Monitoring a EVPN-VxLAN fabric with BGP Monitoring Protocol

Davide Pucci
davide.pucci@os3.nl

Giacomo Casoni
giacomo.casoni@os3.nl

Vivek Venkatraman
Cumulus Networks

Attilla de Groot
Cumulus Networks

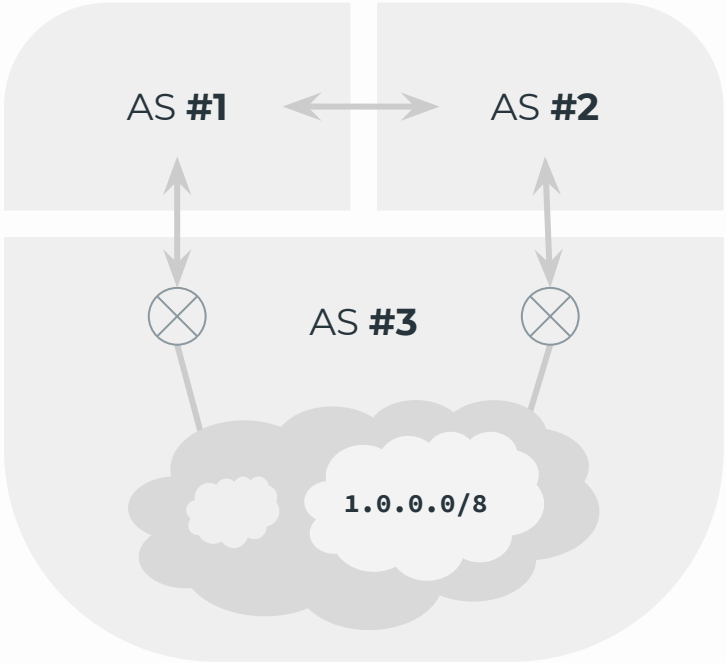
Donald Sharp
Cumulus Networks

Border Gateway Protocol (BGP)

BGP is the *de-facto* Internet **routing protocol**.

Pulls intra-Autonomous System prefixes, relying on **iBGP**.

Exchanges these internal prefixes with neighbouring Autonomous Systems to enable proper routing, relying on **eBGP**.



BGP in Data Centers

Third-wave applications moved most of the traffic to a **east-west direction**.

This change introduced the need of more elastic **Data Centers**.

All the switches represent a (**private**) **Autonomous System**.

RFC 7938

Use of BGP for Routing in Large-Scale Data Centers

August 2016

BGP-based tunneling with EVPN / VxLAN

MP-BGP introduced the possibility to extend BGP behaviour.

Ethernet Virtual Private Network (**EVPN**) makes use of it to build an overlay network relying on the physical structure, by adopting Virtual Extensible LANs (**VxLAN**), encapsulating **layer-2** VLAN-like packets in **layer-4** messages.

RFC 7209

Requirements for Ethernet VPN (EVPN)

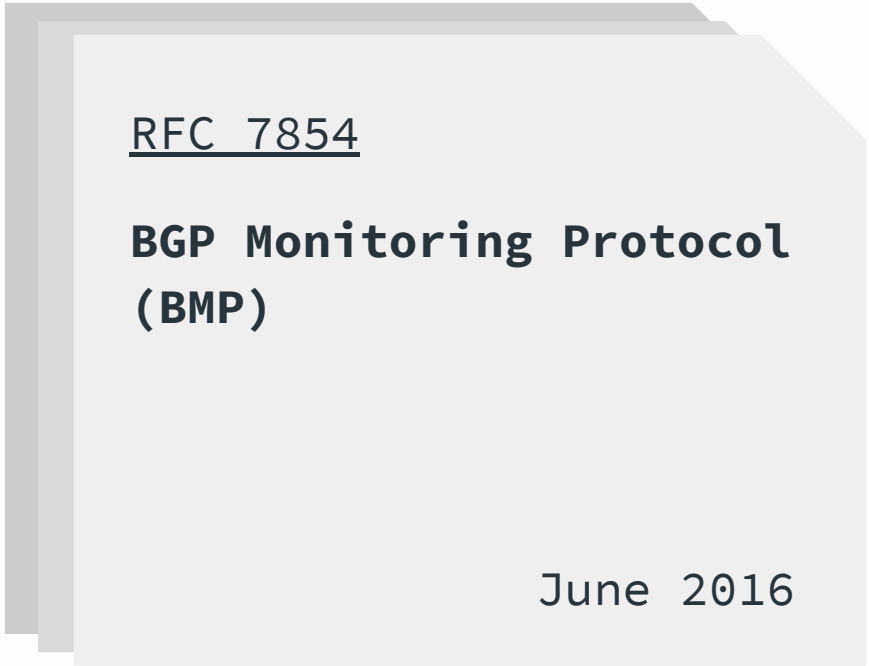
May 2014

BGP monitoring solutions

- ➔ Route collectors: ad-hoc BGP peering sessions
  - not scalable, no information regarding actual routes received
- ➔ Screen scraping
  - manual, not feasible for our use case
- ➔ IP duplication
  - lack of filtering options, TCP stream reassembling

Monitoring BGP

BGP Monitoring Protocol (**BMP**) is a BGP extension which makes BGP speakers forward BGP packets to BMP servers.



