

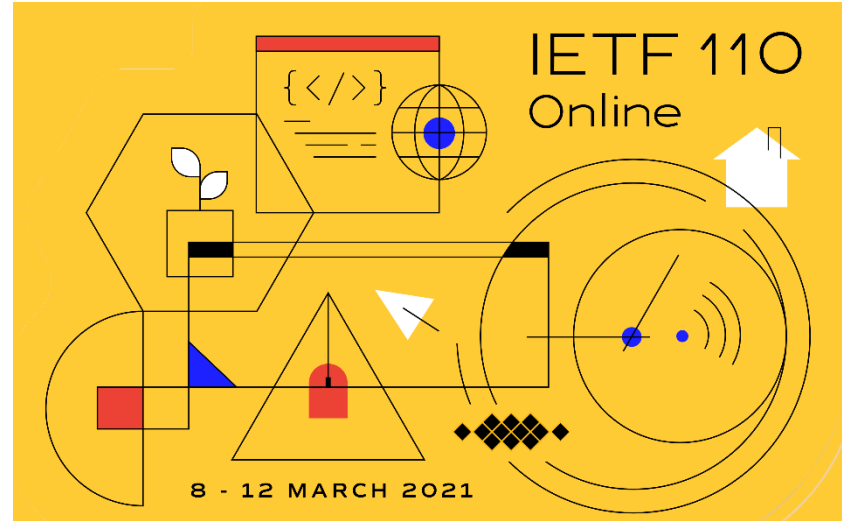
# BMP & YANG

GROW and NETCONF WG

IETF 110

March 1-5th, 2021

Virtual Hackathon



# BMP Hackathon - Plan

## Performance

- Measure CPU and memory consumption of BGP process when BMP Adj-RIB IN, OUT and Local-RIB with path-marking TLV is enabled and BMP session is flapping.
  - [draft-ietf-grow-bmp-local-rib](#) (BGP Local RIB)
  - [draft-grow-bmp-tlv](#) (TLV support for BMP Route Monitoring and Peer Down Messages)
  - [draft-cppy-grow-bmp-path-marking-tlv](#) (Path Marking TLV)
- Verify if with BMP route-monitoring mirrored BGP RIB state under BGP congestion is always accurate. Perform loss analysis if loss is present.
- Verify possible BGP route-propagation delay impact when BMP is enabled on a transit node. Perform delay analysis if delay is present.

# YANG Push Hackathon - Plan

## Functionality

- Finalize development of open-source UDP-based Transport for Configured Subscriptions data collection library and mockup publisher.
  - [draft-ietf-netconf-udp-notif](#) (UDP-based Transport for Configured Subscriptions)
  - [draft-ietf-netconf-distributed-notif](#) (Subscription to Distributed Notifications)
- Integrate udp-notif library into pmacct open-source network data-collection.

## Performance

- Test efficiency and throughput with various packet sizes on one core.

