffic by network, AS, BGP peer, etc. BGP information can be obtained by peering with other routers (more later)
- **Selective route download** - Feature that allows to pick a subset of the routes on the RIB and install them on the FIB.

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 and the BGP Controller
- pmacct will receive in addition sFlow data

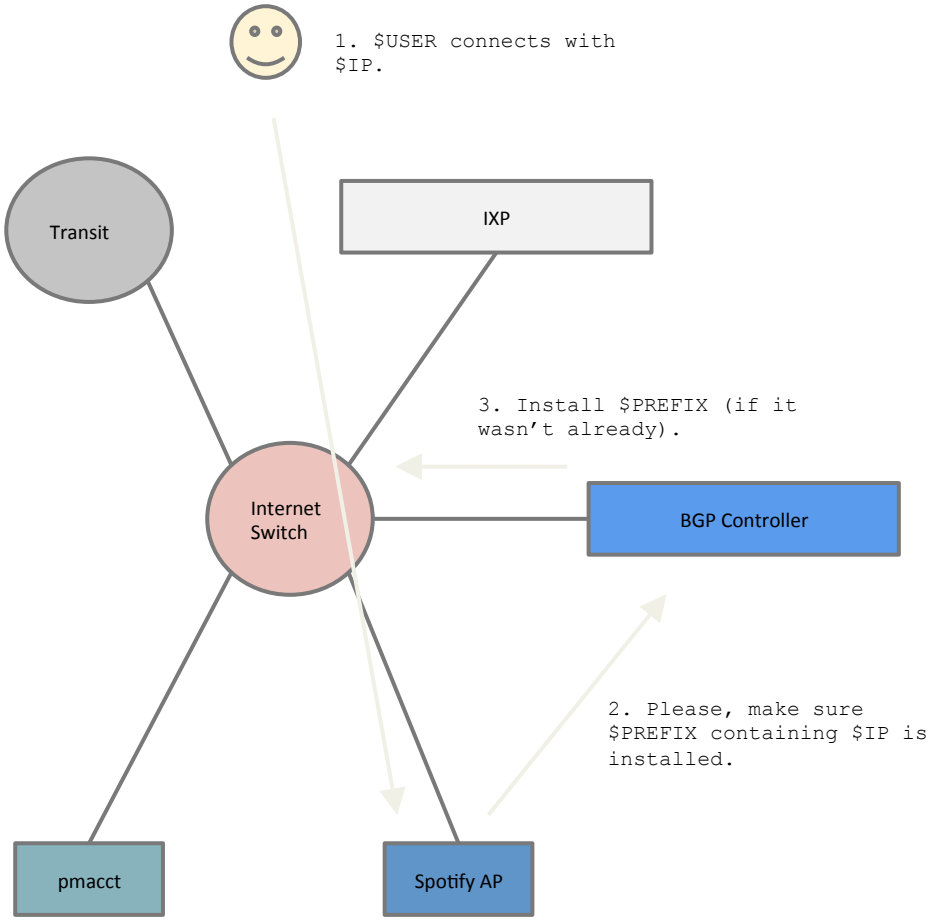
pmacct



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

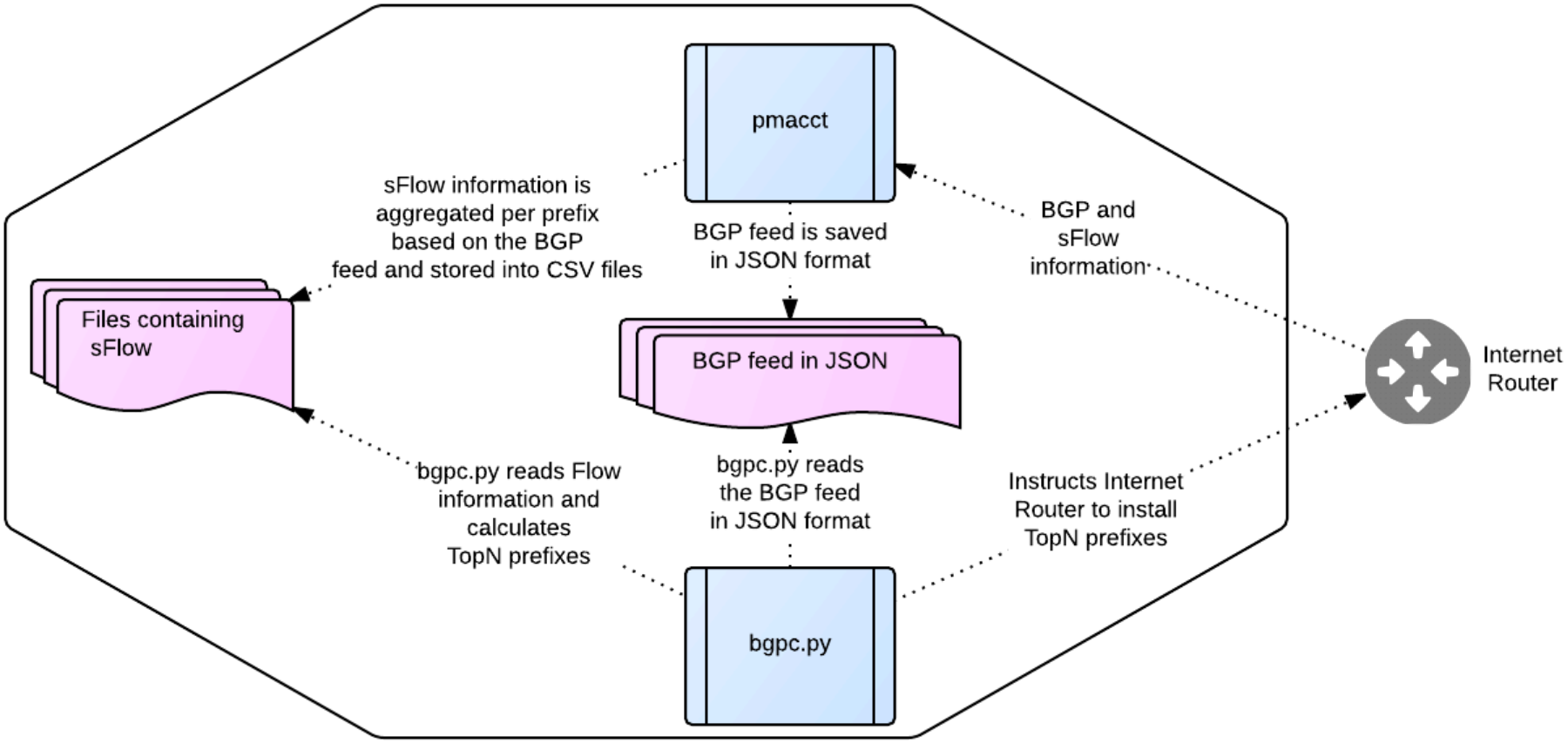
* N is a number close to the maximum number of entries that the FIB of the Internet Switch can support