Monitoring BGP / Dual mode

Monitoring mode

Once BMP session is up, the client sends all the routes stored in the `Adj-RIB-In` (`-out`) of those peers using standard `BGP Update` messages, encapsulated in `Route Monitoring` messages. Ongoing monitoring is done by propagating route changes in `BGP Update PDU`s as well.

Mirroring mode

Mainly for troubleshooting purpose, this mode provides full-fidelity view of all messages received from its peers, without state compression: as soon as the client receives / generates a raw BGP packet, it sends it out to the BMP server.

Is **BMP** an effective solution for **monitoring** **EVPN**-based overlay networks?

BMP applicability / Use cases

Main bulk of BGP monitoring research focused on BGP **prefix hijacking**

➔ given the usual applications of **EVPN-VxLAN**, such case is not relevant to our research

The following monitoring use cases have being identified instead:

VM movements history	**MAC flapping**
Infrastructure **convergence time** estimations	Inconsistencies in **MAC Mobility counters**
BGP sessions status	**Prefixes authority** history

BMP applicability / Collectors and requirements

BMP is a fairly new protocol: it is still **lacking of open implementations**

- ➔ **OpenBMP** has questionable EVPN support
- ➔ **Wireshark** capable of parsing BMP, but allowing limited capabilities

A custom solution is needed, to achieve the following capabilities:

- ➔ **parsing** BMP / BGP EVPN messages: other protocols not important for the presented use case
- ➔ **analyze** and **draw statistics** from data
- ➔ **visualize** results

It has been built in **Python**, using the **ELK** (ElasticSearch and Kibana) stack, for storage and debugging [1]

[1] **EVPN-BMP-Listener** (https://github.com/giacomo270197/EVPN-BMP-Listener)

BMP applicability / FRR-based client

The FRR suite already implemented BMP, but only to track IP uni/multicast routes.

➔ we extended the FRR suite to make BMP support this use case as well [2]

```
From: streambinder <posta@davidepucci.it>
Date: Tue, 16 Jun 2020 14:50:37 +0200
Subject: [PATCH] bgpd: bmp: add support for L2VPN/EVPN routes
---
 bgpd/bgp_bmp.c | 122
++++++++++++++++++++++++++++++++++++----------
 bgpd/bgp_bmp.h |  10 +++-
 2 files changed, 105 insertions(+), 27 deletions(-)

diff --git a/bgpd/bgp_bmp.c b/bgpd/bgp_bmp.c
index fb4c50e3e..7c4746948 100644
--- a/bgpd/bgp_bmp.c
+++ b/bgpd/bgp_bmp.c
@@ -164,9 +174,16 @@ static uint32_t bmp_qhash_hkey(const
struct bmp_queue_entry *e)
    key = prefix_hash_key((void *)&e->p);
    key = jhash(&e->peerid,
              offsetof(struct bmp_queue_entry, refcount) -
              offsetof(struct bmp_queue_entry, peerid), key);
+   if (e->afi == AFI_L2VPN && e->safi == SAFI_EVPN)
+       key = jhash(&e->rd,
+                   offsetof(struct bmp_queue_entry, rd) -
+                   offsetof(struct bmp_queue_entry, refcount) +
+                   PSIZE(e->rd.prefixlen), key);
+
+ ...
```

[2] **lib: prefix: add prefix_rd type** (https://github.com/FRRouting/frr/pull/6582)
bgpd: bmp: add support for L2VPN/EVPN routes (https://github.com/FRRouting/frr/pull/6590)

BMP applicability / FRR-based client contd

IP RIB

SUPPORTED

It is a normal table collecting all the routes announced over IP (v4 / v6).

```
192.168.0.0/24 via swp1
192.168.1.0/24 via swp2
...
```

