

Diving in Data Plane Programming, Silicon and Oceans of Data

Mauro Campanella (GARR), Tomas Martinek and Mario Kuka (CESNET), Joseph Hill (UvA), Matteo Gerola (FBK), Jakub Kabat (PSNC), Marinos Demolianis and Nikos Kostopoulos (NTUA), Theodore Vasilopoulos (GRNET)

TNC22, Trieste
15 June 2022

Agenda

Why Programming the Data Plane

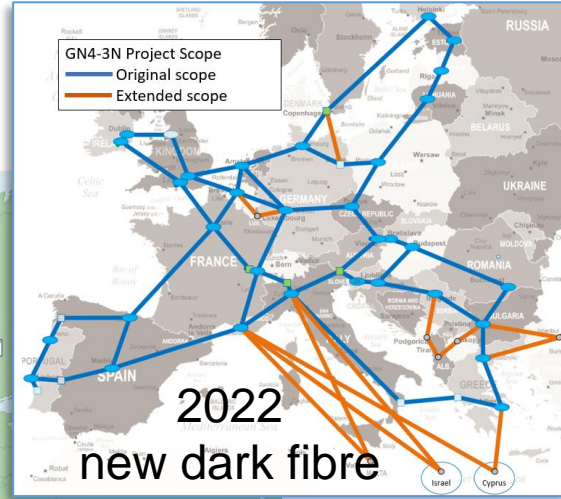
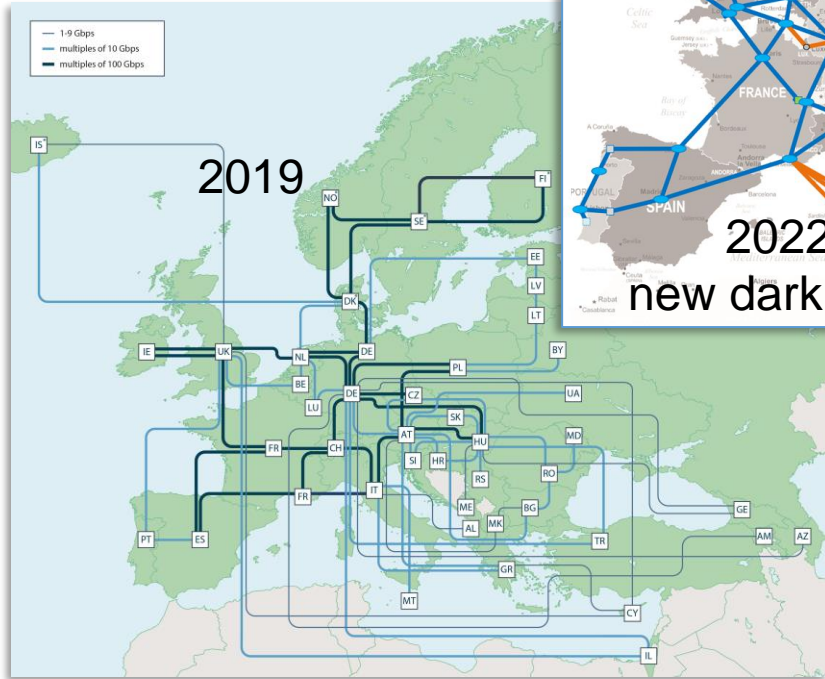
Who and Where

What and some outcomes.

