Streaming Telemetry in zero-footprint monitoring studies in WP6T3

Pavle Vuletic, UoB/AMRES
Marinos Dimolianis, NTUA

Telemetry and Big Data Workshop,

10th November 2020

Public

www.geant.org

Zero-footprint monitoring goals

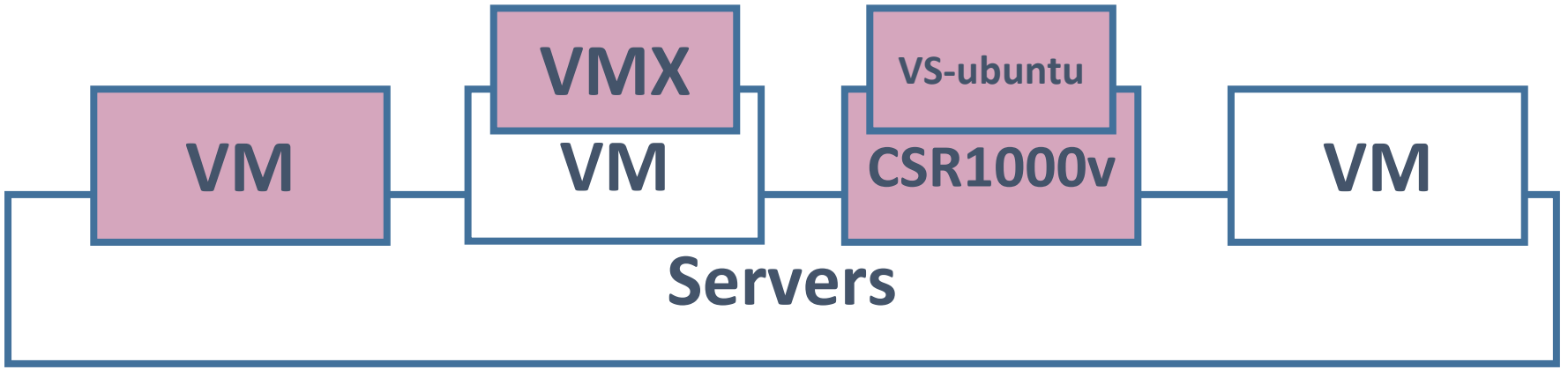
- (active) performance monitoring usually means adding additional devices (probes) into the PoPs - increased PoP complexity
- Performance - SLA parameter monitoring: delay, jitter, loss
- Can we monitor network service performance using standard based protocols and without the use of any additional equipment in PoPs? (zero footprint)
- Previous issues:
  - Proprietary protocols (Juniper RPM, Cisco SLA)
  - No standard protocol implementation
  - Monitoring probes became small, but even very small or virtual probe mean additional hardware in PoPs
- Other goals: Monitoring path end to end and per-segment, monitoring separate network services

Landscape has changed recently

- TWAMP implementation on Juniper and Cisco devices (Cisco - responder only)
- Virtual services on Cisco - can install Linux on spare CPU cycles and on that Linux e.g. perfSONAR,
- Cisco guestshell - run custom Linux applications, for automated control and management
- Are all these implementations interoperable?
- Can these TWAMP implementations be used for detailed network performance monitoring? (end-to-end and per segment)
- Export the data using streaming telemetry

Two-Way Active Measurement Protocol (TWAMP)

- Host to Host (twping – perfsonar/owamp)
- Host to Router (twping – TWAMP server)
- Router to Router (TWAMP server – client)
- Host to virtual service on a router (twping - twping)
- Virtual service on a router to router or host

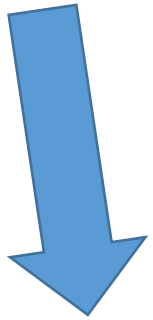


Although interoperable, the metric set is not the same

- **Juniper Routers (SNMP)**
  - Round Trip Time (RTT)
  - RTT Jitter
  - RTT Inter-arrival Jitter
  - Egress Jitter
  - Egress Inter-arrival Jitter
  - Ingress
  - Ingress Jitter
  - Ingress Inter-arrival Jitter