# Hackathon – Software

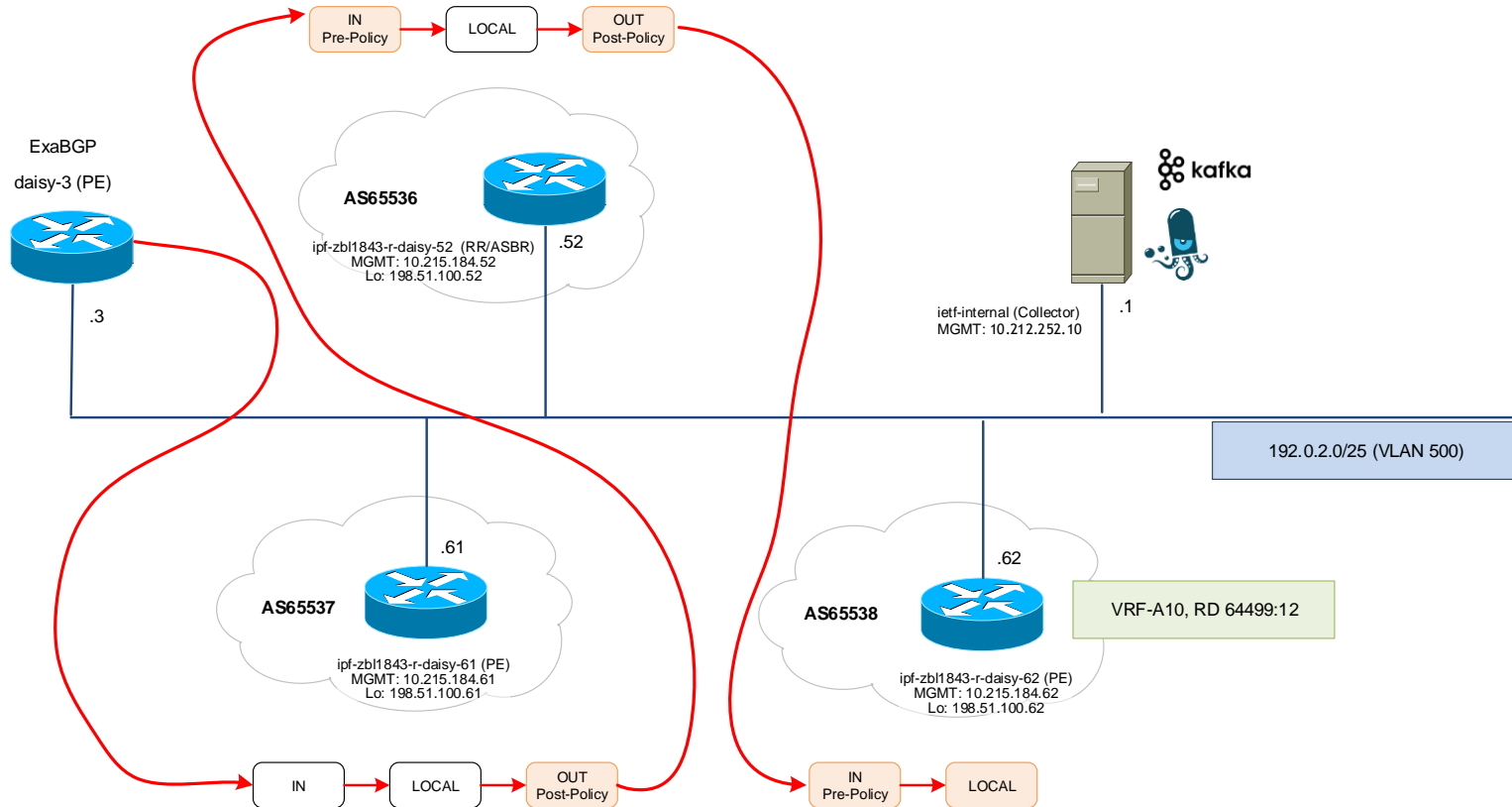
## Software

- [pmacct](#) nfacctd for IPFIX and BMP data collection
- [pmacct](#) udp-notif for YANG push data collection
- Apache [Kafka](#) as message broker
- Apache [Druid](#) as timeseries DB
- [Pivot](#) as user interface
- Wireshark [BMP dissector](#) for packet analysis
- [ExaBGP](#) for BGP VPNv4/6 route generation

## Tutorial

- <https://imply.io/post/add-bgp-analytics-to-your-imply-netflow-analysis>

# Hackathon - Network



# Swisscom – lab environment

## Achievements

- Test automation contains ExaBGP for sequenced BGP VPNv4 unicast route generation, BMP state initialization, BMP metric and YANG push cpu and memory process usage data collection.
- BMP route-monitoring prefix loss and delay can be automatically measured.
- CPU and memory usage now monitored on BGP process level.

## Next Steps

- Redo same tests with Cisco IOS XR and Juniper JunOS and compare results.
- Redo same tests with improved timestamping on Huawei VRP.