Spotify AP

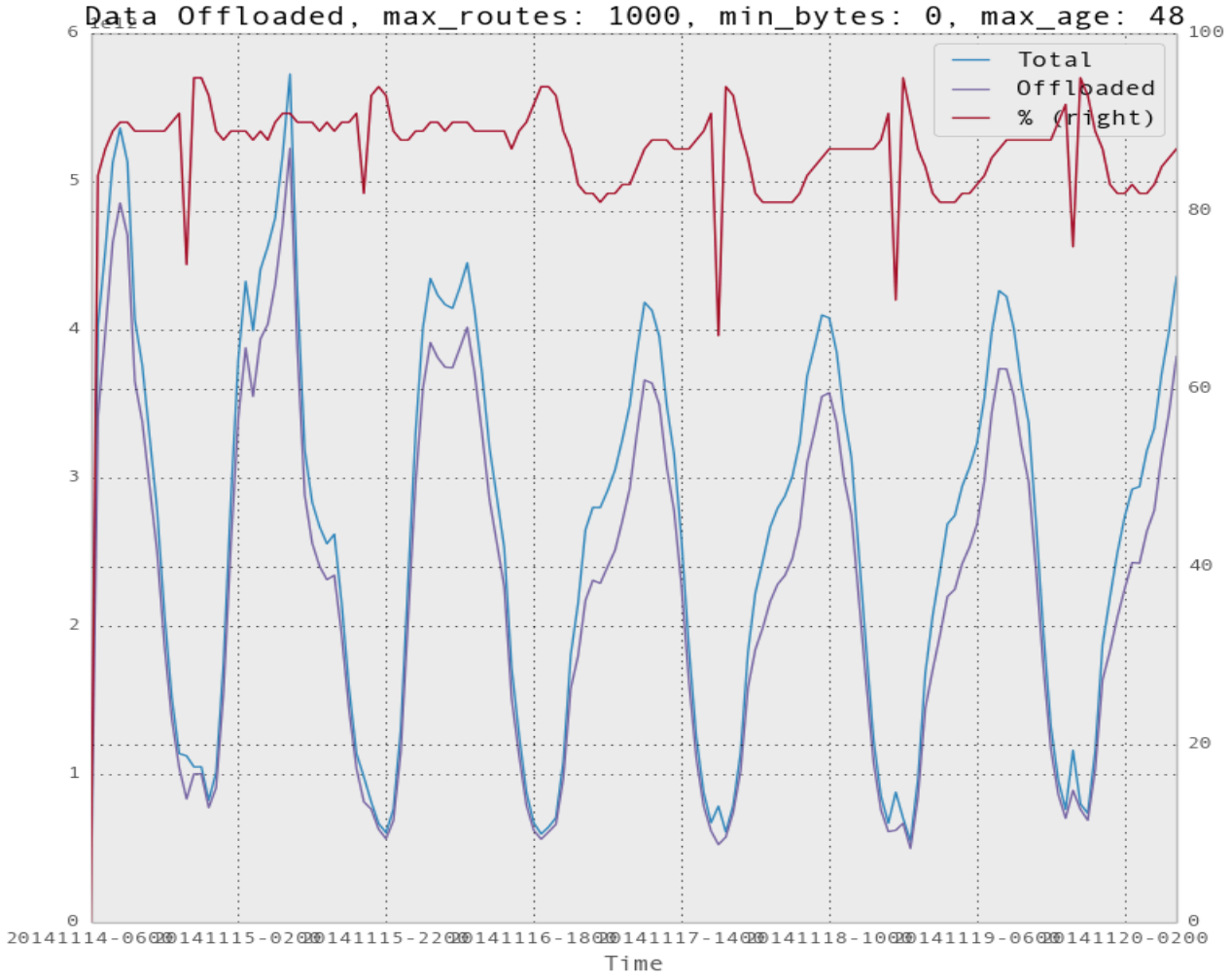


- $USER connects to the service
- The application informs the access point that $USER has connected and requests that the $PREFIX containing his/her $IP is installed on the FIB
  - It might be installed already as another user within the same range might have connected previously or because pmacct reported that prefix as being one of the TopN prefixes
- The BGP controller instructs the Internet Switch to install the prefix if necessary