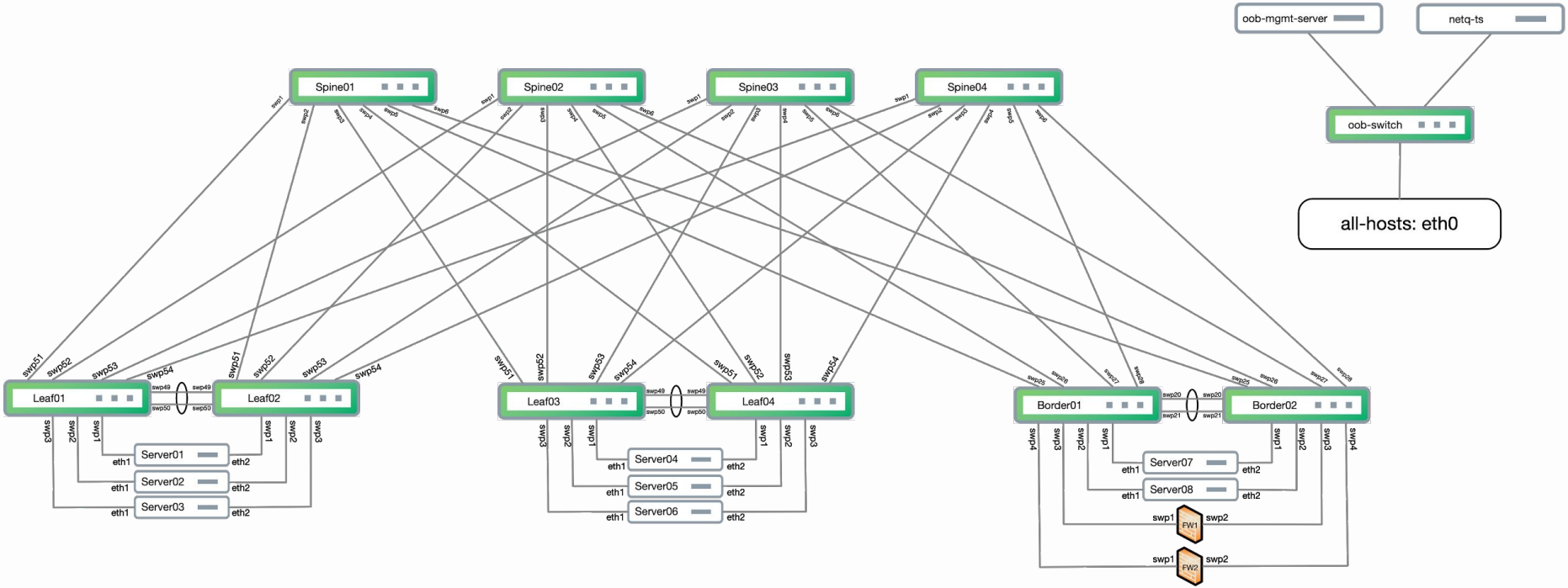
EVPN RIB

UNSUPPORTED

It is a two-layer table:

1. per-RD / VRF discrimination
2. normal IP-like routes

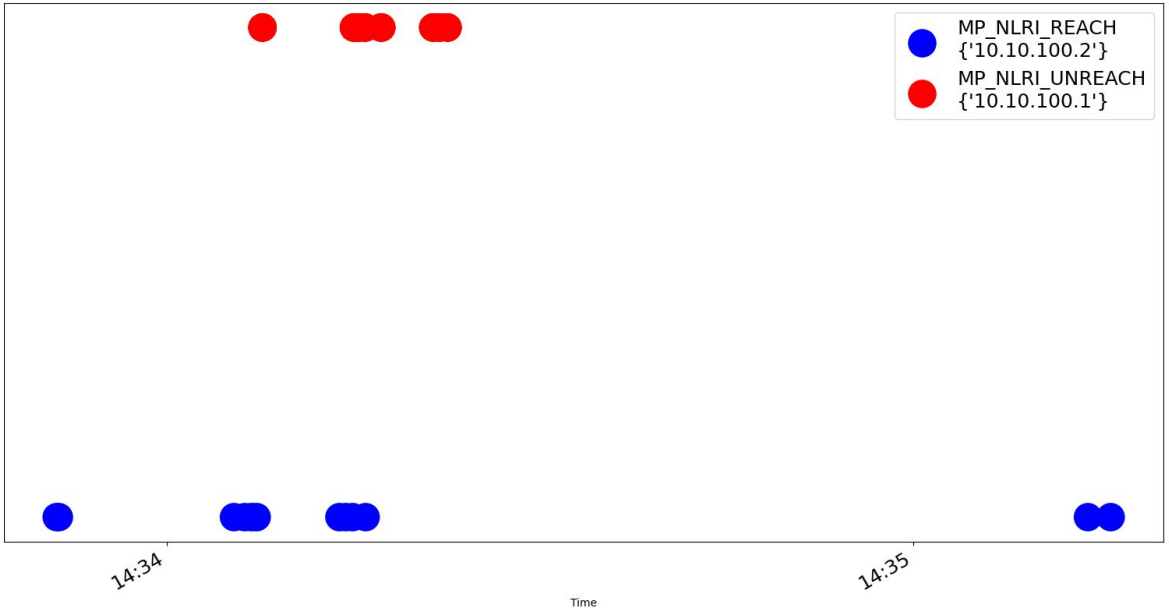
Proof of concept / The environment



Use cases / VM movements and convergence

- ➔ Detect events for a given MAC using time deltas:
  - using time delta mean, standard deviation, number of messages and user input we can detect which messages indicate a new event
- ➔ Allows to detect when and where to a VM was moved
- ➔ The difference between the time the first BMP message was received and the last one gives a measurement for the convergence time of the network