# Pmacct & INSA – nfacctd/udp-notif

## Achievements

- C Implementation of a collector for [draft-ietf-netconf-udp-notif-01](#)
- C Implementation of a producer API, part of the library
- Segmentation option supported
- Integrated as a library in [pmacct](#)

<https://github.com/pmacct/pmacct/>

```

{
  "node_id_str": "ipf-zbl1843-r-daisy-81",
  "subscription_id_str": "DAISY3",
  "sensor_path": "huawei-debug:debug/cpu-infos/cpu-info",
  "proto_path": "huawei_debug.Debug",
  "collection_id": "11480",
  "collection_start_time": "1614900107648",
  "msg_timestamp": "1614900107660",
  "collection_end_time": "1614900107660",
  "current_period": 10000,
  "except_desc": "OK",
  "product_name": "NE40E",
  "encoding": "Encoding_JSON",
  "data_str": {
    "row": [
      {
        "timestamp": "1614900107658",
        "content": {
          "debug": {
            "cpu-infos": {
              "cpu-info": [
                {
                  "position": "3",
                  "overload-threshold": 90,
                  "unoverload-threshold": 75,
                  "interval": 8,
                  "index": 16973825,
                  "system-cpu-usage": 12,
                  "monitor-number": 48,
                  "monitor-cycle": 10,
                  "overload-state-change-time": "0000-00-00 00:00:00",
                  "current-overload-state": "Unoverload"
                }
              ]
            }
          }
        }
      ]
    }
  },
  "delete": [],
  "generator": {
    "generator_id": "0",
    "generator_sn": "0",
    "generator_sync": false
  }
},
"software_version": "V800R013C00SPC006T"
}

```

we did it!

OMG I can't believe we're going to be parents



# Pmacct & INSA – nfacctd/udp-notif

## Test Setup

- i7-7700HQ, 2x8G@2400 MHz, x86\_64 Linux 5.4.0-66-generic
- Collector affinity set to one core
- Sample traffic sent from the other cores using Producer API
- Average performance on 10 runs with 500K messages sent (not much variance observed)

## Throughput

- 200B messages: 431Mbps
- 1500B MTU : 3,5Gbps
- 9000B MTU : 11,5Gbps

# Huawei - VRP

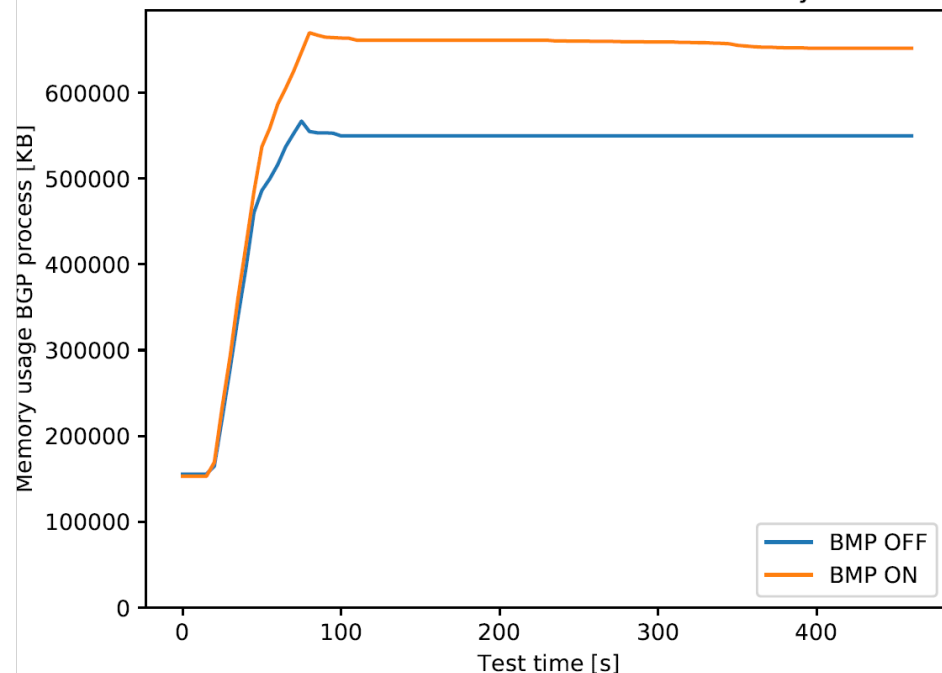
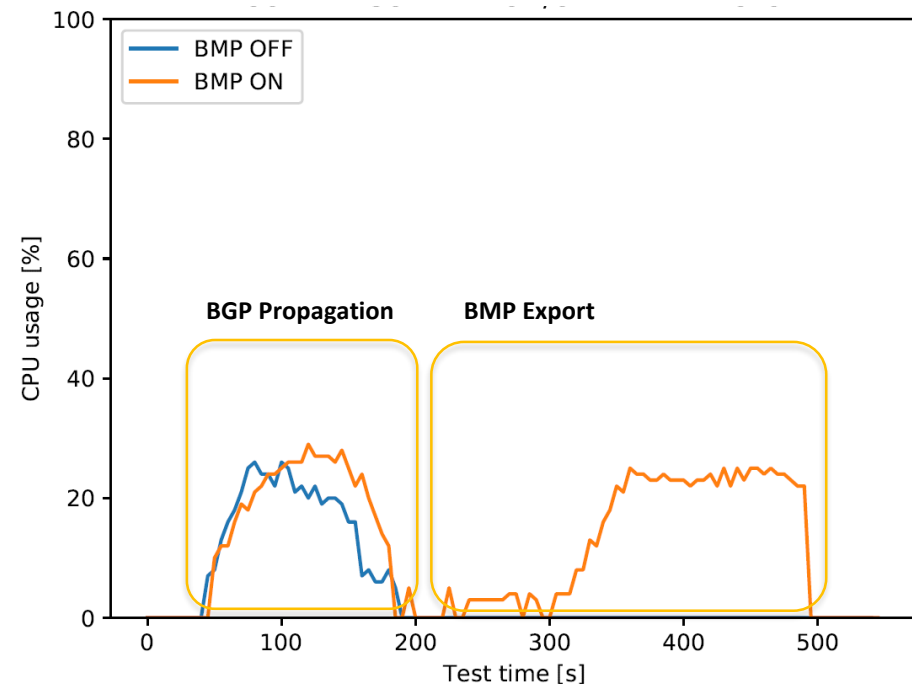
## Achievements

