In-band Network Telemetry

Mauro Campanella (GARR), Tomas Martinek (CESNET), Damian Parniewicz (PSNC), Federico Pederzolli (FBK), Damu Ding (FBK)

GÉANT INFOSHARE
02 March 2021

www.geant.org

GEANT Infoshares are intended to a create space to engage, improve knowledge sharing and discussion about services and strategic topics, and to build a human network across the Research and Education community.

Co-organised by WP6 and GÉANT Partner Relations Team, within the Community Programme

Public Infoshares are on Wednesdays and are recorded
Other Infoshares will be ‘invitation only’ events on other weekdays

Go to the main Infoshares Wiki page to suggest future topics

Recordings are available in the e-Academy, GLAD website and on Wiki pages after the event

Questions: partner-relations@geant.org

* During the infoshare, GÉANT may do audio or video recordings, that can include collective or individual images, and collect information provided by others (especially during training sessions), to evaluate your performance as a speaker or as a trainer. We could also make and store a recording of your voice in certain instances. GÉANT may use this information, including your personal data and the recorded sessions, in online communication channels managed by GÉANT such as YouTube, GÉANTtv, GLAD website, eAcademy and Wiki Pages.

Agenda

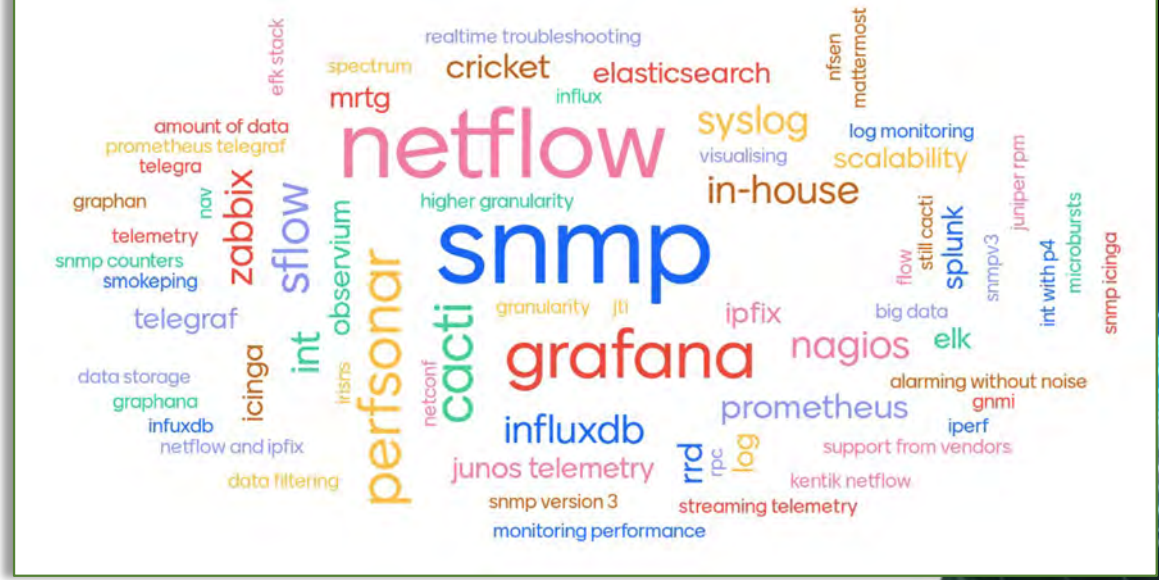
Introduction to INT	Mauro Campanella	GARR
Data Plane INT P4 implementation	Damian Parniewicz	PSNC
Q&A		
"End user" use case testbed	Federico Pederzolli	FBK
Packet behaviour - INT view	Tomàs Martinek	CESNET
Q&A		
Handling INT "big data	Damian Parniewicz	PSNC
Next steps and summary	Mauro Campanella	GARR
Mentimeter, Q&A		

Introduction to INT

Why work on INT (& telemetry & data plane programming)?

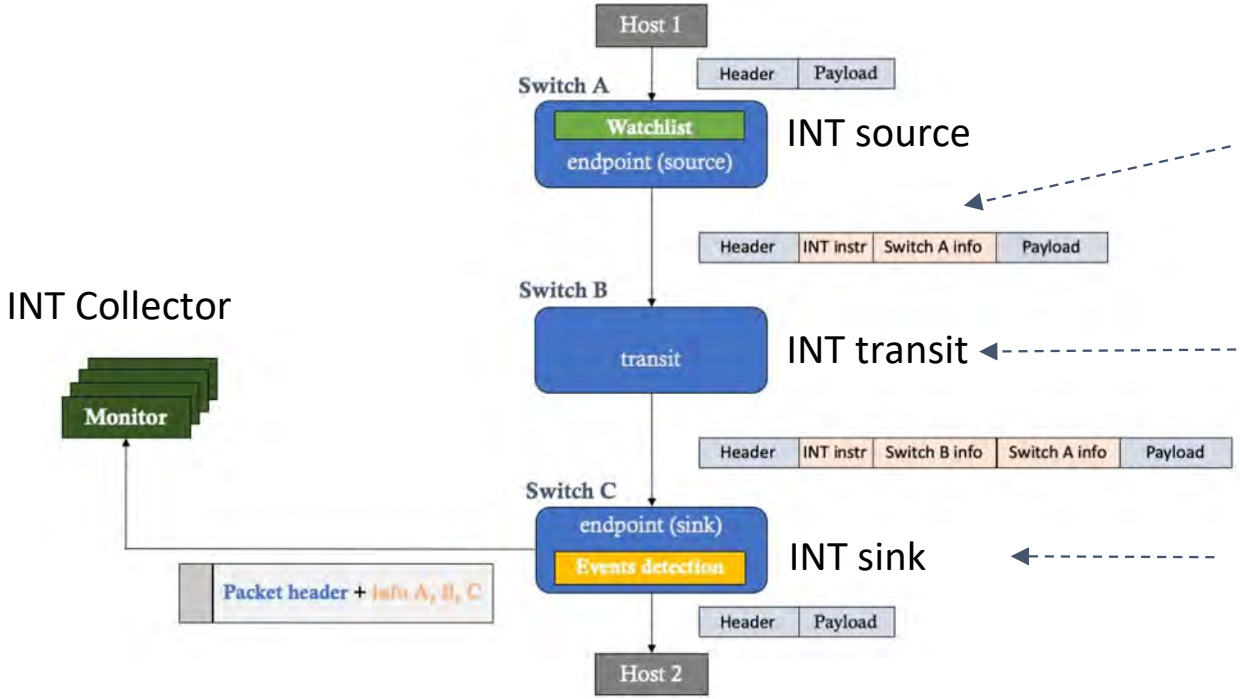
- Telemetry is the next step to provide data to monitoring and control plane, starting with streaming telemetry
- INT adds to telemetry monitoring granularity (choice of flows / packet / protocols/…) and **programmability in the data plane,** realtime
- Gather experience with network "**Big Data**" handling
- Evaluate its usefulness to better monitor and control our networks (creating **new knowledge**)

From participants to the Telemetry and Big Data Workshop: **monitoring protocols most used** today



In-Band Network Telemetry (INT) summary

INT was specified by the P4 language community to provide very detailed information on network behaviour



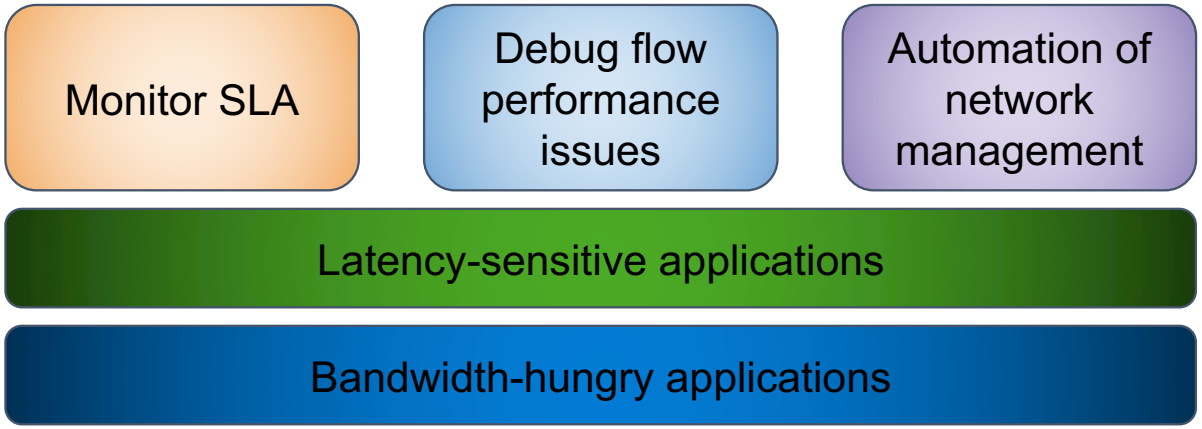
INT-enabled **source** node add a small **INT header** to **every chosen packet** containing Switch IDs, Interfaces IDs, Timestamps, Link and queue utilization e.g.

INT transit nodes add specific local

The last **INT sink** node exports INT data to the collector

What INT can tell about your packets and be used for?