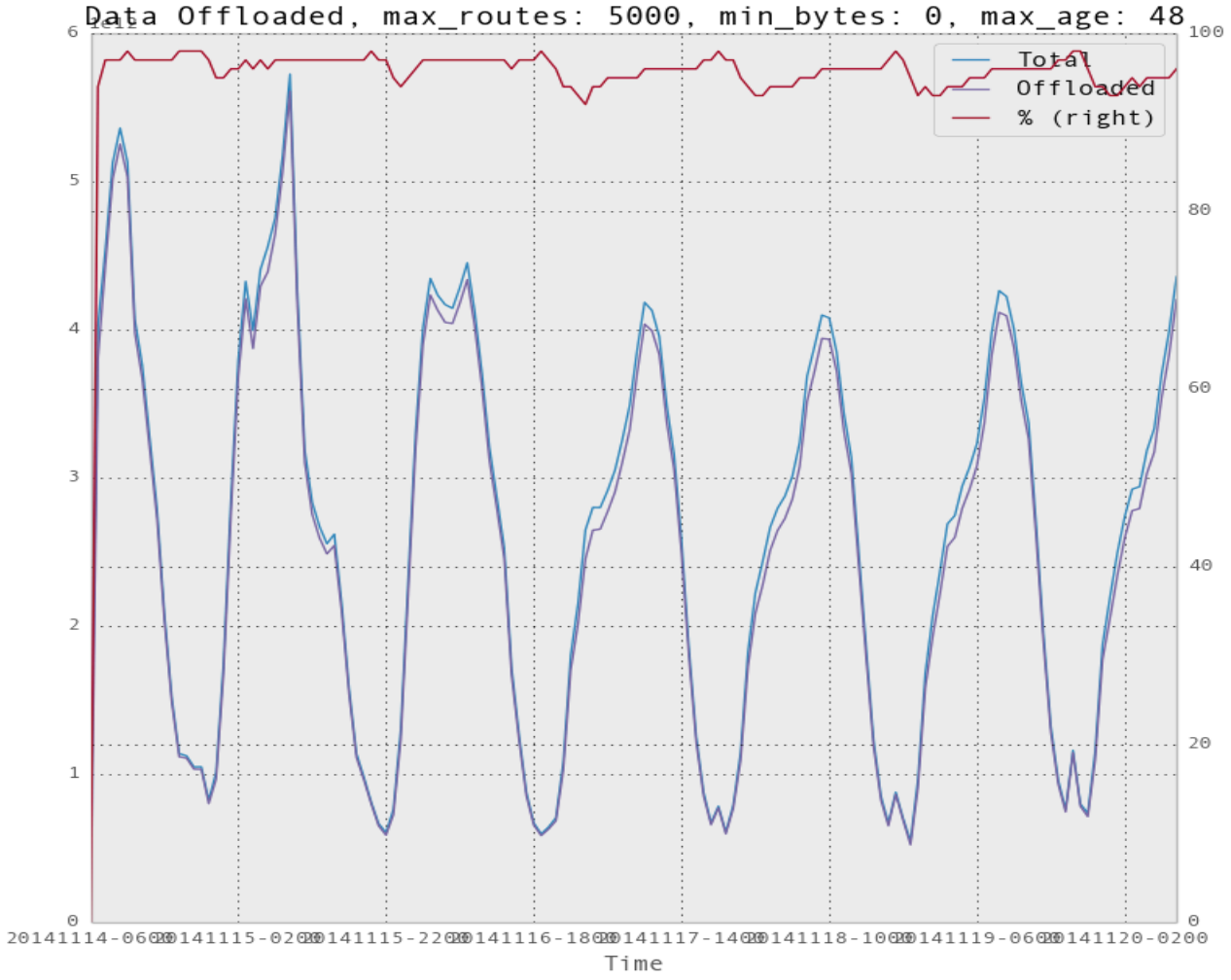
Internals


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