Knowledge collected



GÉANT fibre Topology

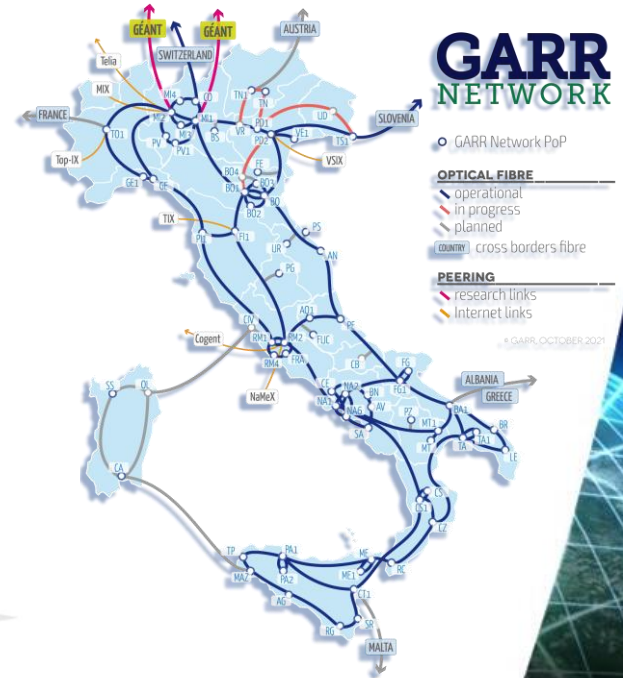


Traffic Growth



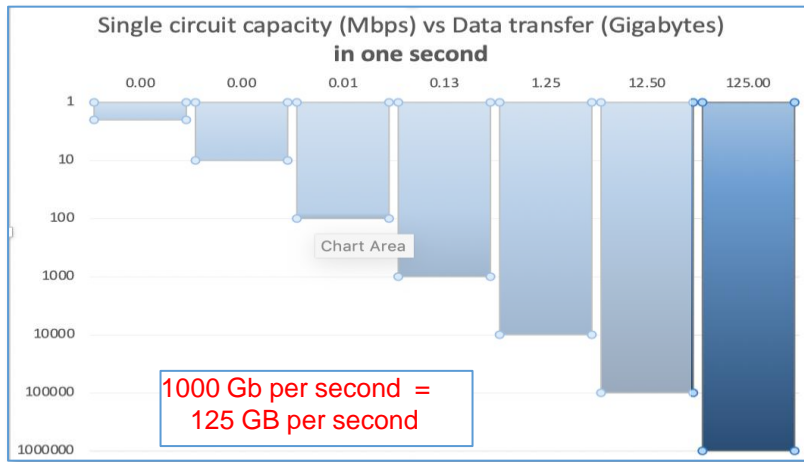
+30%

Average annual increase in network traffic over last five years



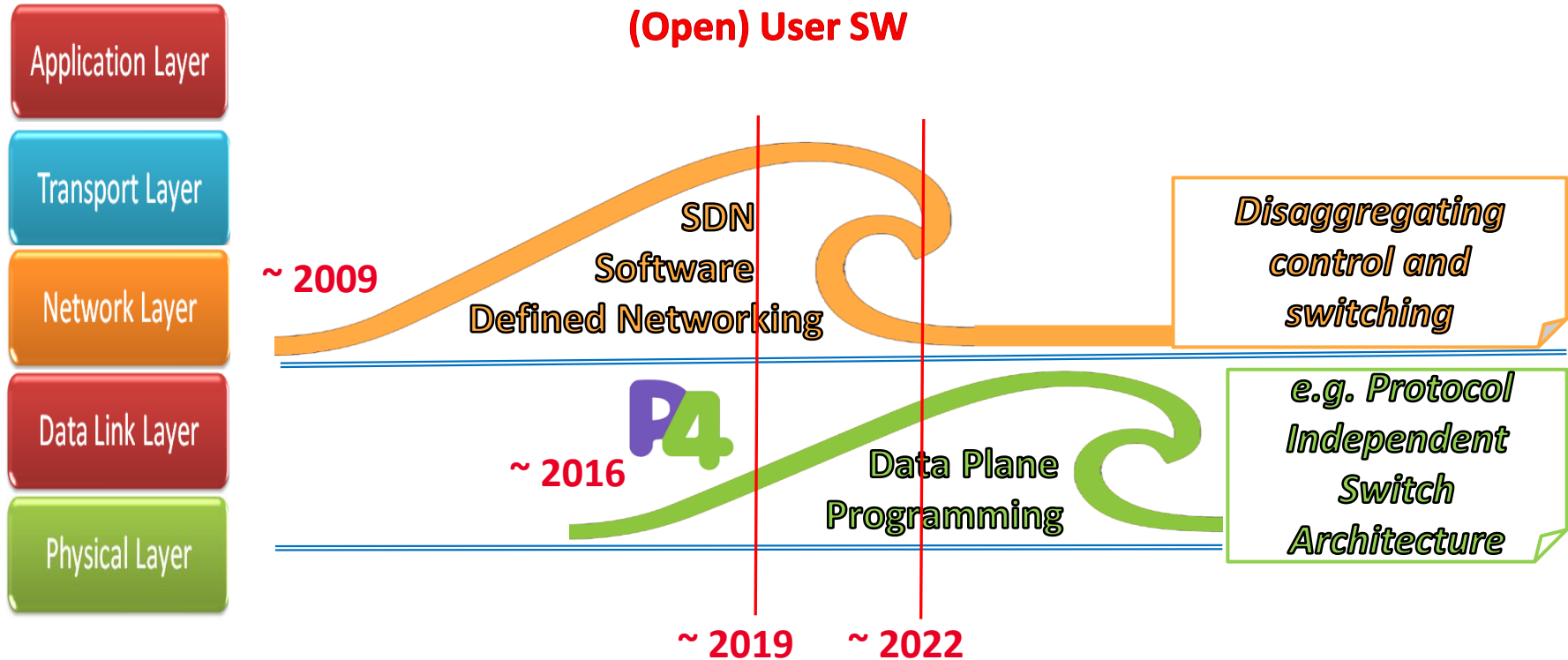
Why this effort ?