- BMP enabled on route-reflector and provider edge routers for Adj-RIB In pre-policy, Local RIB and Adj RIB Out post policy with path marking support.
- CPU increased **after** BGP converged when BMP is enabled. Slight overall increase of memory consumption observed.
- At the end of all the tests, BMP exported **RIB state** with route-monitoring always **matched** with RIB state on routers. **Impressed!**
- The BGP propagation delay, compare when BMP is enabled/disabled in transit, could not be measured accurate enough to draw final conclusions.

## Next Steps

- Improve BMP time stamping accuracy.

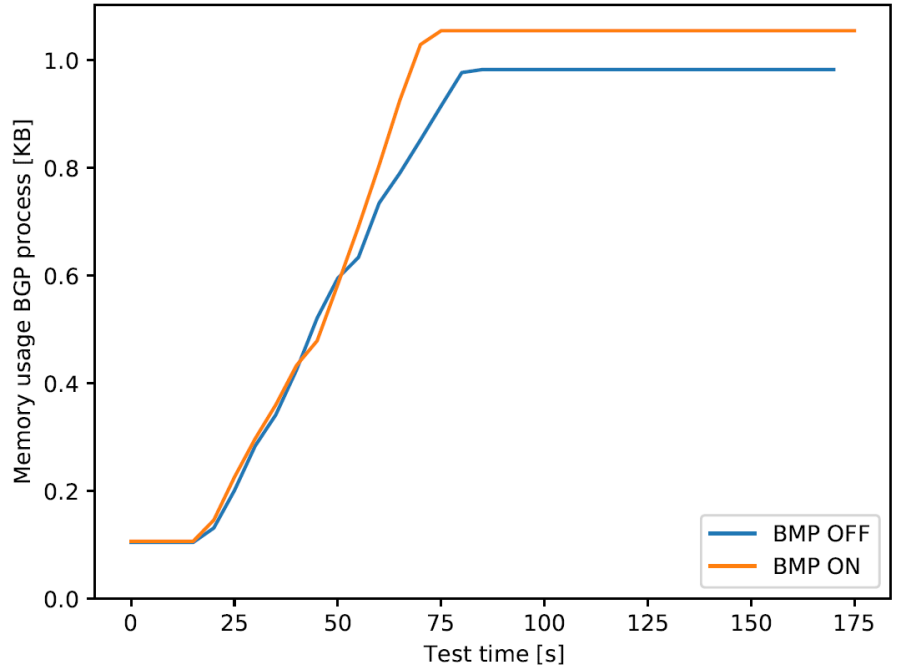
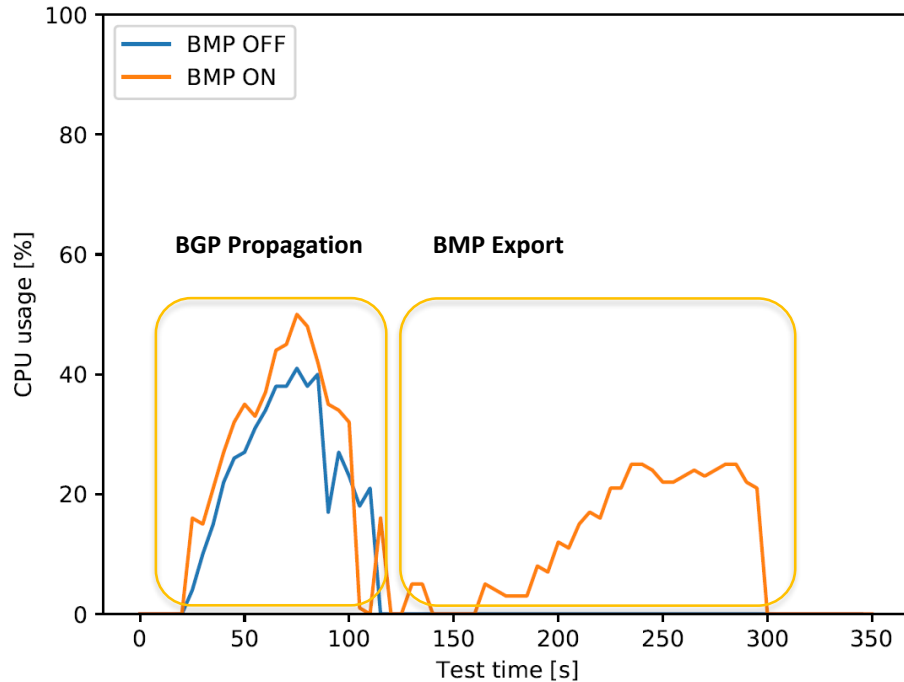
# BMP ON/OFF Test – Route Reflector



1'000'000 BGP VPNv4 unicast paths advertised as fast as possible to 10 peers.

BMP session on/off, enabled on 1 Adj-RIB In pre-policy and 1 Adj-RIB Out post-policy peer each.