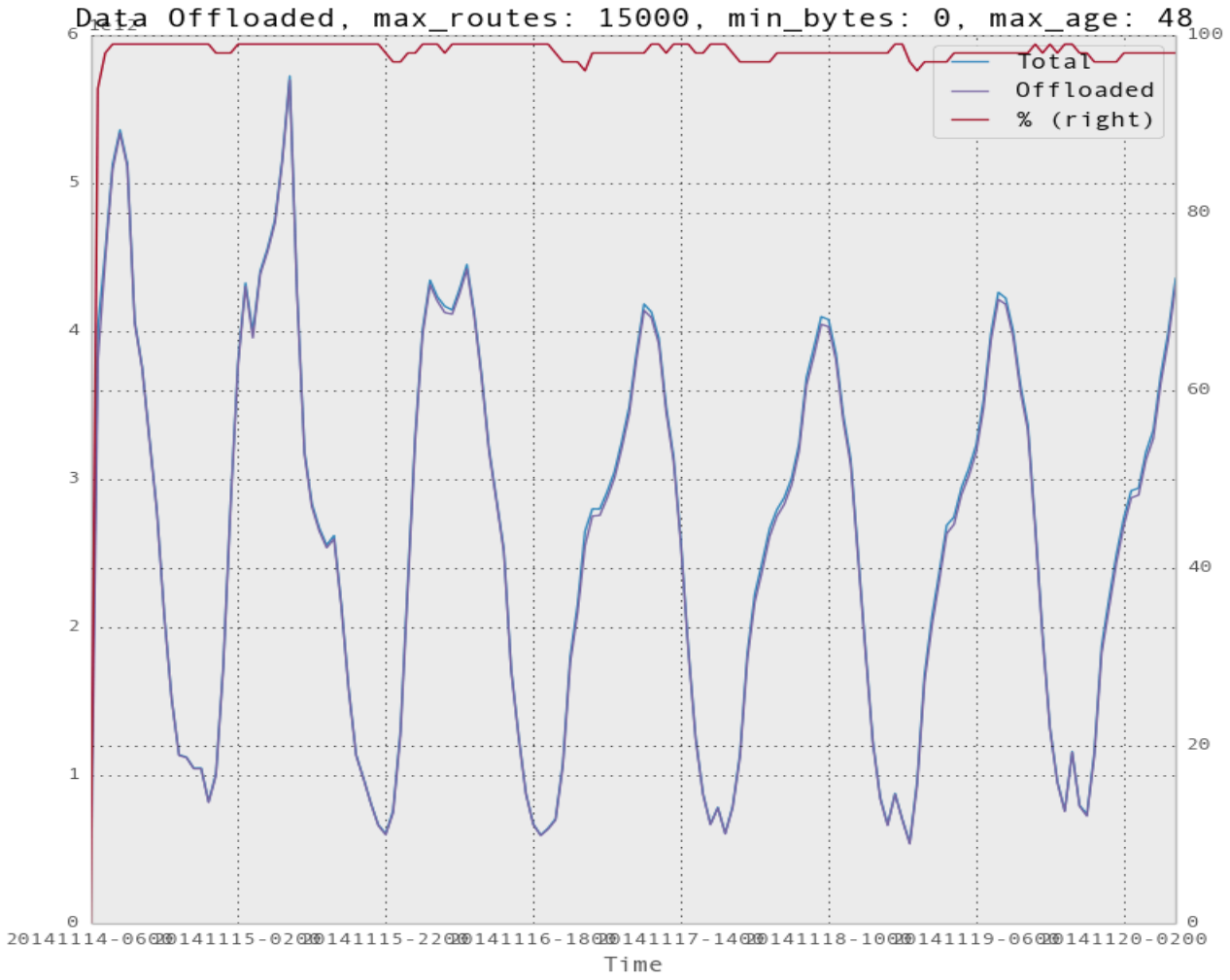
Results: top 1k routes (1/4)