Min, Max, Average, StdDev

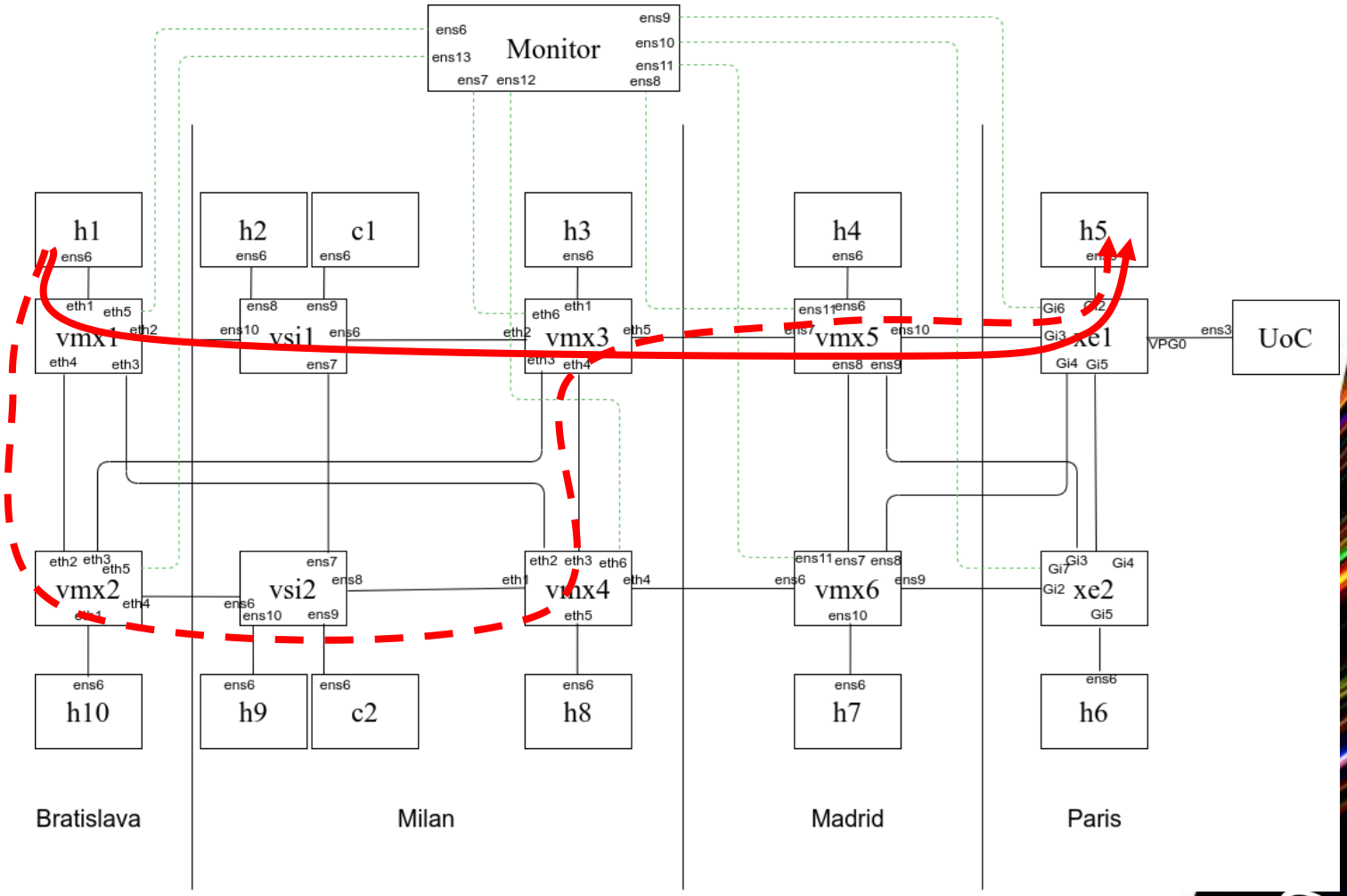
- **Linux Hosts**
  - Round Trip Time (RTT)
  - Send Time
  - Reflect Time
  - Two Way Jitter
  - Send Jitter
  - Reflect Jitter