Use cases / VM movements and convergence



MAC

44:38:39:ff:00:19

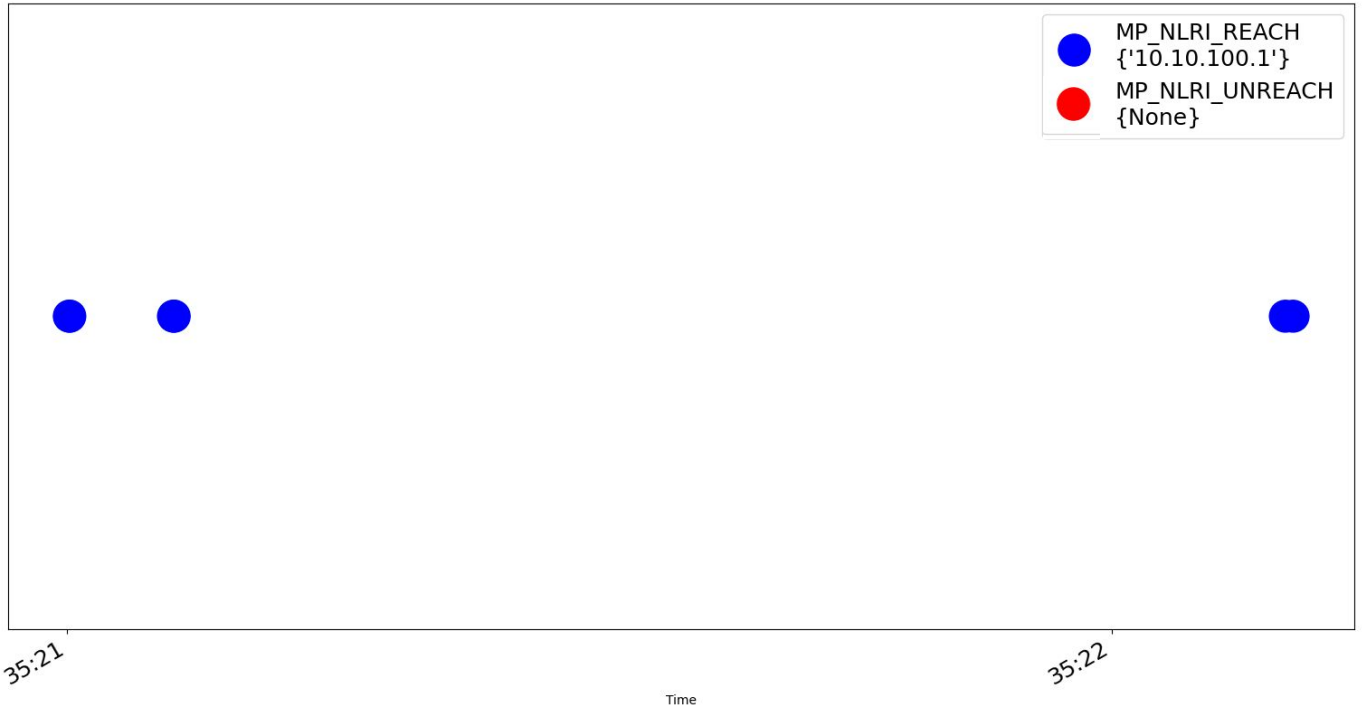
Convergence time mean

1.46s

Convergence time stdev

0.25s

Use cases / VM movements and convergence

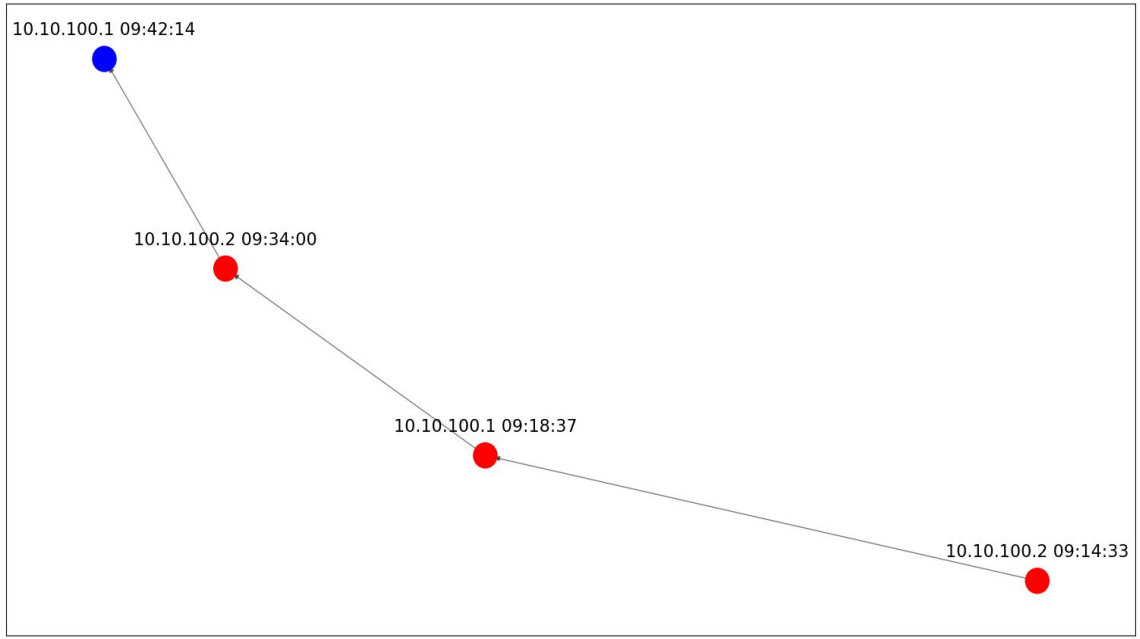


Use cases / MAC flapping

- ➔ EVPN type-2 messages are used to distribute MAC reachability information
- ➔ A given MAC address should be reachable from a single address, in the case of our simulation, the anycast address assigned to the two pairs of leaf switches
- ➔ If the same MAC address is advertised by more than one, this could be an indication of misconfiguration: this “*makes the network more vulnerable and wastes network resources*”. [3]

[3] https://juniper.net/documentation/en_US/junos/topics/task/configuration/configuring-mac-mobility-settings.html