- **Packet route** doesn't follow the operator's policy
- Notification when a specific flow **path changes**
- If some packets in the flow are affected by **latency spikes**
- When and where packets **microbursts** happen
- Is there **congestion** and where packets are **dropped**
- Which flows cause congestion

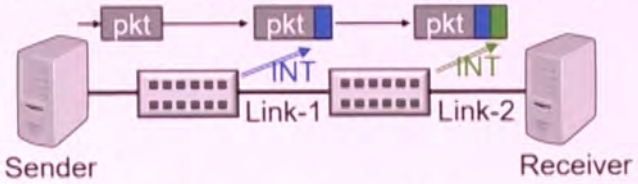


Example: INT-based High Precision Congestion Control

HPCC: use INT as precise feedback

Alibaba Cloud | Worldwide Cloud Services Partner

- In-band network telemetry (INT) provides many details per packet



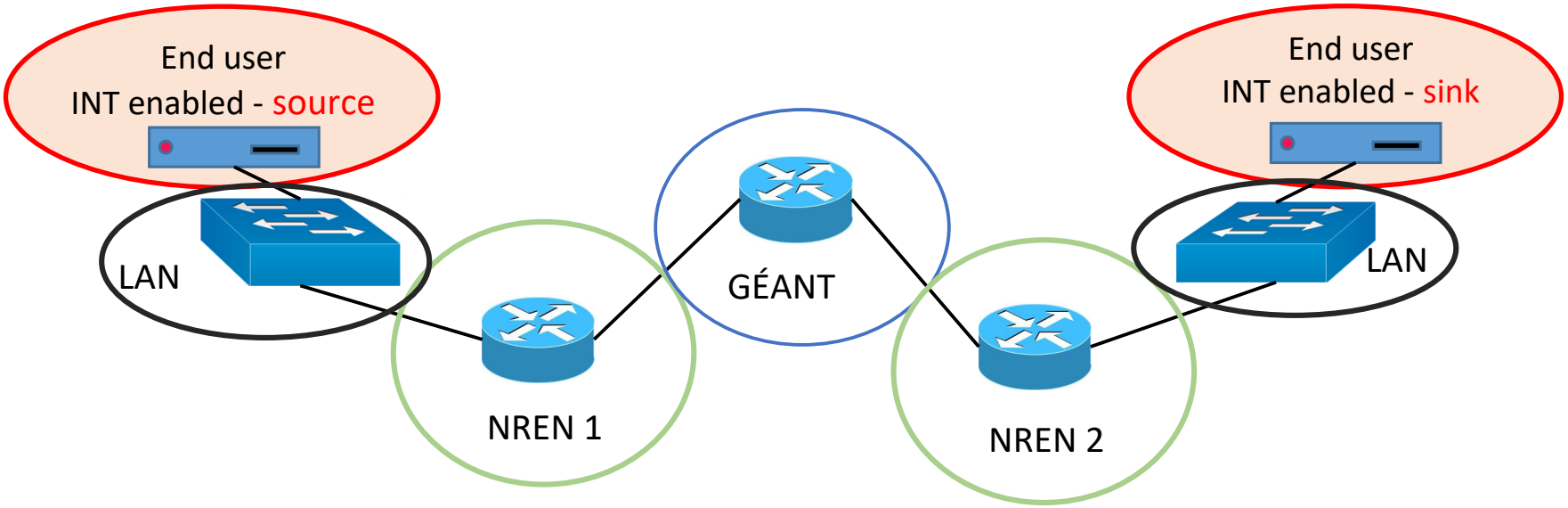
- Broadcom&Barefoot have INT in recent products.
- Alibaba has >10,000 switches support INT in production.
- Widely used for diagnosis and monitoring in production

HPCC solves the 3 problems

➢ Using INT as the precise feedback

- Fast convergence
  - ➢ Sender knows the precise rate to adjust to, on every ACK
- Near-zero queue
  - ➢ Feedback does not rely on queue
- Few parameters
  - ➢ Precise feedback, so no need for heuristics which requires many parameters

End user uses INT on its traffic to measures its IPDV, Loss,…



Edgecore Wedge100BF-32X Arista 7170-32c

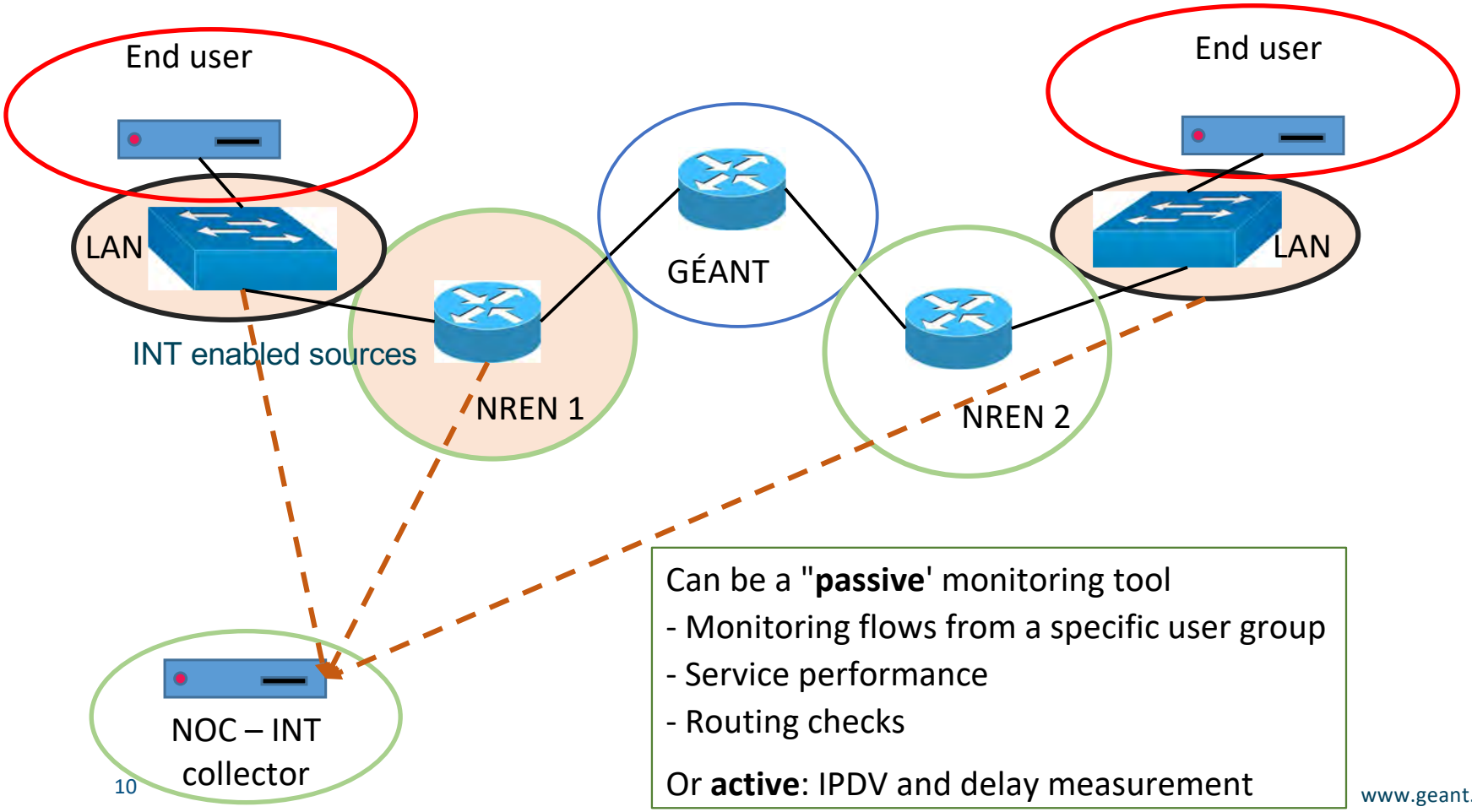
Tofino (Barefoot/INTEL)

BMv2 virtual
P4 software switch

P4 on DPDK
implementation

FPGA

NREN uses INT to debug network behaviour



Can be a "**passive**' monitoring tool
- Monitoring flows from a specific user group
- Service performance
- Routing checks

Or **active**: IPDV and delay measurement

The steps to set-up INT from scratch

Choose INT data and Header position	INT headers contain source and destination timestamp, sequence number, placed between UDP and payload
Coding using P4	HW programming, limited cycles and memory, lack of complex arithmetic operations, lower flexibility in the use of registers
Collector for INT data	Tuned InfluxDB (up to 260K INT data/s multiple threads)
Visualization	Grafana, Plotly , added IPDV computation in data plane
Analyze	"Knowledge" under development now

Time synchronization to be tuned to sufficient precision (few microseconds, ns), need also node internal clock stability

Data Plane INT P4 implementation

INT implemented using the P4 language

High-level, open programming language used to describe how packets should be processed within the network node

P4 was created in 2013 by researchers from researchers Princeton and Stanford University to overcome the limitation of OpenFlow

P4 is supported by:

- Ethernet switches build on **Intel/Barefoot Tofino** chipsets (Arista, Cisco, EdgeCore, Accton, …)
- **SmartNIC/FPGA cards** (Agilio, Xilinx, Intel, Pensando, Netcope, ...)