- Networks' **growth** (in capacity and size) “may” need **innovative tools** for control and monitoring
- Contribute to the evolution toward Automated, Programmable, Agile, and Personalized networks
- Consider **security** and **real time applications challenges @ 1Tbps**
- **Integration of WAN, LAN, Clouds, Edge in softwarized “platforms”**



New Knowledge on how networks behave at these new “depths”

The opportunity : innovation in Programming the network stack



Programming the data plane

- Applying "non switching" logic to all/selected packets
- Altering the structure of all/selected packets with "in-house" logic
- Improve and personalize monitoring beyond streaming telemetry
- fast-feed data plane "new" information to control plane

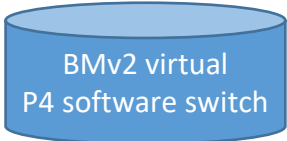
Elements became available :



Programming Protocol-Independent Packet Processors :
High level, C-like, coding language for controlling packet forwarding planes in networking devices

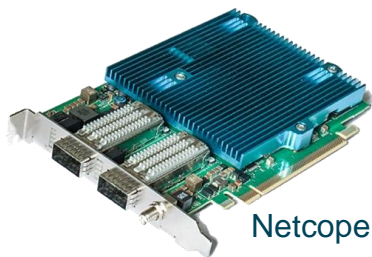
- New silicon (Tofino), FPGA, all P4 enabled at line speed

P4 code developed for these platforms (on GitHub)



BMv2 virtual
P4 software switch

BMv2 Behavioural Model v2 – emulation of Tofino
Uses Mininet



Netcope NFB-100G2

FPGA P4 compiler developed by Cesnet up to 2x100
Gbps



Edgecore Wedge100BF-32X

Tofino 1 Asic engineered for data plane programming by
Barefoot (now Intel), specific SDE



P4 on DPDK "off-
the-shelf" PC HW

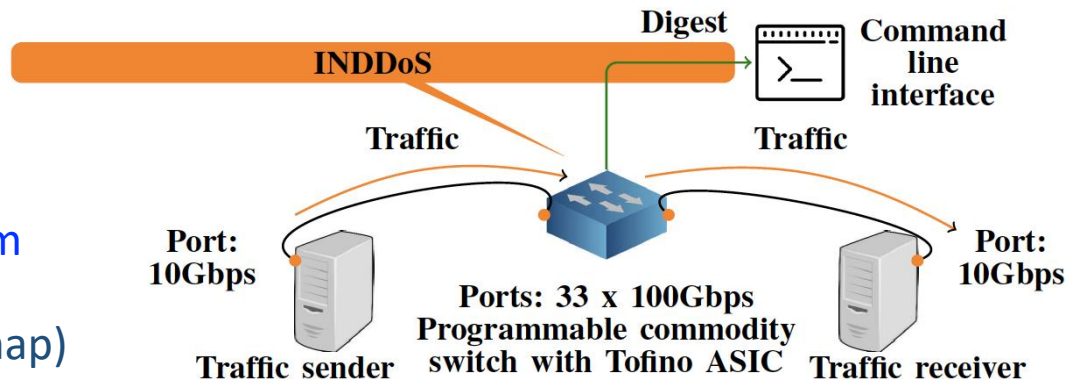
DPDK (Data Plane Development Kit) kernel software
acceleration, 2 P4 to DPDK compilers (T4P4S
and P4C-DPDK)

First Use case: DDoS detection in the data plane

A P4 program:

Threshold based identification.

Developed an algorithm (BACON) (based on count min-sketch, Bitmap) to minimize memory occupancy, preserving identification precision of attackers and targets for further decision on mitigation by an external controller.