Min, Max, Median

Experimental evaluation

- Path through the Linux, Juniper and Cisco devices
- Monitor per-segment and end-to-end service performance
- Added latency, jitter
- Test dynamic path changes



How it looked like

15:00: Additional delays are added using the tc tool on selected interfaces of the VS1 and VS2 devices:

- VS1 ens10: 20ms, ens6: 40ms
- VS2 ens6: 40ms, ens10: 60ms

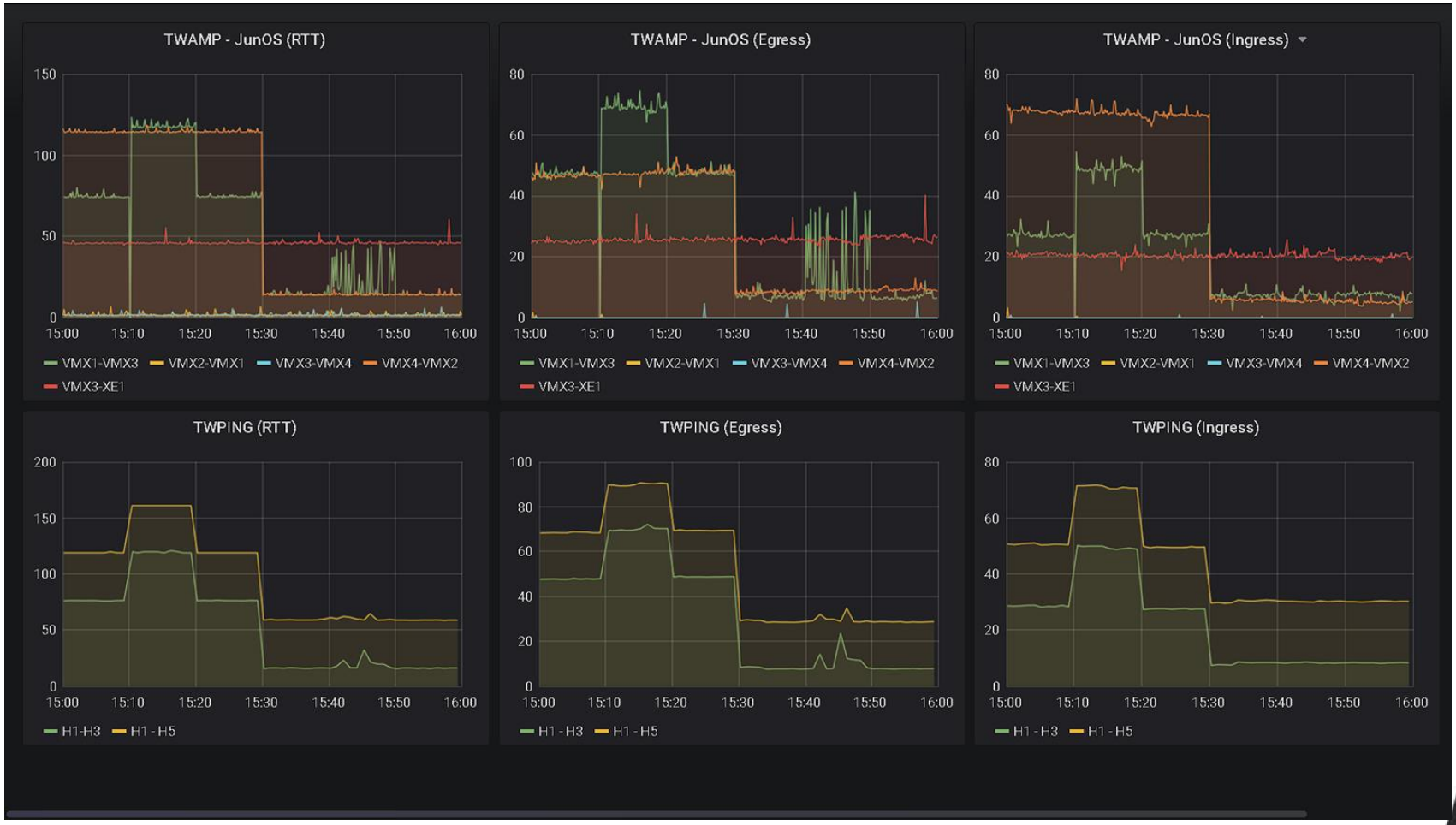
15:10: Traffic is redirected via vMX1-vMX2-VS2-vMX4-vMX3.