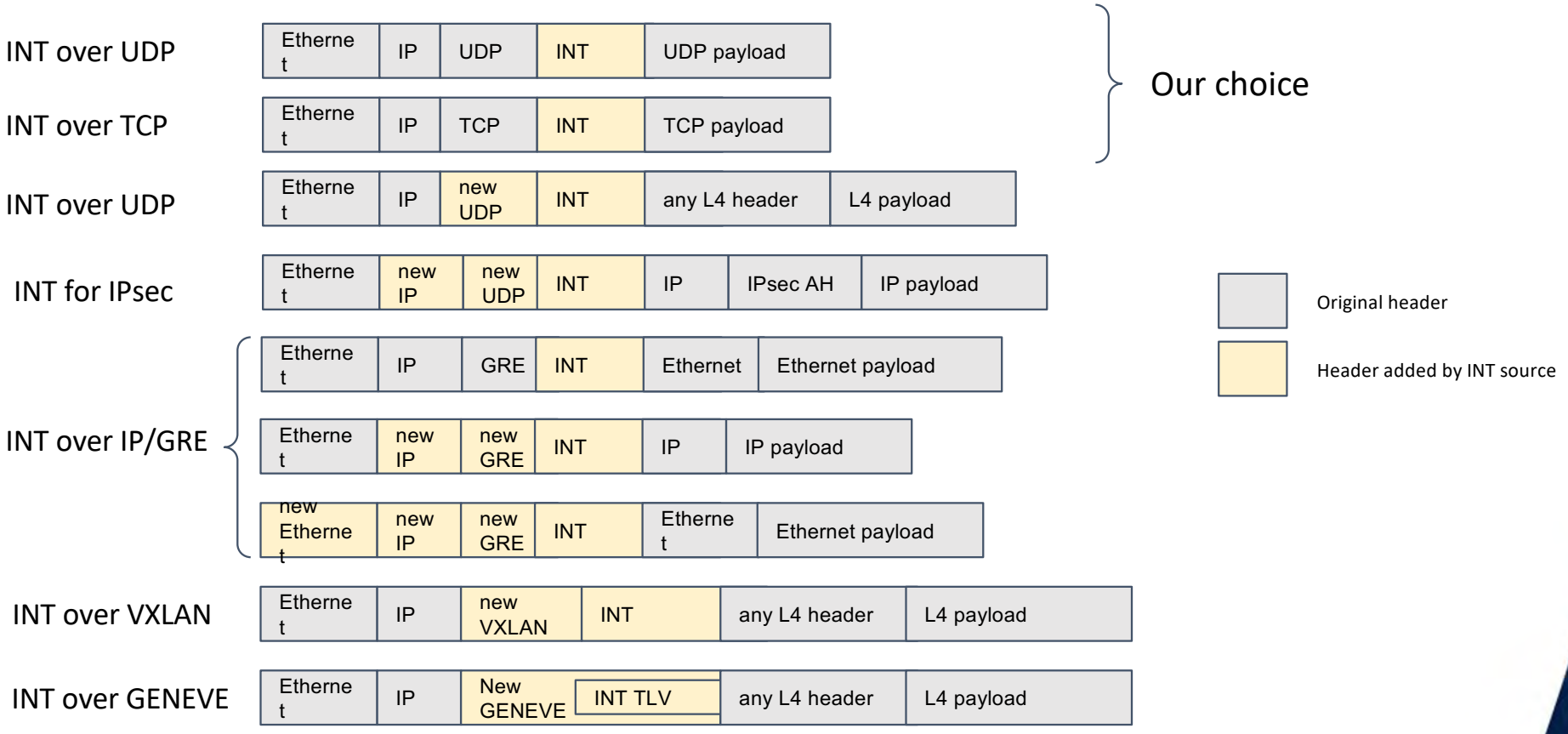
```
#include <v1model.p4>
header ethernet_t { bit<48> dst; bit<48> src; bit<16> etherType; }
struct headers { ethernet_t ethernet; }

parser MyParser(packet_in packet, out headers hdr) {
        state start { transition parse_ethernet; }
        state parse_ethernet {
                packet.extract(hdr.ethernet);
                transition accept;
}
control ForwardEgress(inout headers hdr, inout standard_metadata_t meta) {
        table send_frame {
                key = { hdr.ethernet.src: exact; meta.ingress_port: exact; }
                actions = { rewrite_smac; NoAction; }
                size = 256;
        }
        action rewrite_smac(bit<48> smac, bit<8> egress_port) {
                hdr.ethernet.src = smac; meta.egress_port = egress_port; }
        apply { send_frame.apply(); }
}
```

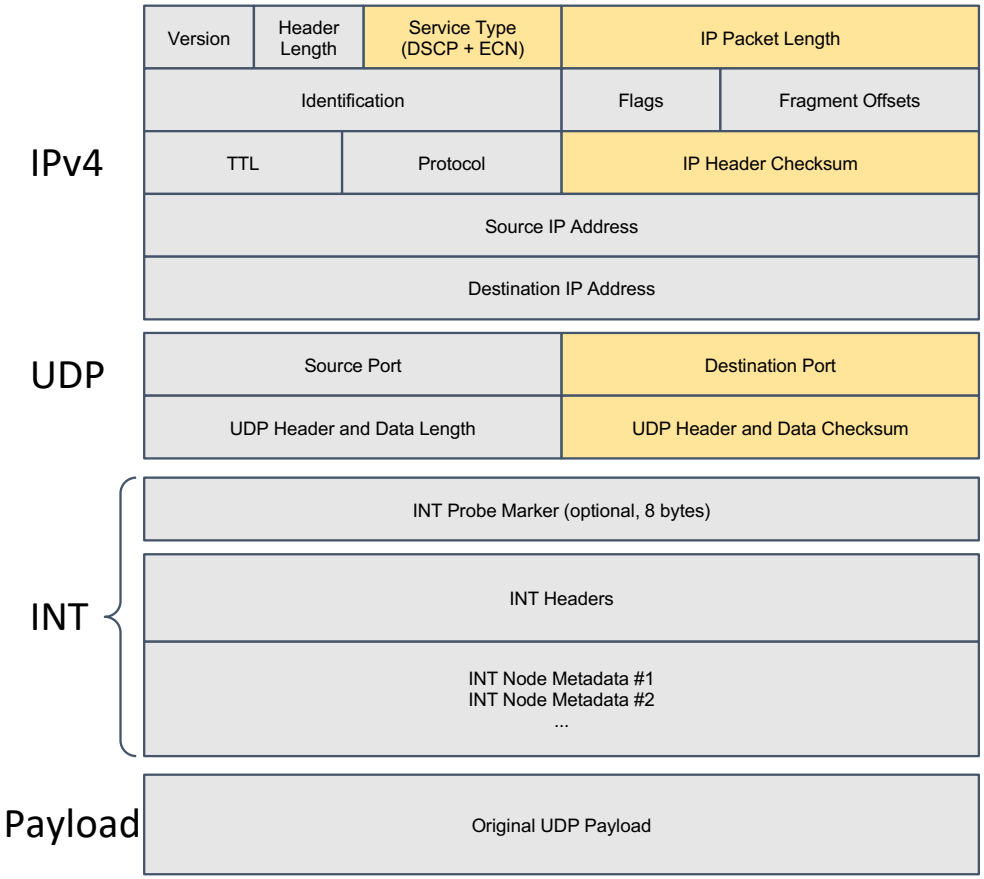
Implementation choices in our tests

- INT used to monitor **one** or a **set of traffic flows**
- **All packets** in the monitored flow transport INT headers
- Only INT sink generates **INT reports** (Hop-by-Hop INT)
- Use minimalistic L1/L2 **frame forwarding**
  - Another GÉANT activity (**RARE** - Router for Academia, Research and Education) is implementing a full-featured P4 router
- **Common P4 code** (v16) for all INT platforms (P4 v14 for FPGA)