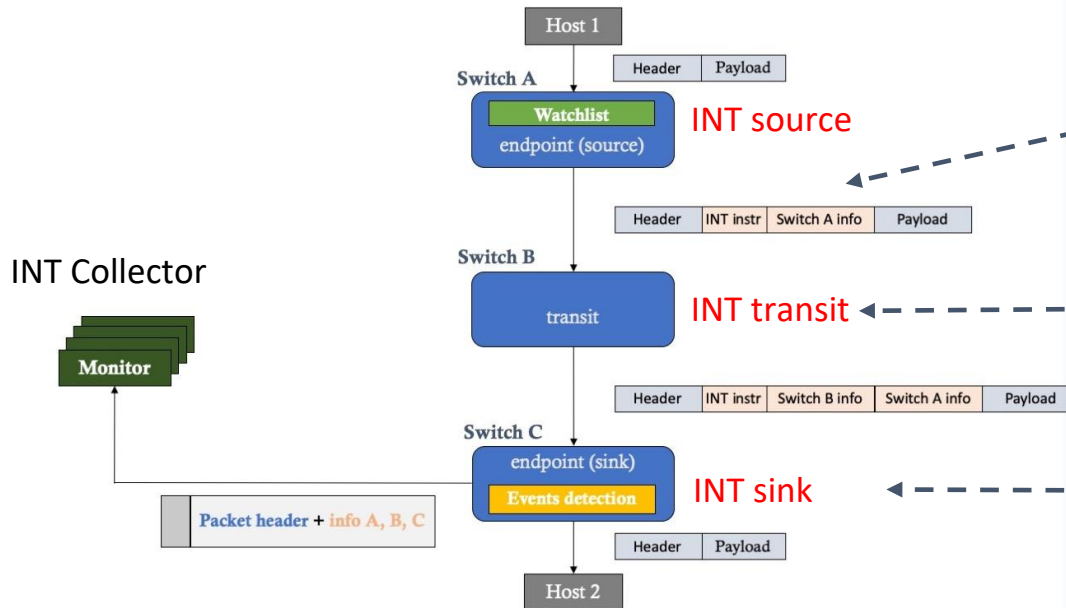
BACON Sketch size ($d \times w \times m$)	Recall	Precision	F1 score
$3 \times 1024 \times 1024$	0.96	0.99	0.97
$1 \times 2048 \times 1024$	0.98	0.54	0.70
$1 \times 1024 \times 2048$	0.94	0.38	0.54
$5 \times 1024 \times 512$	0.12	1.0	0.22
$5 \times 512 \times 1024$	0.96	0.89	0.92
Spread Sketch size ($d \times w \times m$)	Recall	Precision	F1 score
$3 \times 1024 \times 1024$	0.92	0.94	0.93

DDoS : lessons learned

- Developing the P4 program in the virtual environment facilitates prototyping, **implementing in hardware requires careful tuning**
- **Efficient** identification, threshold based, in the data plane at line-rate works well, within scale ranges
- **Algorithms require many/all processing stages in the Tofino ASIC.** The P4 program has to be limited in complexity to preserve line-rate processing and preserve staged for switching logic
- Central processing and memory resources are **only partially used**
- **Processing time increases packet delay slightly (about 100 ns)**

Second Use case: In-Band Network Telemetry (INT)

INT alters structure of selected packets, in the fly, to store and transport information in-band, in real time