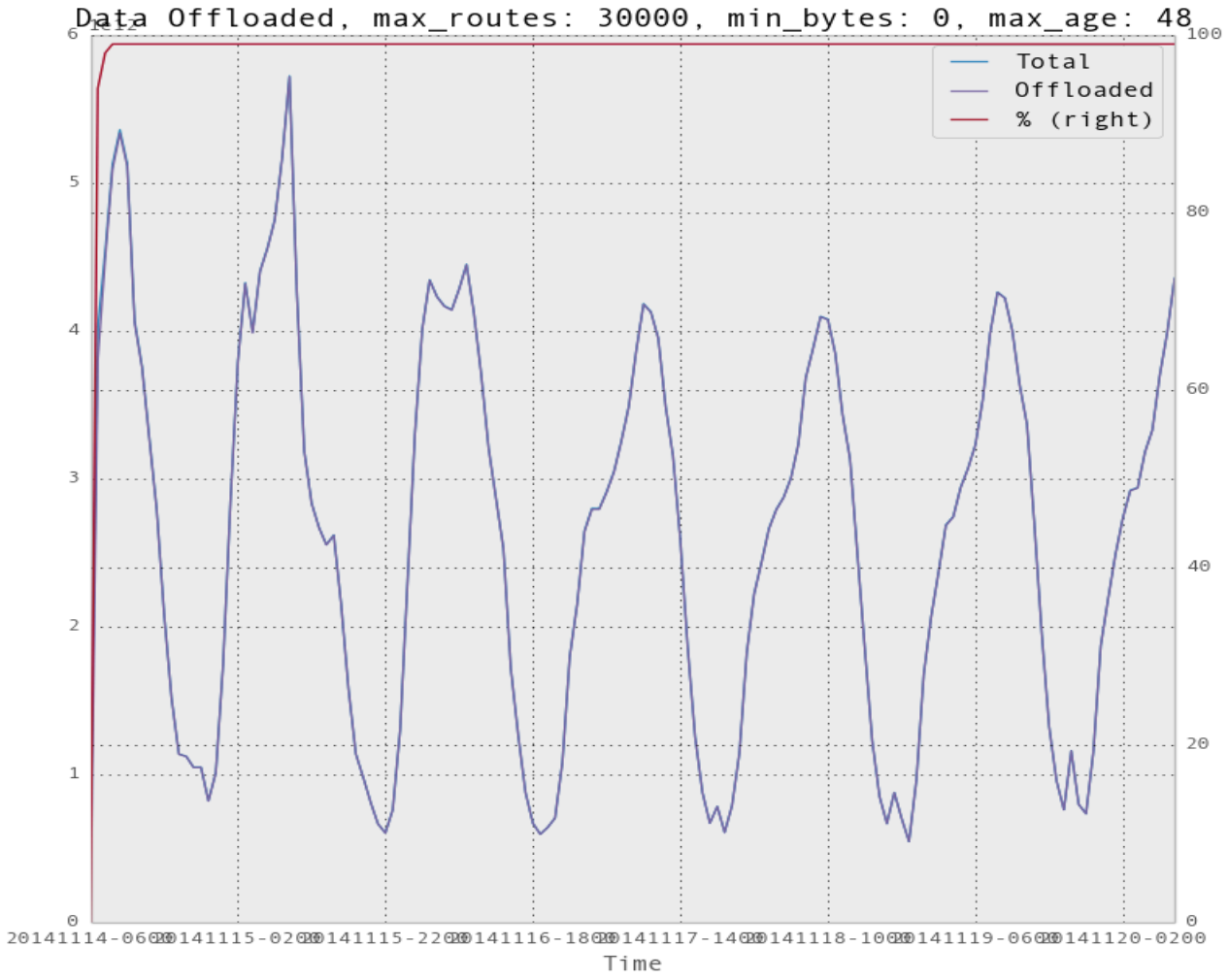
Results: top 5k routes (2/4)



Results: top 15k routes (3/4)



Results: top 30k routes (4/4)



Considerations

- The BGP controller updates a prefix list containing the prefixes that the device must take from the RIB and install on the FIB (that is, **selective route download** applied):
  - If a prefix is removed from the RIB it will be removed from the FIB by the device
  - If the BGP controller fails the prefix list remains in the device. Allowing the device to operate normally as per the last instructions

Present and future (1/2)

- 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

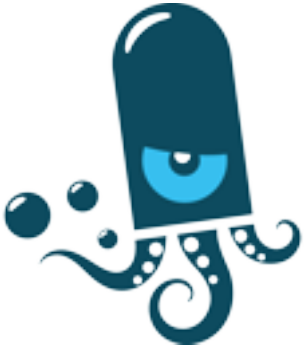
Present and future (2/2)

- The BGP controller only computes top prefixes and passes all the information used and the results to plugins
- Plugins can in future do with this information whatever they want:
  - Build reports
  - Build a prefix list and send it to a router
  - Compare possible next-hops, AS PATH's with other active/passive measurements to choose peers based on reliability, latency, etc.

```
# cat etc/config.yaml

max_age: 48
csv_delimiter: ";"
max_routes: 30000
min_bytes: 0
packet_sampling: 10000

… (output omitted)

plugins:
  - 'prefix_data.SavePrefixData'
  - 'statistics.RouteStatistics'
  - 'statistics.OffloadedBytes'
  - 'bird.Bird'

… (output omitted)
```