Choosing INT headers location in the packet



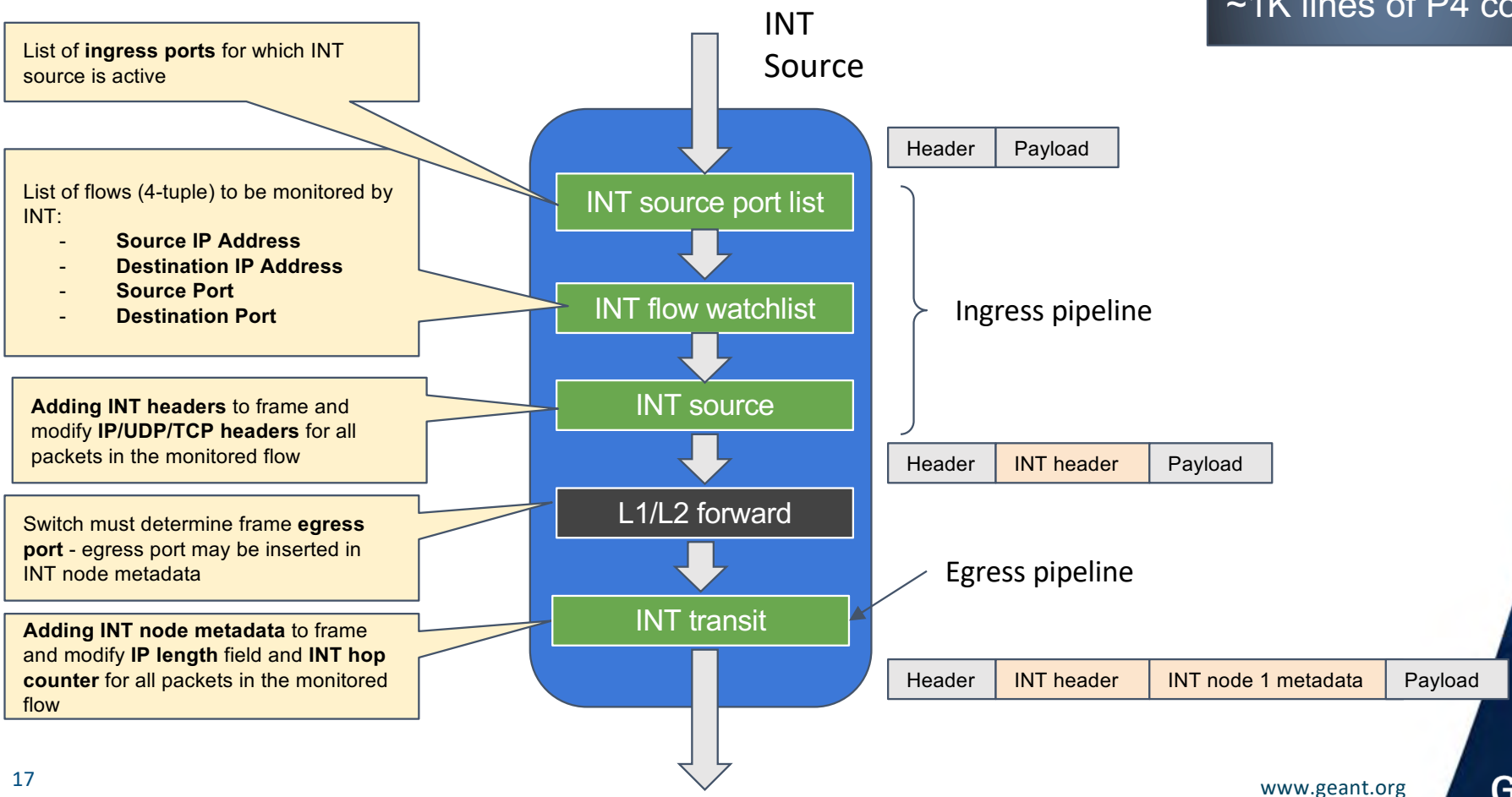
INT over UDP/TCP



- Presence of INT header indicated by:
  - UDP/TCP Destination Port number
  - IP DSCP number
  - Probe maker fields
- Original Destination Port or DSCP values are copied to INT headers
- A common INT identification method must be used within a network domain

INT source implementation

~1K lines of P4 code



How much does INT cost in bytes?

Our deployment

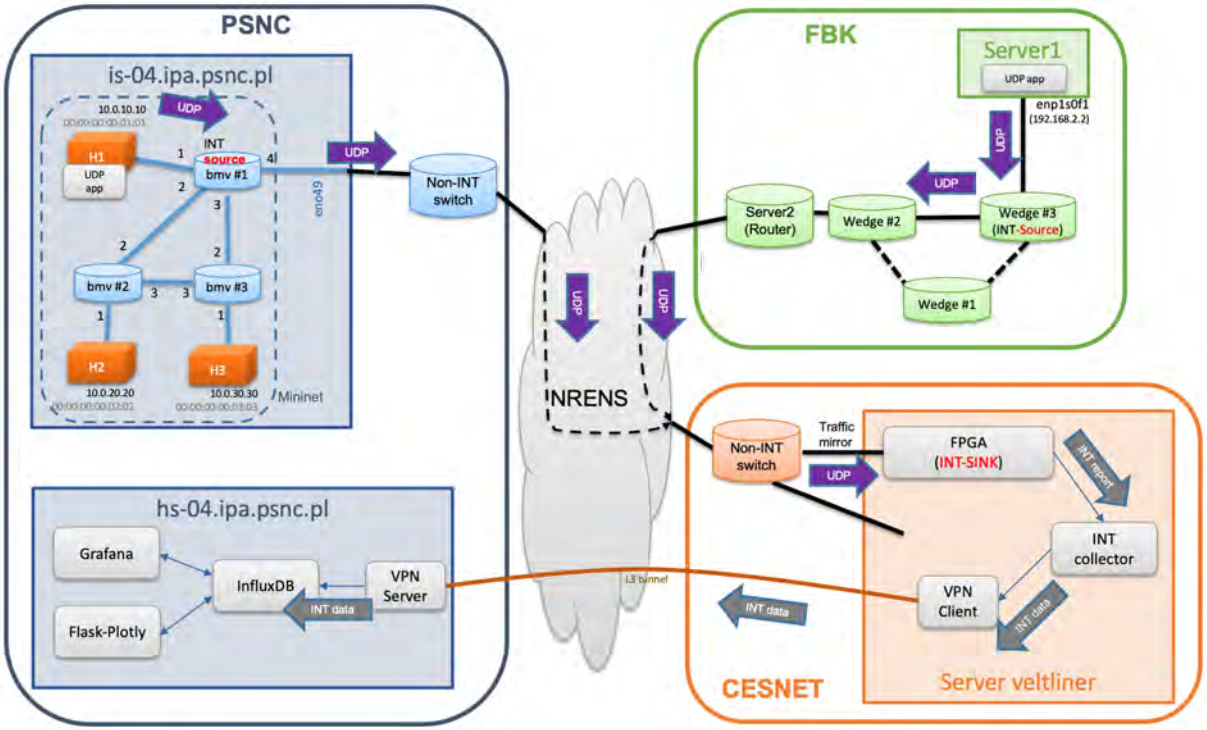
INT header element	Size [bytes]
Main INT header	16
✓ Switch ID	4
✓ Ingress Interface ID	4
✓ Egress Interface ID	4
✓ Ingress Timestamp	8
✓ Egress Timestamp	8
Hop Latency	4
Egress Port TX Link Utilization	4
Queue Occupancy	4
Buffer Occupancy	4

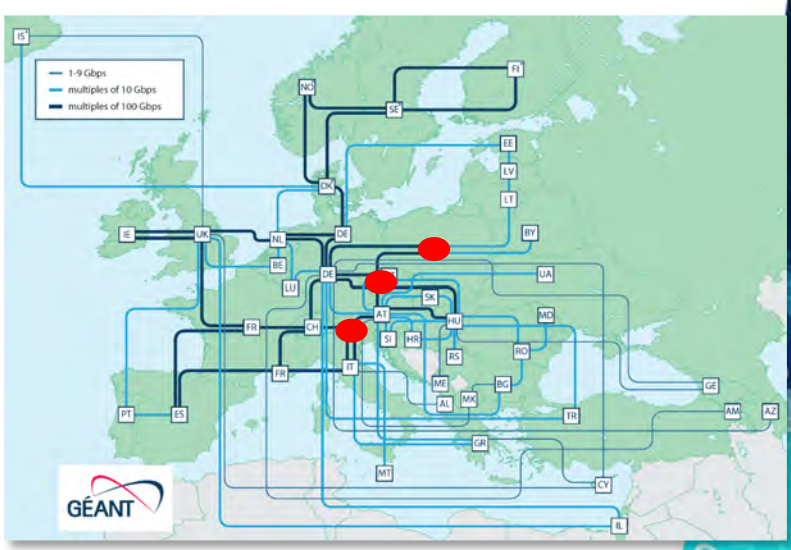
28 bytes x (INT_hops-1)

44 bytes x (INT_hops-1)

Testbed

INT: testbed over production NREN networks





- 3 switch types
- UDP packets flow on NRENs networks
- Collected INT data in CESNET is sent back to PSNC for collection and presentation.