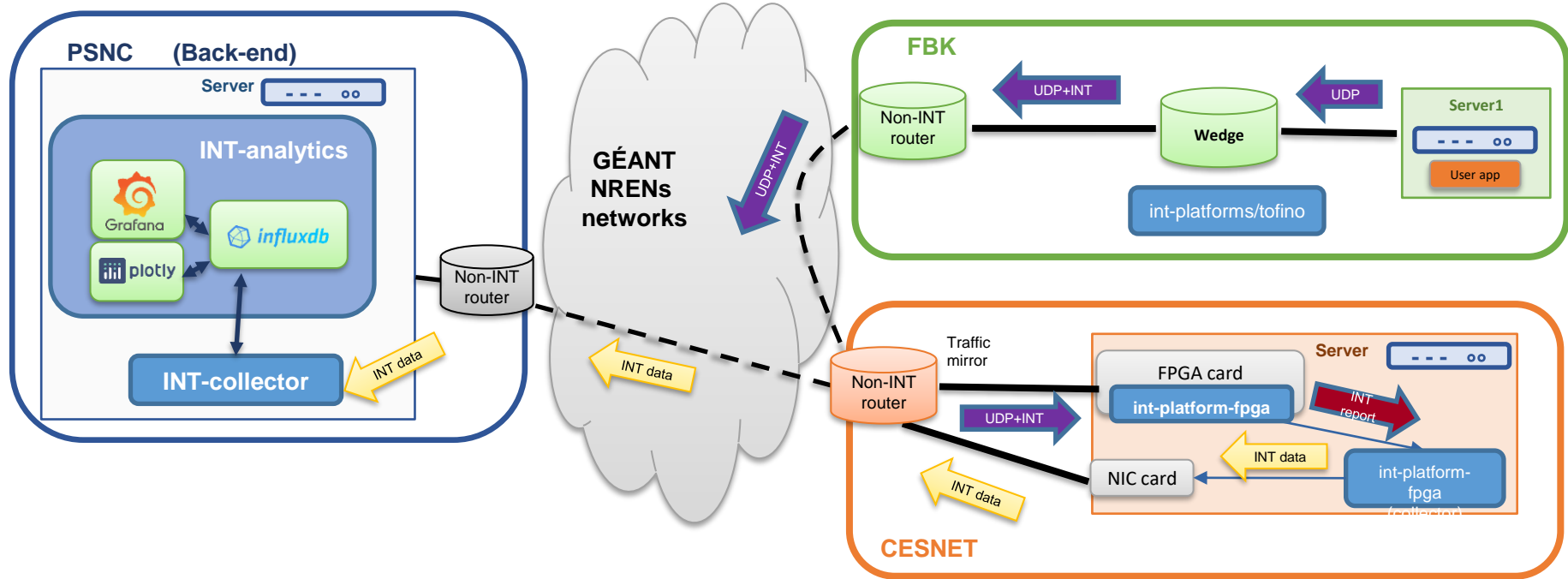
INT functions

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

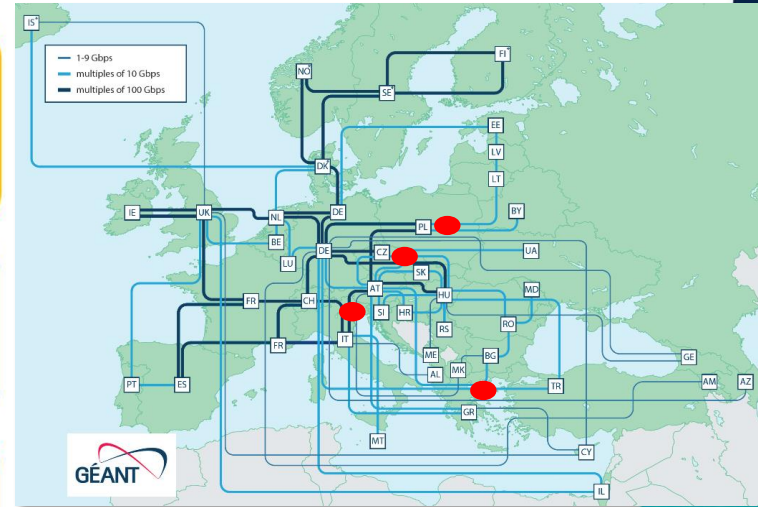
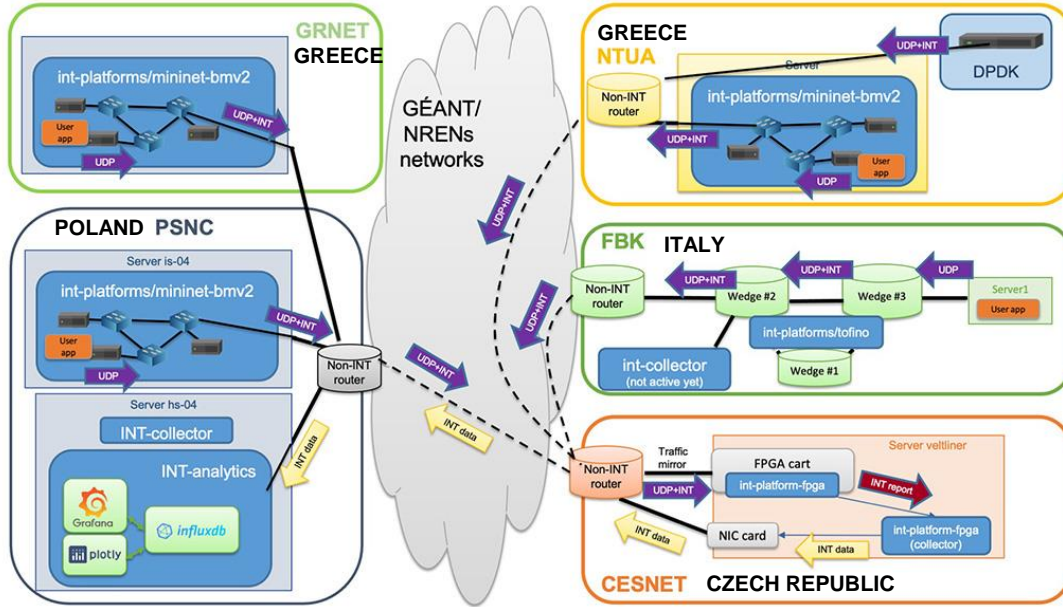
INT transit nodes add specific local