Use cases / MAC flapping

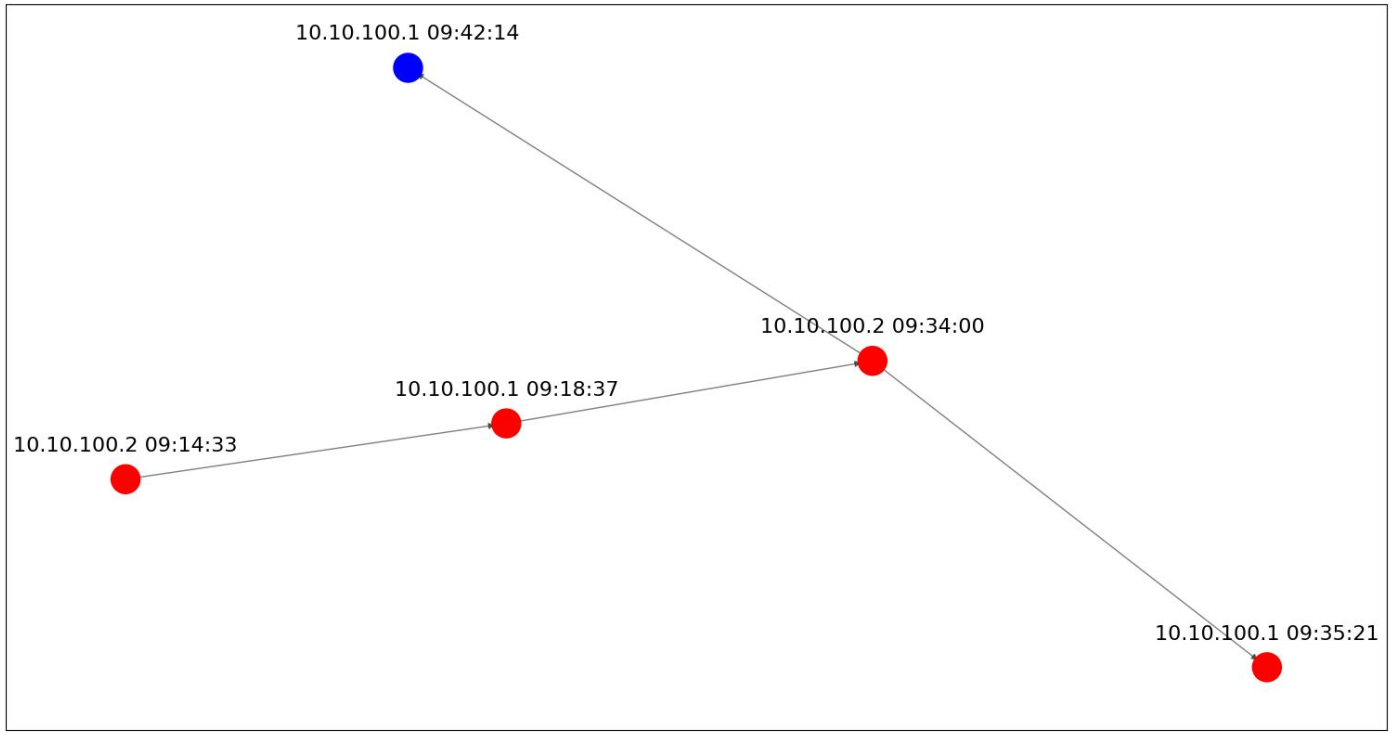


Red nodes have been invalidated

Blue nodes are currently active

Labels represent **`Next Hop`** network address and **`time`** of creation

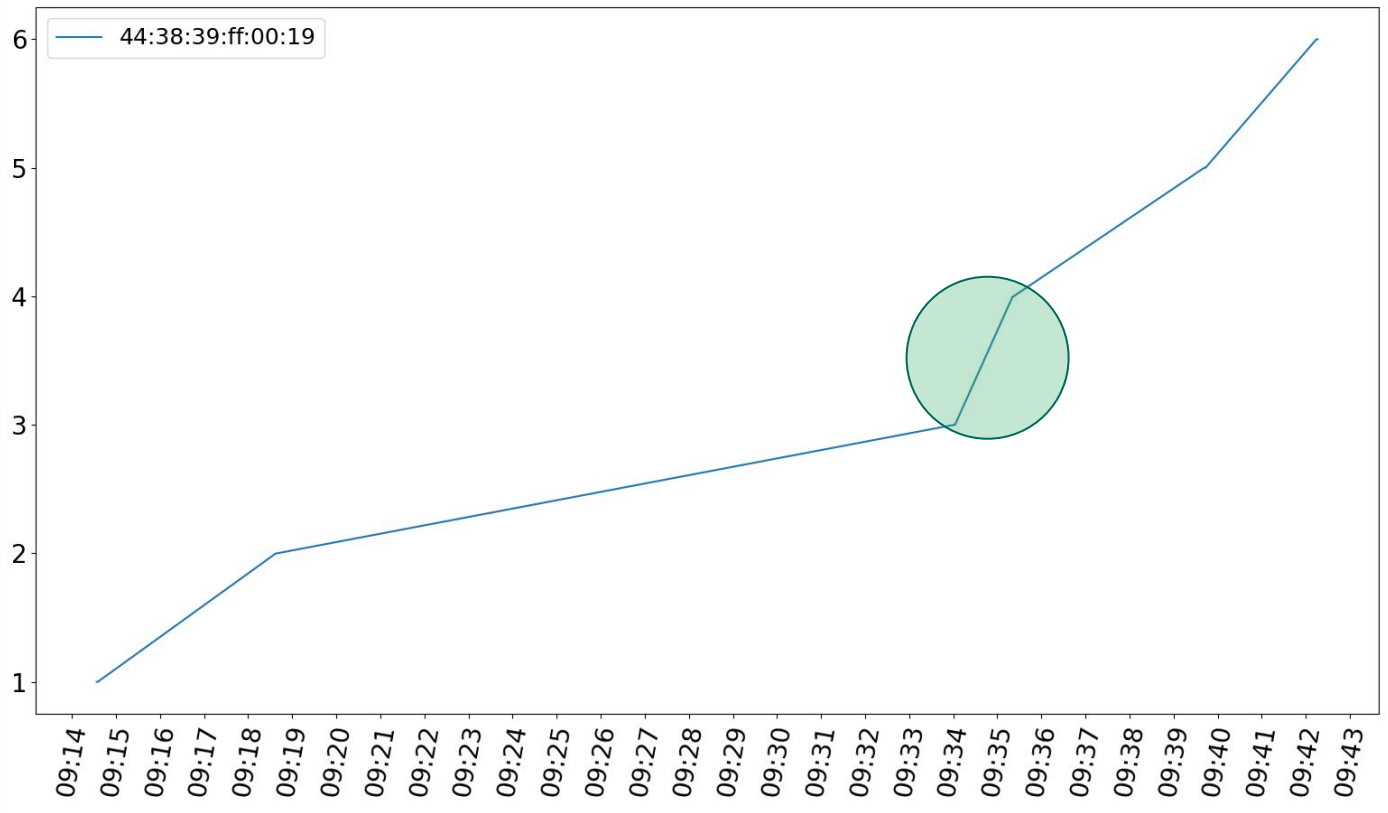
Use cases / MAC flapping



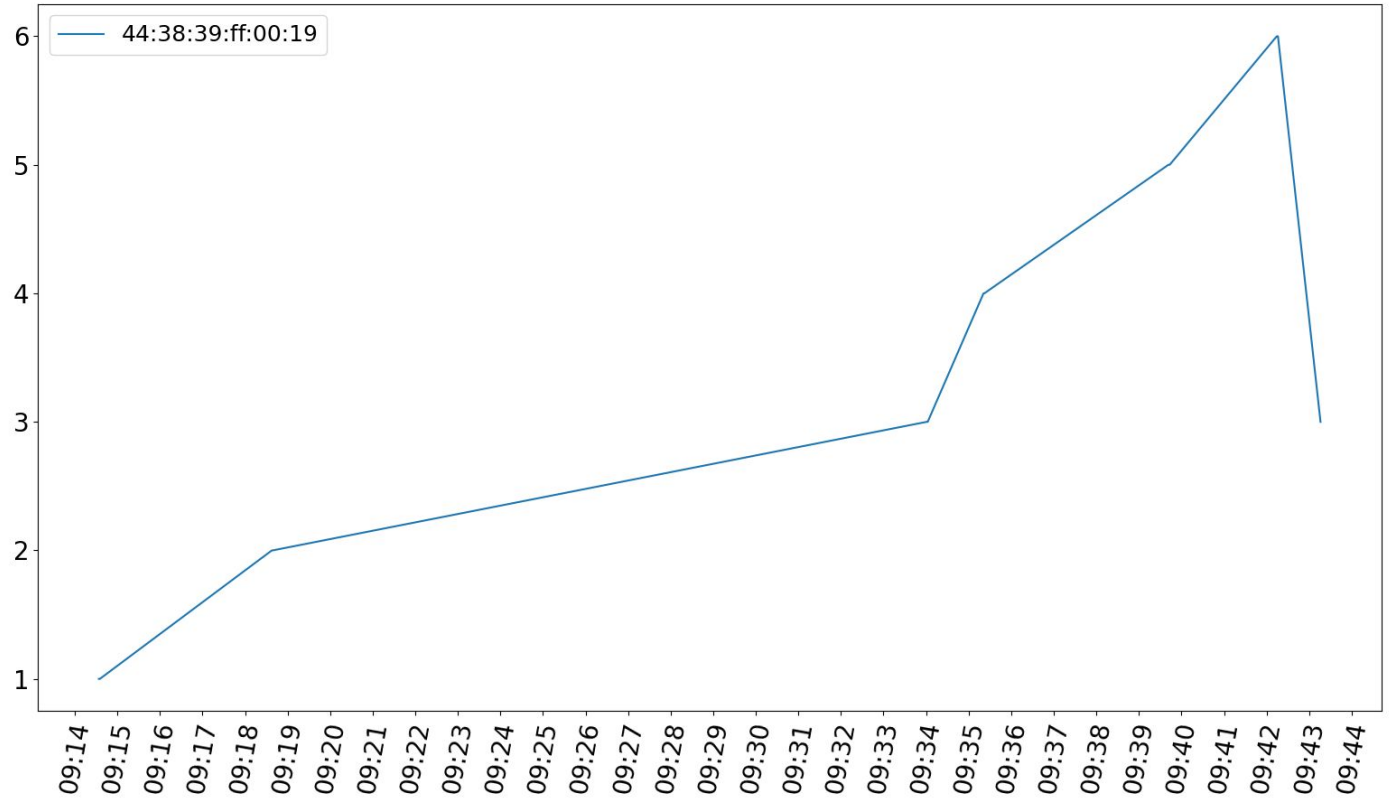
Use cases / MAC Mobility counter

- ➔ The MAC Mobility counter keeps track of how many times a MAC address has been moved across Ethernet segments
- ➔ Irregularities in a MAC Mobility counter for a given MAC can be indications of large network latencies or VM management misconfigurations
- ➔ MAC Mobility counter should not decrease (other than when it wraps around), nor increase unusually quickly