Challenges encountered in the testbed and their solutions

- VPN (encryption) vs. Public IPs
  - VPNs introduce noise in jitter measurements
  - *Solution*: switch to public IPs, unencrypted traffic
- Remote Collector bandwidth bottleneck
  - Sink-Collector TCP connection couldn't provide sufficient bandwidth at all times
  - *Solution*: use multiple parallel TCP connections to avoid hanging on a single missing ack
- Firewalling
  - Sink firewall blocking INT packets
  - *Solution*: whitelist expected source IPs

PoP specifications

- FBK
  - 2x Servers (Intel Xeon X3480, 32GB RAM, 10Gbps NICs, Debian)
  - 2x Wedge100BF-32X Tofino Programmable Switches
  - *Challenges*: OS support for SDK & INT (-> Open Networking Linux), no support for setting the clock on the ASIC side
- PSNC
  - 2x HP ProLiant DL380 Gen9 server equipped with Intel Xeon E5-2650, 40vCPU, 160GB RAM and 1.2TB SSD raid disk, Ubuntu
  - 2x Arista 7071 Tofino 32x 100G
  - *Challenges*: Mininet performance, collector I/O & query performance, Tofino-gRPC wrapped by Arista libraries (lack of access to some P4 resources)
- CESNET
  - 1x COMBO-200G2QL FPGA
  - 1x Server

INT Platforms: Lessons Learned

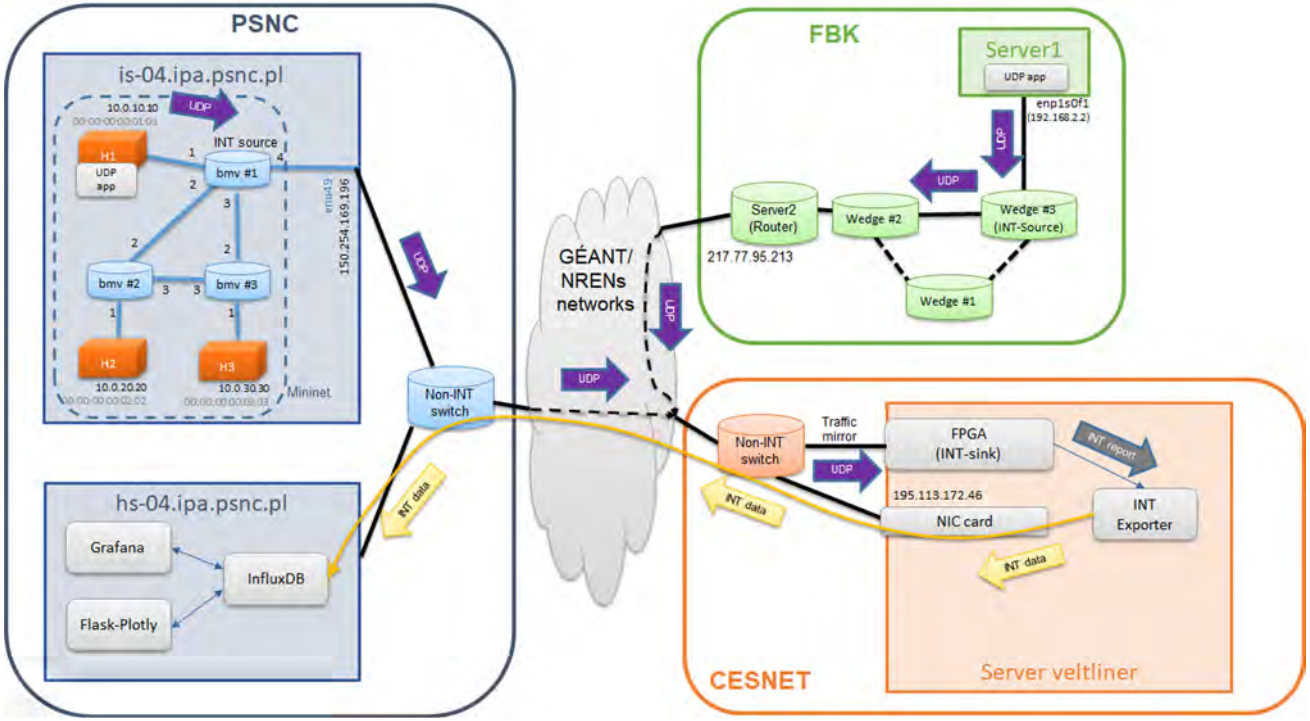
Pick the solution that works for your use case!

- Bmv2/mininet: good to start initial P4 code development!
  - Performance ceiling, virtual routing
- Tofino switches: potentially feature-rich for P4
  - Clock synchronisation, licensing, complexity issues
- FPGA card: fast, flexible HW
  - P4 compiler vital; CESNET compiler only for P4_14
  - HW expertise may be required for some features
- INT-DPDK:
  - Promising performance up to ~10G
  - Needs careful selection of NICs

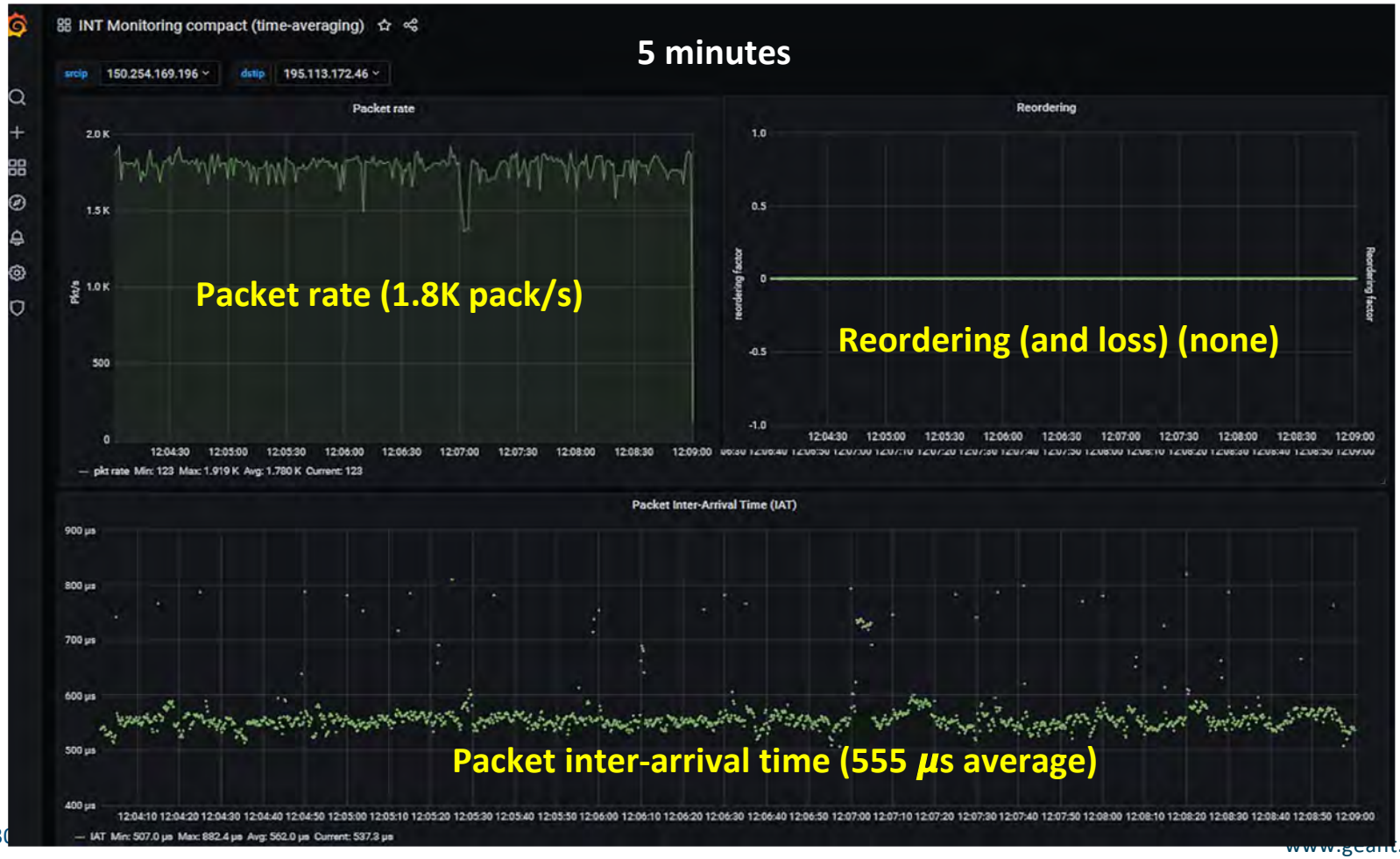
Measurements

What we have measured?