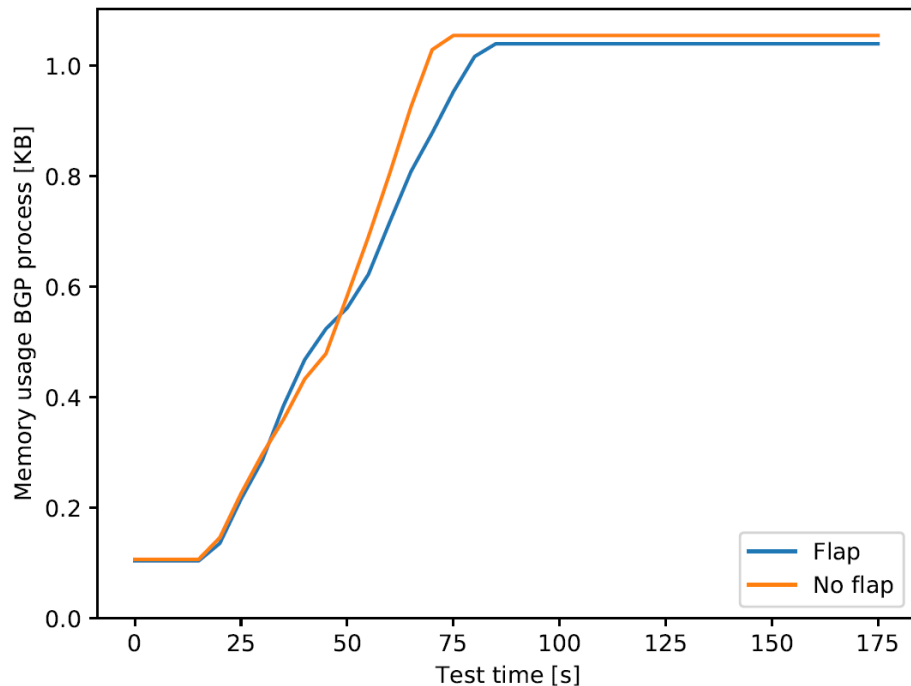
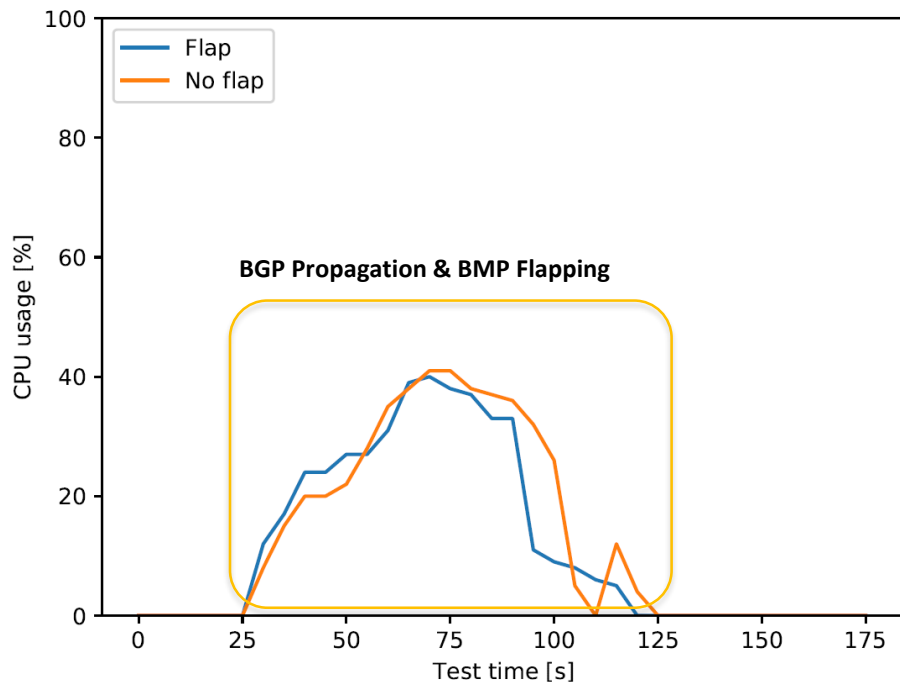
# BMP ON/OFF Test – Provider Edge



1'000'000 BGP VPNv4 unicast paths advertised as fast as possible to 10 peers.

BMP session on/off, enabled on 1 Adj-RIB In pre-policy and 1 Adj-RIB Out post-policy peer each.

# BMP Flapping Test – Route Reflector



1'000'000 BGP VPNv4 unicast paths advertised as fast as possible to 10 peers.

BMP session **flapping**, enabled on 1 Adj-RIB In pre-policy and 1 Adj-RIB Out post-policy peer each.

# What we learned

- Good

- With the 5<sup>th</sup> hackathon, we know the drill. Consistency more and more pays off.
- Good preparation, planning with test automation was gold.
- Slack and MS teams helped to stay connected.

- Bad

- Yet again, missing beers and cocktails after 😊

# Thanks to...

- Alex Huang Feng – INSA
- Pierre Francois – INSA
- Yunan Gu - Huawei
- Binyang Huang – Huawei
- Shuanglong Chen - Huawei
- Paolo Lucente – NTT
- Marco Tollini - Swisscom
- Matthias Arnold - Swisscom
- Thomas Graf - Swisscom

...[ImPLY](#) for providing us the big data,  
Huawei for the network environment and support,  
and Cisco for the test cases.