Use cases / MAC Mobility counter



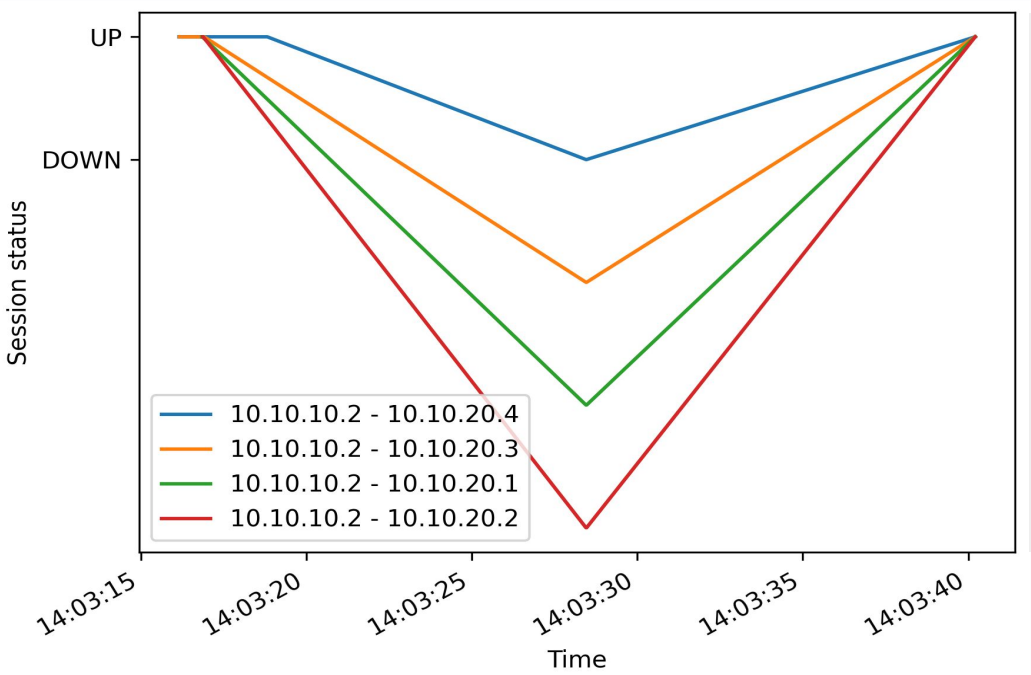
Use cases / MAC Mobility counter



Use cases / BGP Sessions

- ➔ A couple `(bgp_id1, bgp_id2)`, regardless of items order, defines a session
- ➔ BGP sessions are
  - established sending the `BGP OPEN` message: it carries both peers involved BGP IDs
  - terminated sending the `BGP NOTIFICATION CEASE` message: it carries only the BGP ID of the peer triggering the termination, thanks to the BMP header

Use cases / BGP Sessions



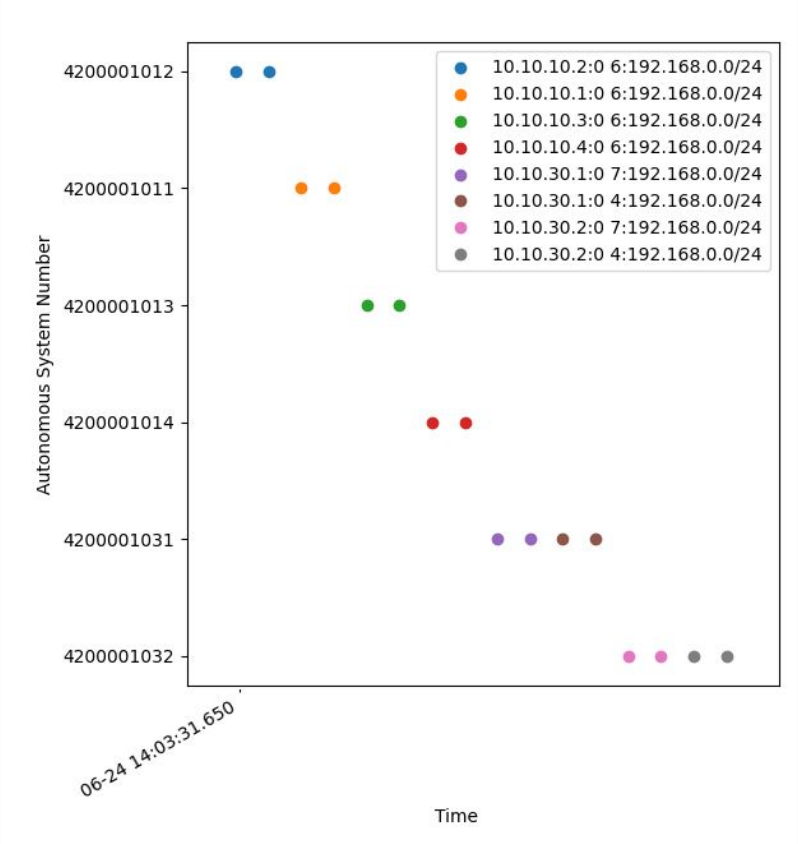
In the case presented, **`leaf02`** (BGP ID **`10.10.10.2`**) has gone down.