The last **INT sink** node extracts, may analyze and and sends information to a collector

INT test set-up (FBK to Cesnet)



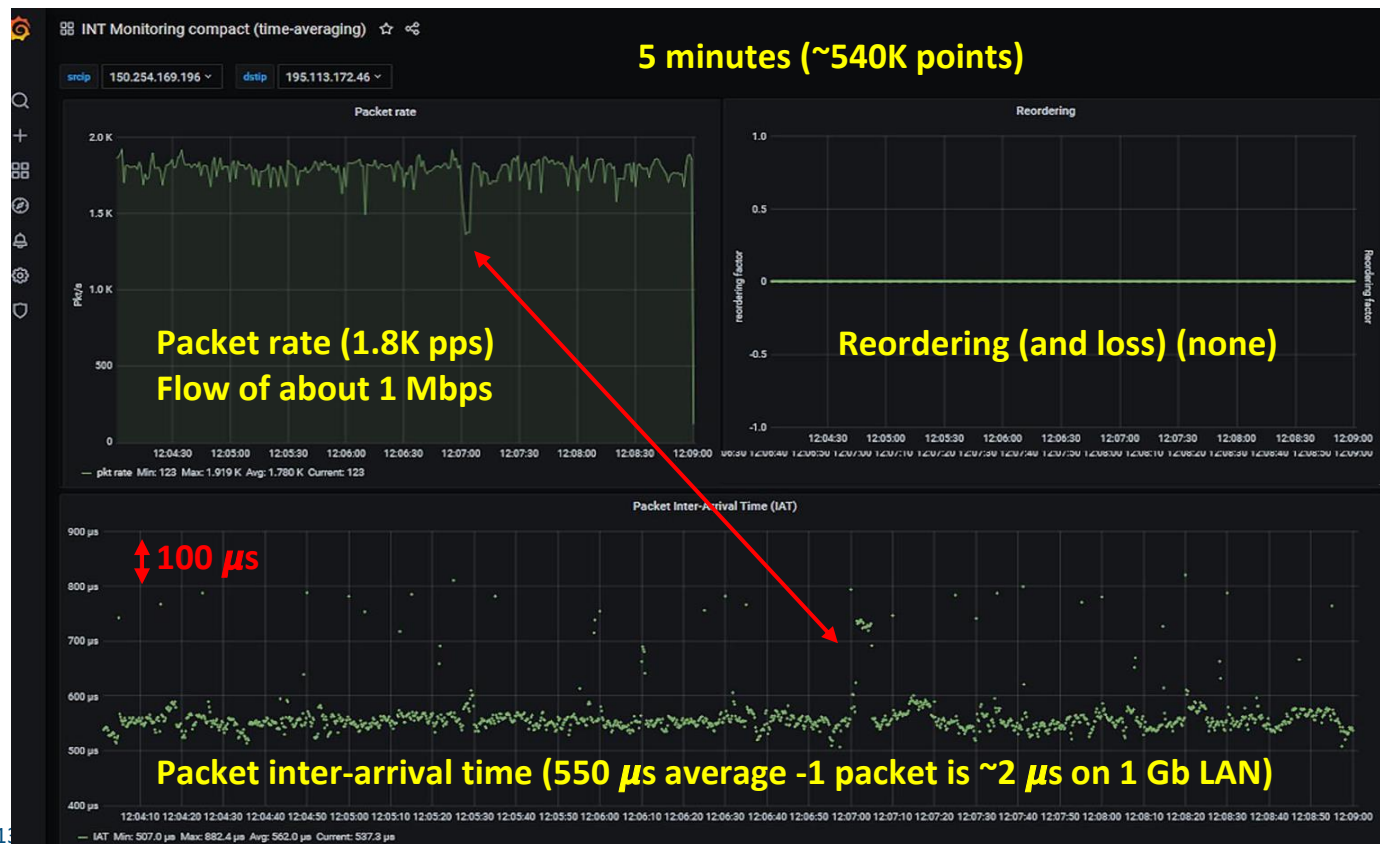
Our INT testbed over production NREN networks