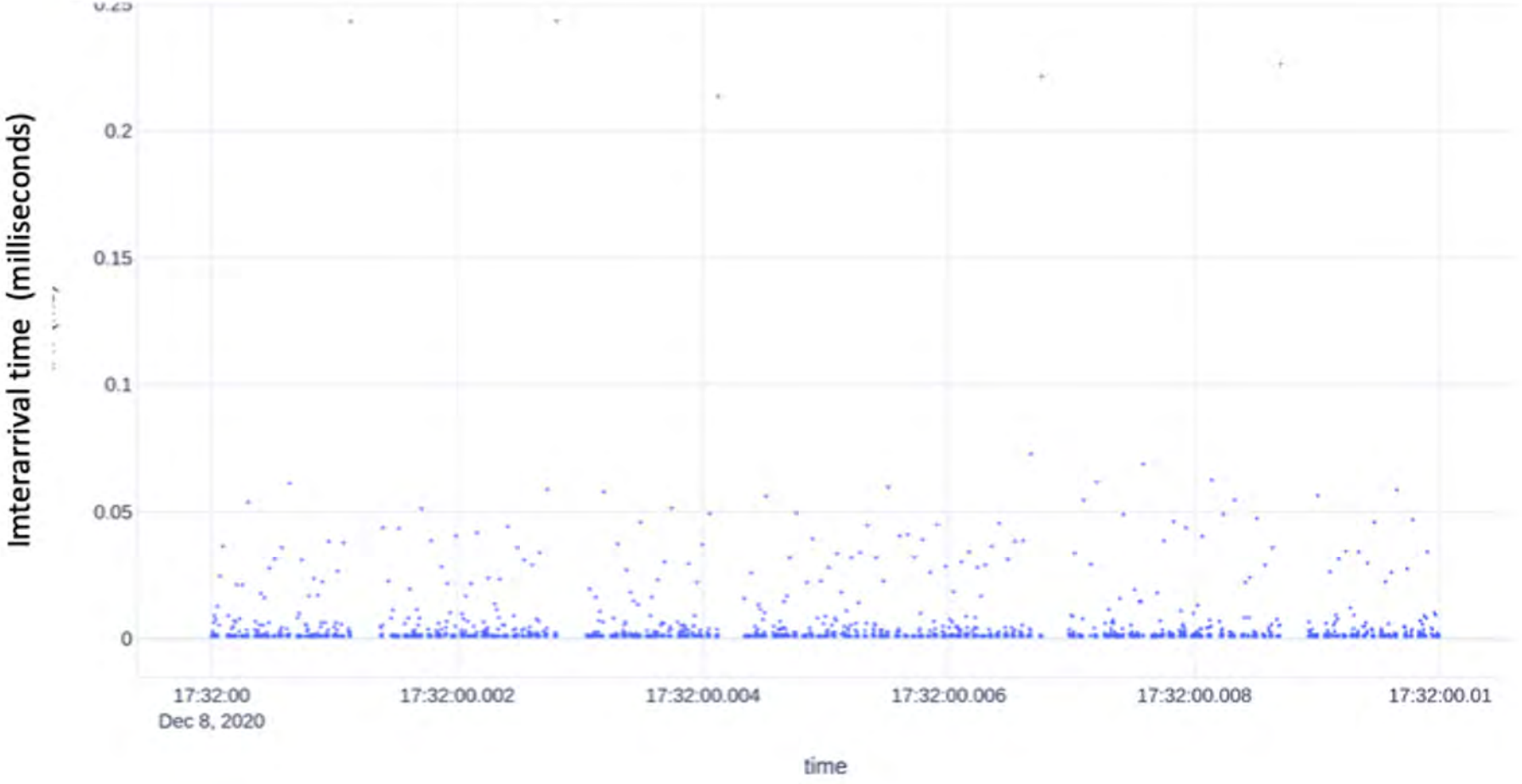
- Packet inter-arrival time and losses between Source nodes (PSNC, FBK) and Sink node (CESNET)
- Input packets generated by Source nodes with constant rate 1k - 300k pps



5 minutes of the INT monitored flow from PSNC to CESNET

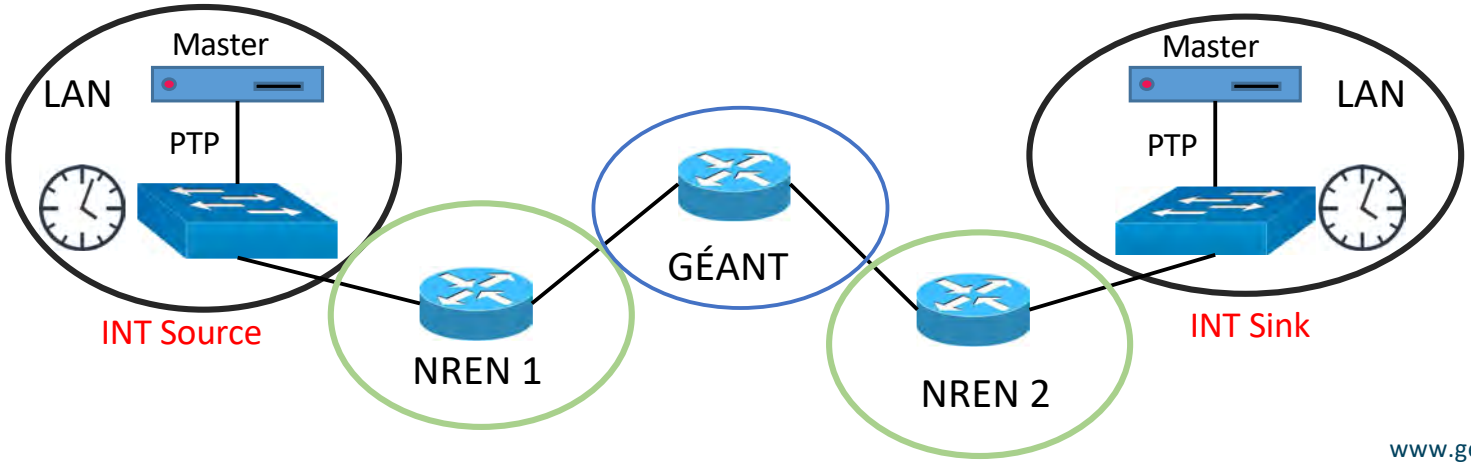


10 ms of inter-arrival packet time for a UDP flow of 260 K pps (4 microseconds average) from FBK to CESNET



Clock synchronization

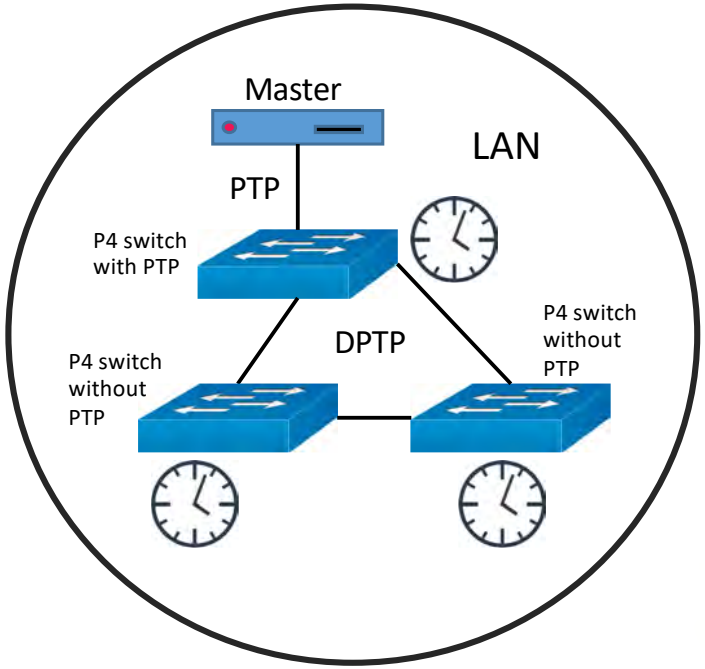
- INT use-case
  - To measure delay, INT Source, Transit and Sink must be synchronized
- Network Time Protocol (NTP)
  - Precision: few ms
- Precise Time Protocol (PTP - IEEE 1588)
  - Precision: < 1us
  - Hardware support required



Limited hardware support for precise clock synchronization

- Tofino-based switches
  - 48-bit Timestamps
    - Time is reset every 3 days
    - Without IEEE 1588 PTP support
  - Complex operation (multiplication)
    - Some computation offloaded to Sink Node
- FPGA-based SmartNICs
  - Current impl. uses NTP
  - PTP synchronization can be implemented additionally
- Commodity NICs
  - Intel: HW Timestamps for selected packets only
  - Mellanox: HW Timestamps for receiving packets only

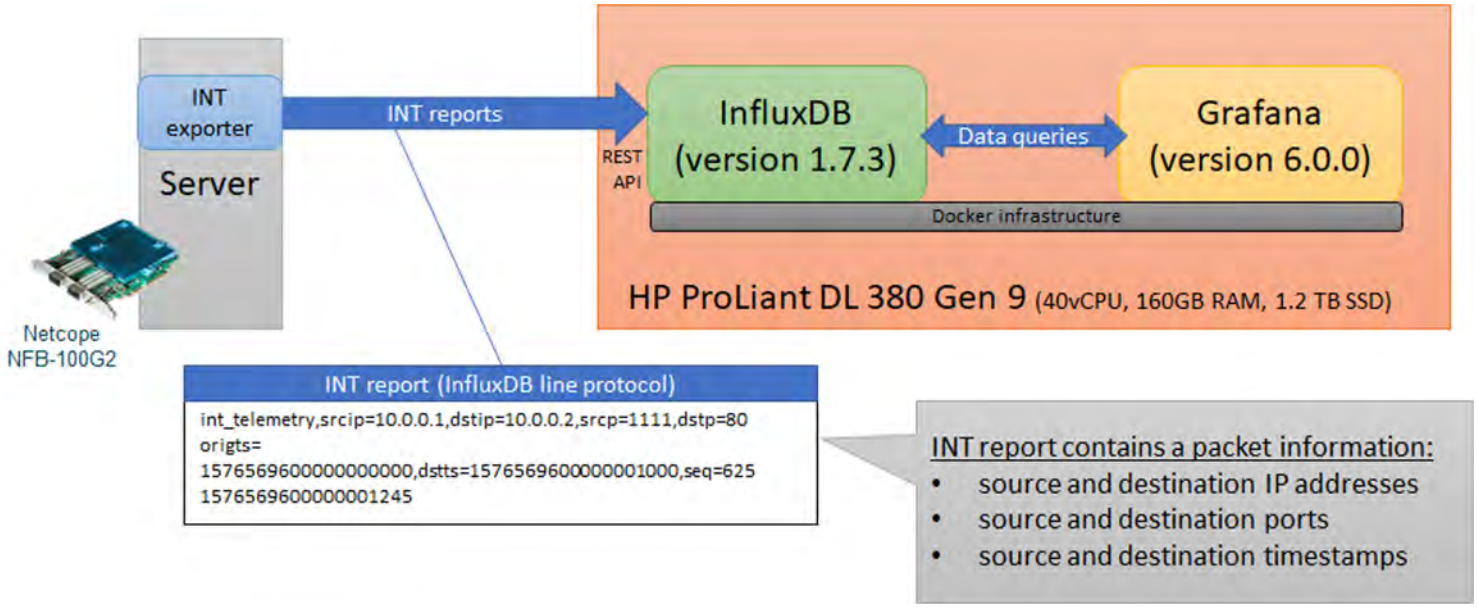
- Solution
  - DPTP approach combined with IEEE 1588 PTP enabled switch