All its neighbours reported the **peer down** event to the BMP server.

Use cases / Prefixes authority

- ➔ A prefix, in EVPN, is exchanged as a type-5 (`IP Prefix Route`) route
- ➔ Carried along with the `BGP UPDATE NLRI`, it is the `AS_PATH` path attribute
  - regardless of the receiver of the message, such attribute can be leveraged to know which peer announced such prefix (*i.e.*, prefix authority)
- ➔ Tracking such announcements allows to infer whether a certain prefix has been moved in terms of authority

Use cases / Prefixes authority



BMP impact on network design

The only requisite of a BMP server is its reachability from the client BGP speaker:

- ➔ for convenience, the BMP connection would be done via a **management network**, so to isolate and manage monitoring on an isolated environment and network segment
- ➔ apart for this consideration, the addition of the BMP server in the topology has **no impact** at all, as it is logically separated by the effective BGP logical network

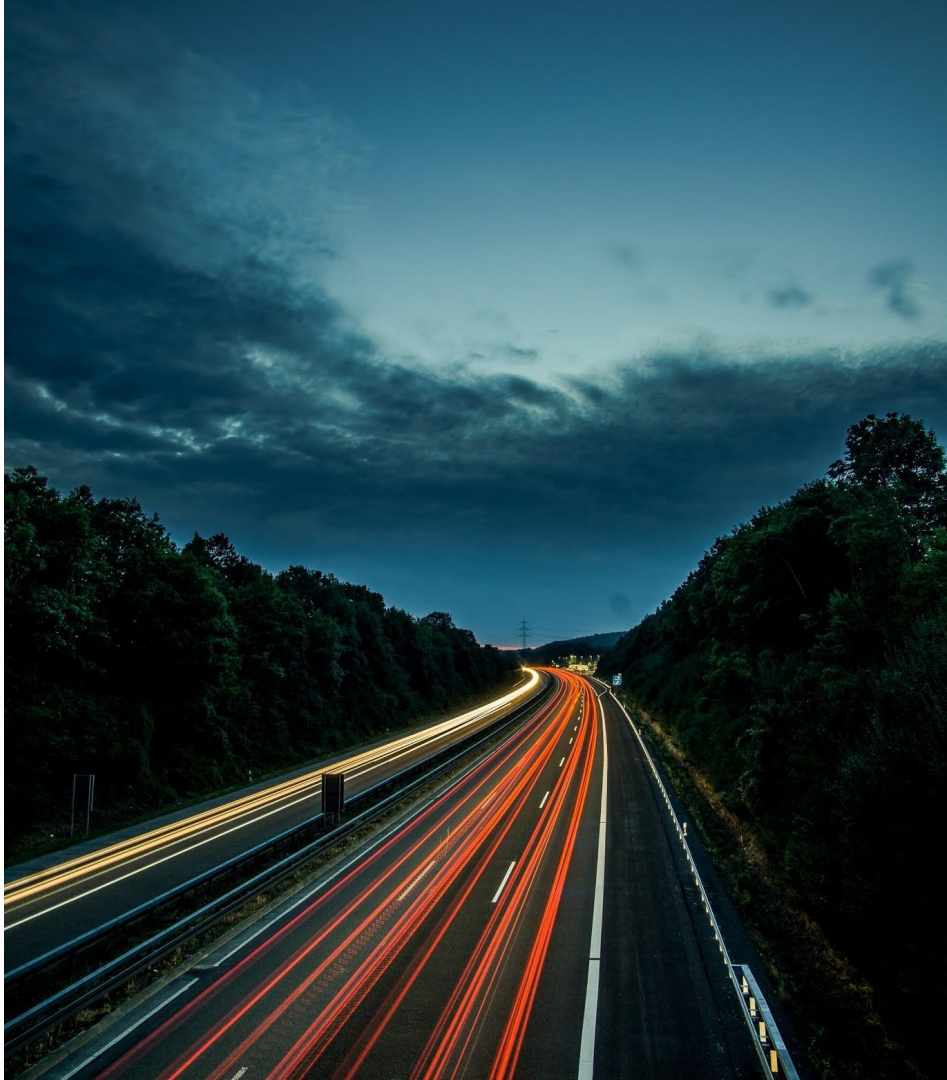
Conclusions

- ➔ BMP client limitations FRR-side
  - we overcame these by extending the existing implementation
- ➔ Lack of open BMP server solutions
  - this was addressed by developing our own ad-hoc BMP server, parser and analyzer
- ➔ Identified a specific set of use cases
  - all of them were successfully fulfilled in the test environment, by deploying our BMP server / client solutions

Conclusions / Further work

- ➔ Improve BMP-wise FRR implementation
  - relayed messages have wrong timestamps
  - monitor mode could be more stable overall
  - make BMP VRF-aware
- ➔ Improve the EVPN BMP listener
  - parsing support for more protocols / path attributes / extended communities could be added: this would also improve stability
- ➔ A small set of use cases was defined: more could be found and developed
- ➔ Only `pre-policy` BGP messages were observed: looking into `post-policy` elaboration could offer more insights into possible routers faults

Thank you.



Cumulus Networks
https://cumulusnetworks.com

Security and Network Engineering
https://os3.nl

University of Amsterdam
https://uva.nl