- Packet carries **timestamps**, **sequence number in INT headers** between Source (all) and Sink node (CESNET)
- UDP packets generated at constant rate ~1k to 300k pps

- 4 switch platforms
- UDP packets flow in NRENs networks
- Collected INT data in CESNET is sent to PSNC for collection and presentation.

5 minutes of the INT tags in a single UDP flow PSNC to CESNET



Testing NRENs' networks effect on a flow – first results

Measure at the source a flow inter packet gap (IPG) as the time between subsequent pairs of packets (function of capacity/packets per second)

Measure at sink for the same flow the IPG between same pairs

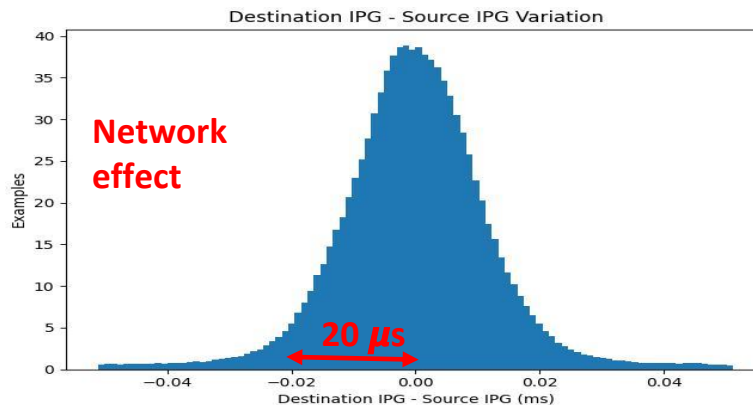
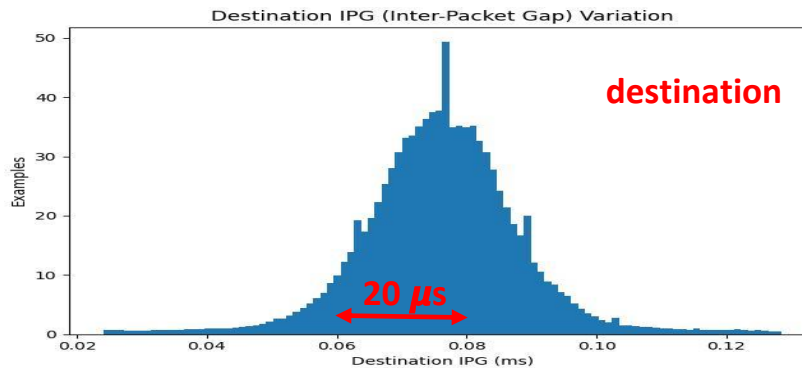
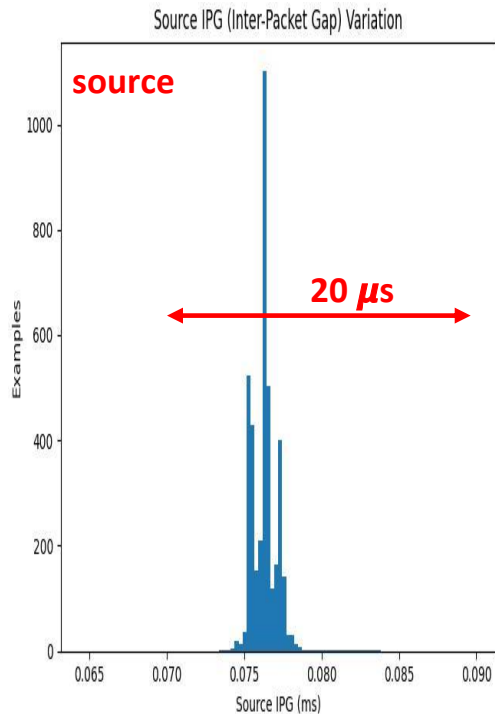
Subtract source IPG from sink IPG to remove variation due to the source