Big Data

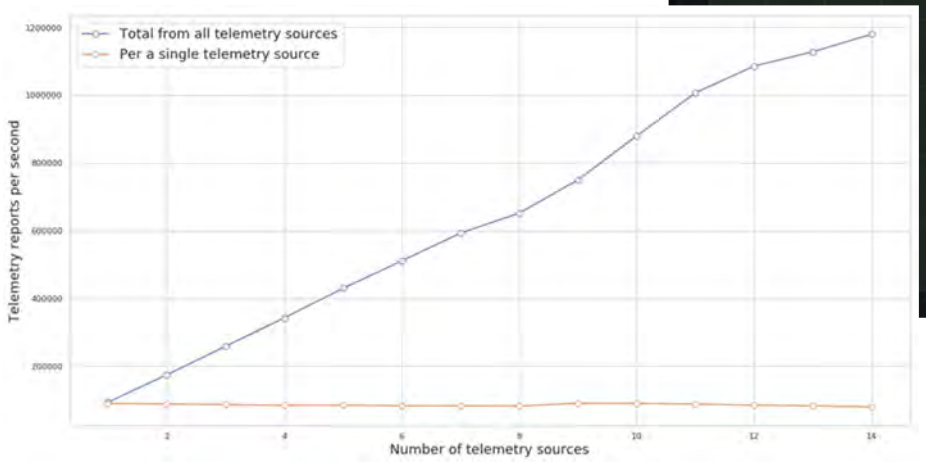
Storing time series data for visualisation in our testbed

- Initially stored just basic INT measurements
- Currently data transformation done by the INT exporter



Testing limits of single InfluxDB instance

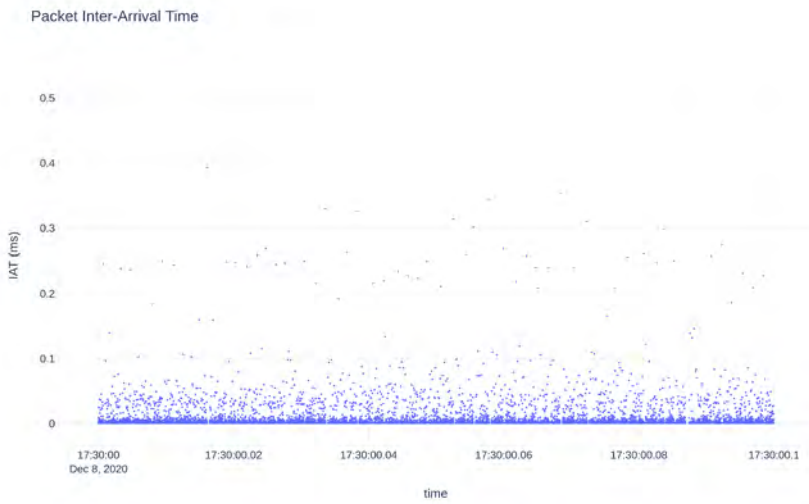
- Using **a single TCP connection** indexing ~50K INT reports/s
- **~260K INT reports/s** using 20 parallel TCP connections (~500Mbps traffic of INT reports)

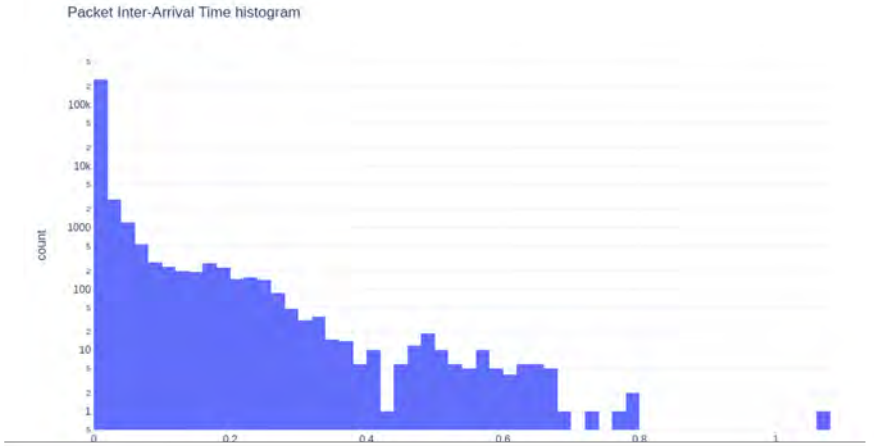


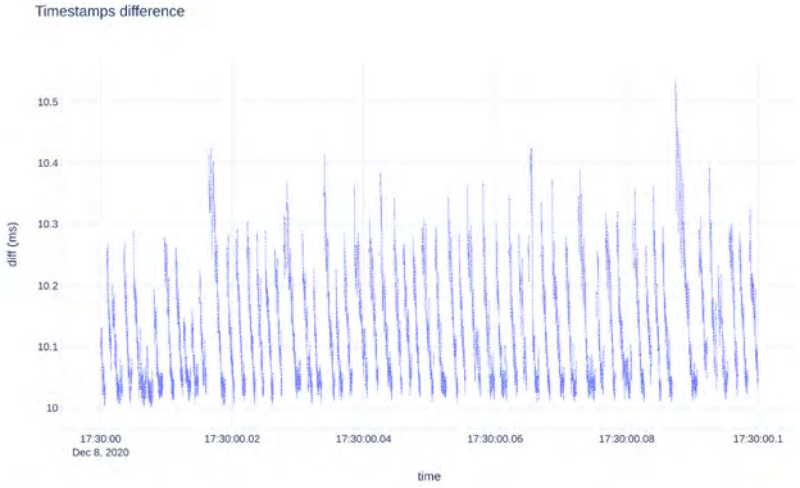
INT visualisation in Grafana (online)

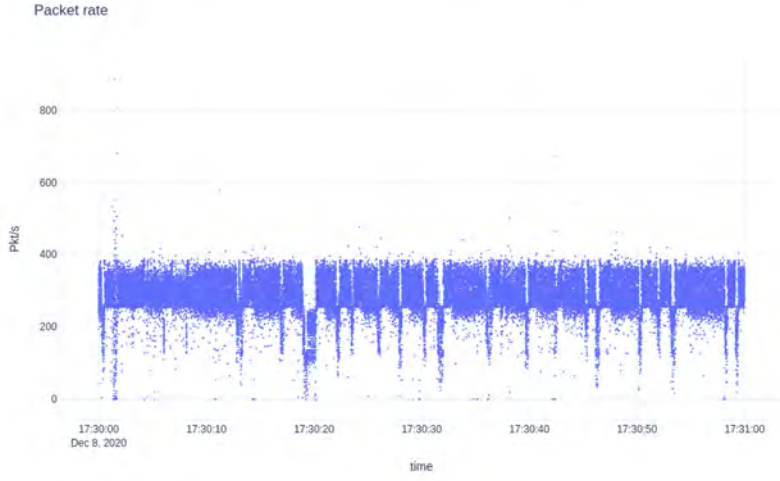
- Don’t do INT data **post-processing** in Grafana queries
- **Time series resolution** down to 1 ms only
- **Time-averaging queries** required for smooth graphs loading (up to 100K INT reports/s)
- **InfluxDB continuous queries** performing INT metrics averaging not helpful

INT visualisation using Plotly (offline)