Netflix use-case

NETFLIX



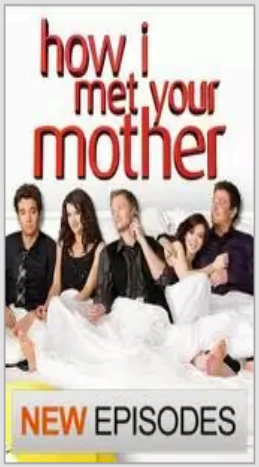
JANOG36 meeting, Kitakyushu – Jul 2015

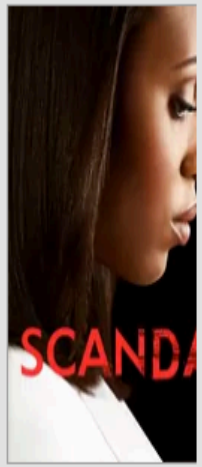
Netflix use-case agenda

- About Netflix
- Brief digression on BGP ADD-PATHS
- Putting all the pieces together

About Netflix

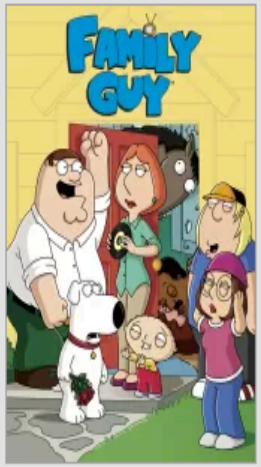
Popular on Netflix





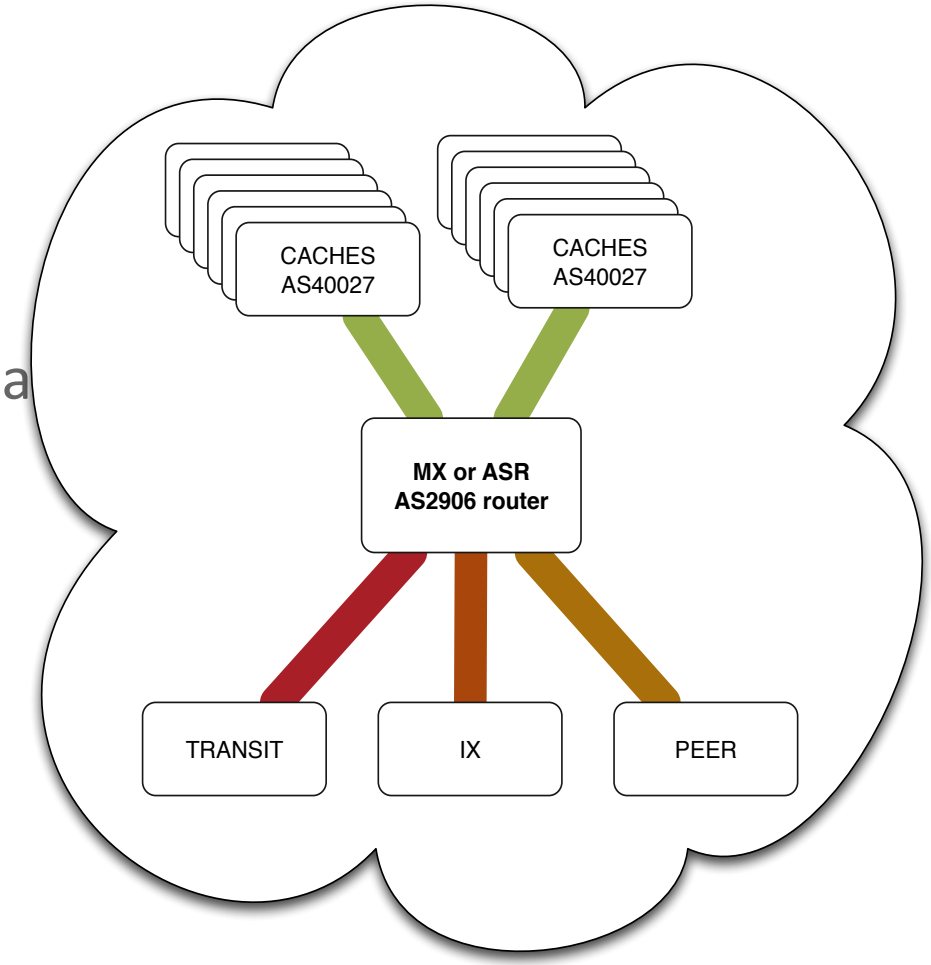
Emmy-winning TV Shows

Based on your interest in...