If the networks behave “well”, it should preserve the flow characteristics and the IPG at the source and at the end should be almost identical

Note:

- Timestamp are set in HW (ns precision) at switch ingress
- No packet losses/reordering have been noticed
- Collector-InFluxDB communication and performance sub-optimal still

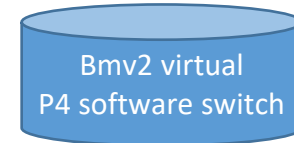
Testing NRENs' networks effect on a flow



2Mbps (user), 120 sec, payload 20B, expected IPG 76.29 μ s, 13108 pkts/s

Platforms for INT : Lessons Learned

- BMv2/mininet: good to set-up initial P4 program development (performance/jitter ceiling)
- Tofino switches: feature-rich for P4, limited central memory, non standard timestamping
- FPGA card: very flexible HW, large memory
 - P4 compiler quality: vital
 - Specific HW insight is required
- INT-DPDK :
 - Promising performance up to few Gbps
 - Needs careful selection of Network Interface Cards, (timestamping, etc...)
 - P4 to DPDK compiler quality may vary



Challenges, engineering and sustainability

- **Expertise** on a tight integration of HW and software. Almost all participants left to private environment since the start of the effort
- **Fast software updates** in each platform requiring P4 program re-validation
- **Not sufficient standardization on timestamp** format, HW implementation and protocol (e.g. PTP) support
- A strong backend is also needed for data collection, storage, presentation. Essential for scalability

Final considerations

- NRENs' networks handle well flows at sub-second scale or lower
- Data Plane Programming) (using P4) is **not business-as-usual**, requires specific, scarce expertise (fastly changing, mixing ICT, network and data)
- The request for specific HW does not facilitate wide implementation yet
- INT offers an agile tool for high frequency **monitoring**, supporting new **control plane** mechanisms and debugging in real time. INT can be used by both providers and independently by end-users
- Monitoring network at second, or lower, scale provides complex results , and transient effects are evident also at milliseconds range

Acknowledgements

The effort reported has received essential contributions from many GÉANT participants and specifically:

Damu Ding (Italy), Federico Pederzoli (Italy), Pavel Benacek (Czech Republic),

Tim Chown (Jisc UK), Ivana Golub (PSNC Poland),

Xavier Jeannin (RENATER France)

Thanks !

More information on GÉANT Data Plane Programmability activity

- **Data Plane Programming / INT GEANT web page** <https://wiki.geant.org/display/NETDEV/INT>
Includes all documents produced and a **pointer to GitHub INT P4 code**
- **Mailing list:** <https://lists.geant.org/sympa/subscribe/int-discuss>,
- **White Paper INT Tests in NREN networks** – DPP WP6 T1 white paper
https://www.geant.org/Resources/Documents/GN4-3_White-Paper_In-Band-Network-Telemetry.pdf
- **DDoS Paper:** "In-Network Volumetric DDoS Victim Identification Using Programmable Commodity Switches", F. Pederzoli, M. Campanella and D. Siracusa, in IEEE Transactions on Network and Service Management, Vol. 18, Issue: 2, June 2021, page: 1191-1202, DOI: 10.1109/TNSM.2021.3073597 and at <https://arxiv.org/abs/2104.06277>
- **The GÉANT First (and second)Telemetry and Big Data Workshop**
<https://wiki.geant.org/display/PUB/Telemetry+and+Big+Data+Workshop>
<https://events.geant.org/event/1104/>

Non GEANT References

- **The Programmable Data Plane Reading List** : <https://programmabledataplane.review/>
- Oliver Michel, Roberto Bifulco, Gábor Rétvári, Stefan Schmid, "**The Programmable Data Plane: Abstractions, Architectures, Algorithms, and Applications**", ACM Computing Surveys, Volume 54, Issue 4, May 2021, Article No.: 82, pp 1–36, <https://doi.org/10.1145/3447868>
[10.36227/techrxiv.12894677.v1](https://arxiv.org/abs/10.36227/techrxiv.12894677.v1) <https://www.univie.ac.at/ct/stefan/csur21.pdf>
- "**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>
- Kaur, Sukhveer & Saluja, Krishan & Aggarwal, Naveen. (2021). "**A review on P4-Programmable data planes: Architecture, research efforts, and future directions**". Computer Communications. 170. 10.1016/j.comcom.2021.01.027.
- **IOAM**: <https://datatracker.ietf.org/wg/ioam/about/>

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).