15:20: The original path is restored and network traffic is rerouted via the path: vMX1-VS1-vMX3.

15:30: Delays are returned to their original values.

15:40: The Jitter is added to vMX1->vMX3.

15:50: The Jitter is removed from vMX1-vMX3.



Why streaming telemetry?

- Supported on the latest releases of network OSes, although still different transport methods and data models:
  - J: UDP, gRPC, C: NETCONF, gRPC, gNMI
  - J: Juniper, OpenConfig, C: YANG
- Real time monitoring - you can retrieve data even within ms
- More reliable, secure than SNMP
- Expected to have smaller processing requirements in comparison to SNMP
- Use streaming telemetry to subscribe on data that the vendor supports or in programmable data planes stream your own data.

Streaming Telemetry export

- Dial-Out
  - Data subscription is configured in the router (periodic or on-event)
  - Simplicity
- Dial-In
  - Data subscription is defined in the collector
  - Scalability

Cisco Dial-Out Configuration

```
telemetry ietf subscription 98
 encoding encode-kvgpb
 filter xpath /ip-sla-ios-xe-oper:ip-sla-stats
 source-address 172.16.0.82
 stream yang-push
 update-policy periodic 5000
 receiver ip address 172.16.0.252 57000 protocol
grpc-tcp
```

Juniper Dial-In Configuration

```
system {
    services {
        ssh;
        telnet;
        extension-service {
            request-response {
                grpc {
                    ssl {
                        port 32767;
                        local-certificate vMX1;}
                    }
                }
            }
        }
    }
```

Streaming Telemetry Data collection

Telegraf - Configuration

Cisco Dial-Out

```
[[inputs.cisco_telemetry_mdt]]
 transport = "grpc"
 # Address and port to host telemetry listener
 service_address = ":57000"
 [inputs.cisco_telemetry_mdt.aliases]
  ifstats = "ietf-interfaces:interfaces-
state/interface/statistics"
```

Juniper Dial-In

```
 [[inputs.jti_openconfig_telemetry]]
  servers =
["vMX1:32767","vMX2:32767","vMX3:32767","vMX4:32767"]
  username = "XXX"
  password = "XXX"
  client_id = "telegraf"
  sensors = [
"/interfaces/",
"collection /components/ /lldp",
"twampmeasurements /junos/twamp/client/probe-test-results/" ]
```

Conclusions

- All the twamp session combinations (H->C, H->J, J->C, C-J, ...) gave very similar results and showed reliability over long periods of time
- The network was not heavily loaded
- Monitoring network services using standard based protocols and no additional hardware is possible.
- Streaming the results from both router platforms without any issues
- Setting up some of the elements (virtual service, finding xpaths of the variables for streaming telemetry) not always trivial and not perfectly documented

Soon...

- Zero-footprint Monitoring cookbook (look at the GEANT's announcements)
  - TWAMP interoperability
  - virtual services
  - streaming telemetry
- if interested, send me an email for the final draft.

dd-mm-yyyy

Zero-footprint Monitoring

Milestone <MX.X - Doc Property: Subject>

Grant Agreement No.:	856726
Work Package	<Insert EC work package number (WP1)>
Task Item:	<Insert task number – e.g. Task 2>
Dissemination Level:	<PU (Public), CO (Confidential), CI (Classified)>
Document ID:	<Doc Property: Keywords>
Authors:	Pavle Vuletić (University of Belgrade), Marinos Dimolianis (NTUA/GRNET), Victor Olifer (JISC)

© GÉANT Association on behalf of the GN4-3 project.
The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).

Thank you

Any questions?

www.geant.org

© GÉANT Association on behalf of the GN4 Phase 3 project (GN4-3).
The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).