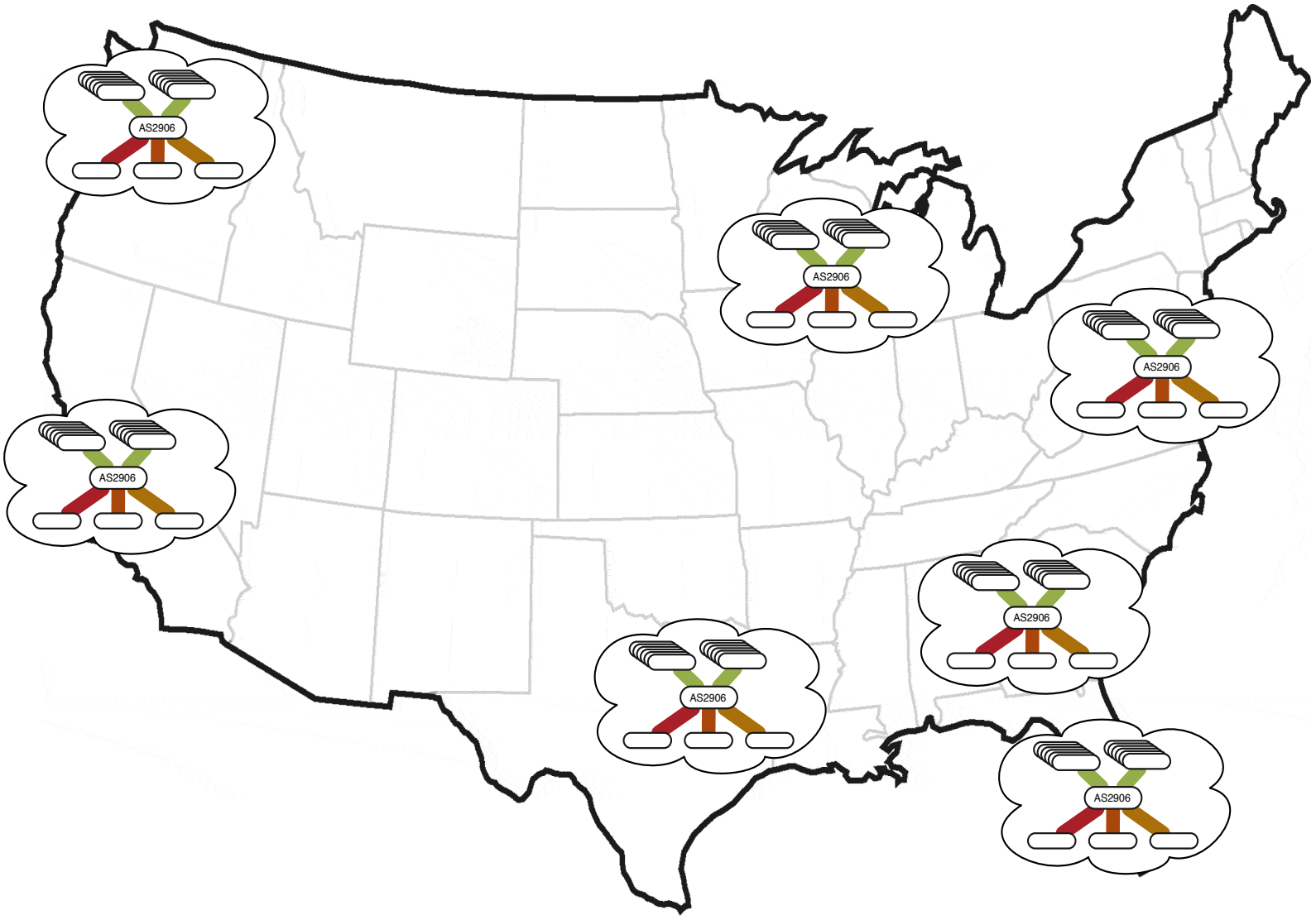
Netflix CDN: Open Connect

- In house CDN
- Designed for efficient video delivery
  - Many POPs
  - No backbone
- Hardware: ASR, MX and Arista 7500e
- Delivery via:
  - Servers embedded in access network
  - Peering
  - Transit