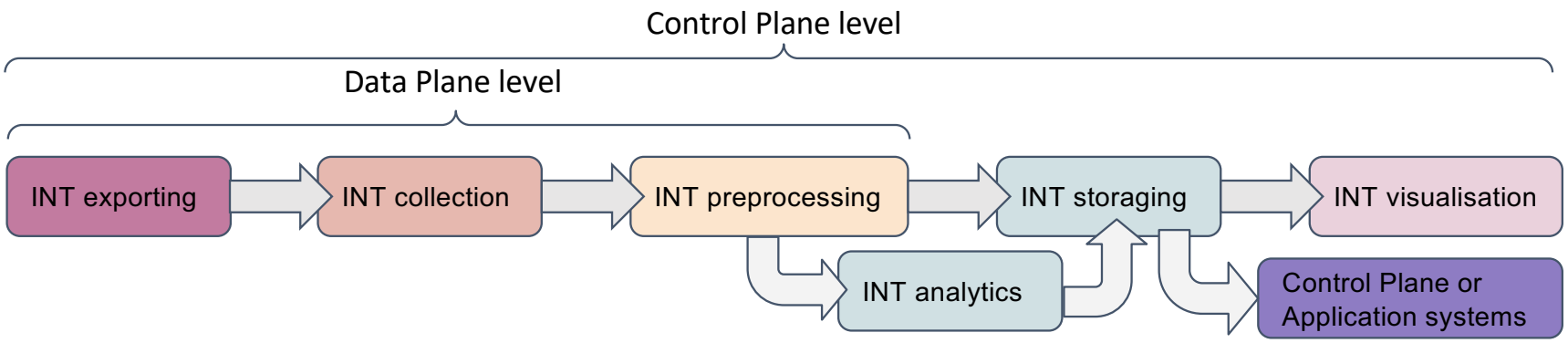
Data “velocity” in In-Band Telemetry

Assuming that every flow packet is monitored:

Flow rate	Only 64B packets (+20B interpacket gap)	Only 1518B packets (+20B interpacket gap)	Only 9018B packets (+20B interpacket gap)
100Mbps	149K reports/s ✓	8.13K reports/s ✓	1.38K reports/s ✓
1Gbps	1.49M reports/s	81.3K reports/s ✓	13.8K reports/s ✓
10Gbps	14.9M reports/s	813K reports/s	138K reports/s ✓
100Gbps	149M reports/s	8.13M reports/s	1.38M reports/s

- Performance must be scaled by the number of simultaneously monitored connections
- INT reports require near real-time data processing or batch processing:
  - Generate events (anomalies)
  - Calculate aggregated statistics
  - Provide visualisation

Improving INT scalability (high-rate flows, more flows)



P4 language

eXpress Data Path

Data Plane Development Kit

extended Berkeley Packet Filter

Remote Direct Memory Access

Data filtering

Key events finding

Data aggregation

Time resolution

Kafka Streaming cluster

ElasticSearch cluster

InfluxDB Enterprise cluster

Spark cluster

Data Plane level

More generality and flexibility, bigger server clusters

Many limitations, hardware special requirements

Control Plane level

Next steps and summary

INT Status and developments

- INT **P4 code** for these tests, using INT Spec 0.4, is **available from Github** for the three platforms
- Data collection and presentation tools and configuration optimized
- **INT transit** code under testing
- Upgrading to to **INT standard 1.0 and then 2.0**
- Improving **clock stability and synchronization**
- I**NT code implementation over DPDK** to provide an **INT tool** that does not require special hardware, runs in off-the-shelf computing equipment and that can provide Gigabit/s performance (in testing phase)

Next steps

- Development of a **BMv2 INT docker image** as a tool to download for easier testing and trials with INT basic use cases and P4 programming (timestamps, sequence numbers, including collection and visualization, based on what is already being used in PSNC
- **Extend testing topology** (up to transatlantic?)
- Introduce "**Big Data**" technologies
- Establish **collaborations** to:
  - Identify and develop **new use cases**
  - Further **improving** the basic tools
  - Discuss and disseminate the **knowledge** gathered
  - In-depth **data analysis**
  - **Standardize** approach to INT and Data Plane Programming

Summary

- **INT** (and Data Plane Programming) (using P4) is not business-as-usual, requires specific expertise, however it offers a great technology for **monitoring, debugging and providing information to control plane, in real time.**
- P4/INT is more and more available in various platforms (switches and linecards, software)
- INT is a powerful **magnifying glass** on network behaviour
- Time synchronization between nodes is important
- As a function of the use case, the INT/P4 bases tools may require handling of **large amount of "raw" data**, to be used for analytics and more. It implies the development of further insight, **knowledge** and specific tools and equipment to scale.

More information

- **Data Plane Programming / INT GEANT web page**
  https://wiki.geant.org/display/NETDEV/INT

- The GÉANT **First Telemetry and Big Data Workshop**
  https://wiki.geant.org/display/PUB/Telemetry+and+Big+Data+Workshop
- **INT Tests in NREN networks** – DPP WP6 T1 sub-task Report – To be published as a white paper :
  https://docs.google.com/document/d/19tnC2AZgPkXlrVX80b5D2sZsTjzpg4WURW9z432fw1c/edit?usp=sharing
- "**A Survey on Data Plane Programming with P4**: Fundamentals, Advances, and Applied Research", Frederik Hauser, Marco Häberle, Daniel Merling, Steffen Lindner, Vladimir Gurevich, Florian Zeiger, Reinhard Frank, and Michael Menth (50 pages). 26 Jan 2021, to be published in" Communications Surveys & Tutorials (COMST) journal -- *https://arxiv.org/pdf/2101.10632.pdf*

  For INT use cases, see section D, page 22

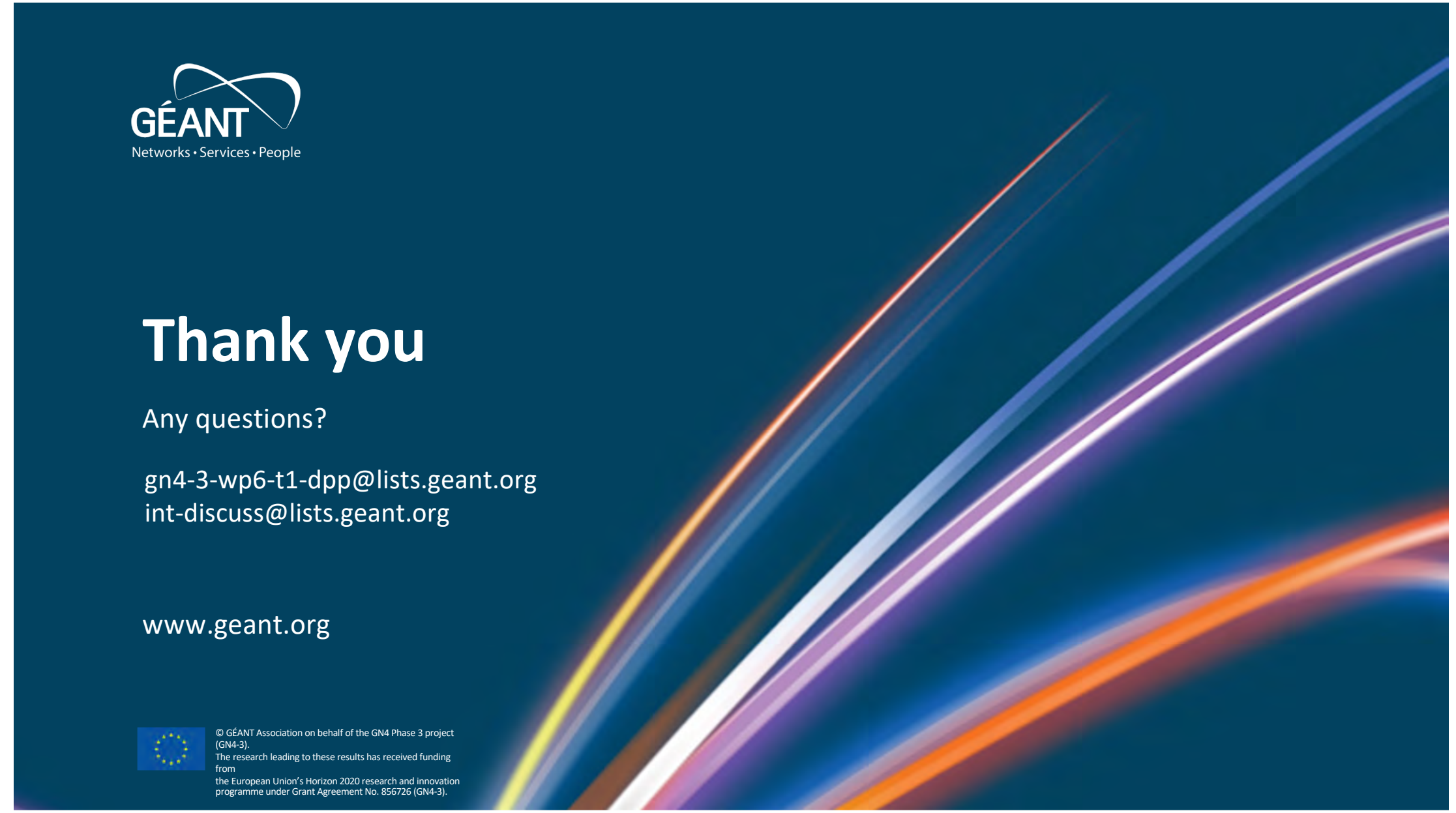
Closing the presentation

A short **mentimedia** questionnaire will follow

http://www.menti.com code : **24 21 59 5**

The INT/DPP group is willing to **collaborate and share** results contact **gn4-3-wp6-t1-dpp@lists.geant.org**

INT mailing list : **int-discuss@lists.geant.org**.



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).