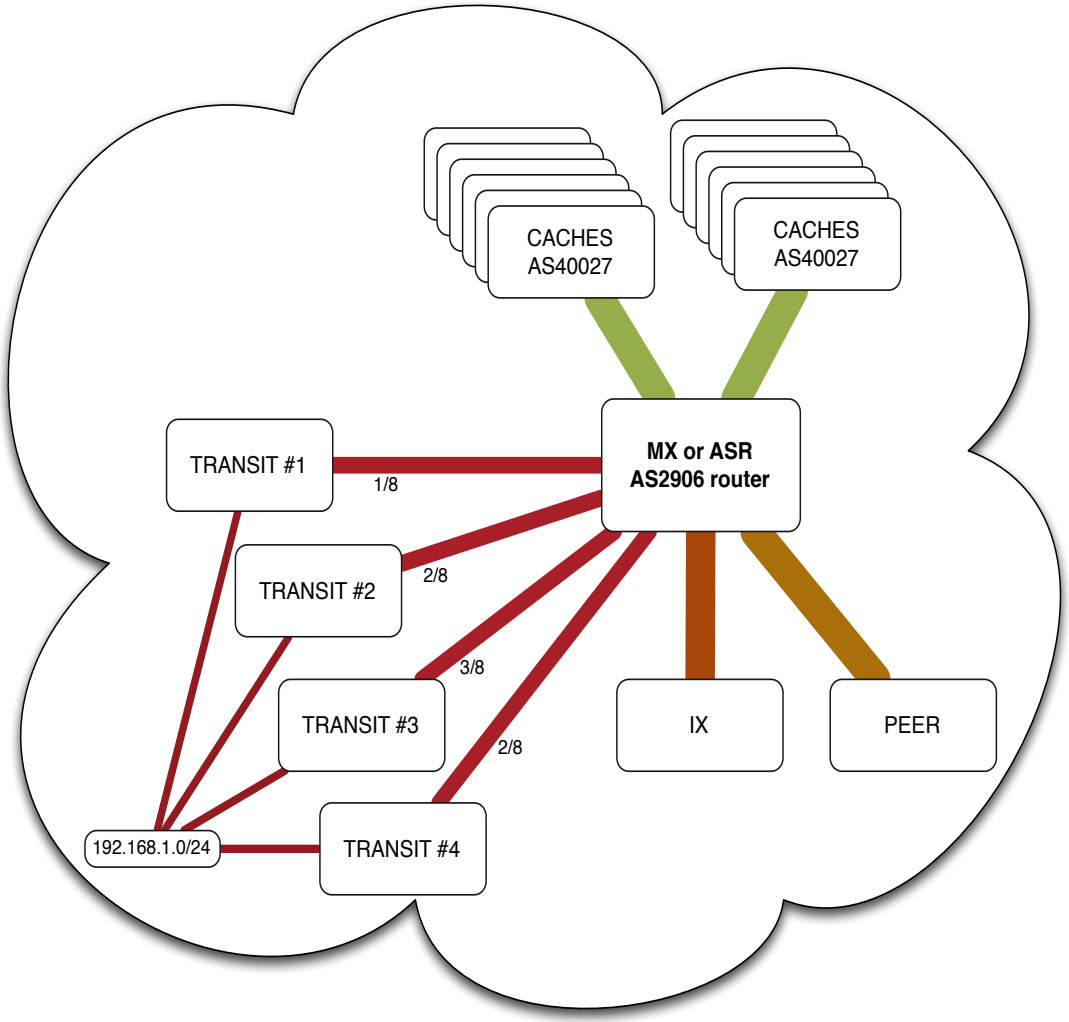
https://www.netflix.com/openconnect

Network Design at Netflix



Egress BGP Hacks

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



Flow Accounting at Netflix

- Primary goal: peering analysis
  - How much traffic is being exchanged with which ASN?
  - How do they perform?
- Software: pmacct
  - NetFlow/IPFIX augmented by BGP using pmacct
- Problem: multi-path, not only one single best path

Brief digression on BGP ADD-PATHS

On BGP ADD-PATHS

- 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

On BGP ADD-PATHS

- New BGP capability, new NLRI encoding:

```
+--------------------------------+
| Path Identifier (4 octets)     |
+--------------------------------+
| Length (1 octet)               |
+--------------------------------+
| Prefix (variable)              |
+--------------------------------+
```

- Capability number: 69

On BGP ADD-PATHS

- BGP ADD-PATHS covers several use cases:
  - Mostly revolving around actual routing
  - Extra path flooding questioned in such context (*)
- Our use-case for BGP ADD-PATHS is around monitoring applications:
  - Not much talk yet in such context
  - Proposal to mark best-paths to benefit monitoring applications: draft-bgp-path-marking (Cardona et al.)

(*) http://www.nanog.org/meetings/nanog48/presentations/Tuesday/Raszuk_To_AddPaths_N48.pdf

Putting all the pieces together: NetFlow and BGP ADD-PATHS with pmacct at Netflix

Wait, so what's the problem?

- BGP multi-path, traffic not only sent to a single best path
- pmacct is 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-PATHS FTW!

- ADD-PATHS provides visibility into the N 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
```

pmacct and BGP ADD-PATHS

- In early Jan 2014 pmacct BGP integration got support for BGP ADD-PATHS
  - GA as part of 1.5.0rc3 version (Apr 2014)
- Why BGP ADD-PATHS?
  - Selected over BMP since it allows to not enter the exercise of parsing BGP policies
  - True, post-policies BMP exists but it's much less implemented around and hence not felt the way to go

NetFlow/IPFIX and BGP ADD-PATHS

- 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-PATHS

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
```

Netflix + NetFlow/IPFIX + pmacct + ADD-PATHS

- Multiple pmacct servers in various locations
- NetFlow is being exported to the pmacct servers:
  - Mix of NetFlow v5, v9 and IPFIX
- 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

Wrap-up

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.)

Use of Flow-Routing Combination

JANOG36 BoF

maoke@bbix.net
paolo@pmacct.net