In-band Network Telemetry

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

Workshop on Network Management and Monitoring Tools
24 March 2021

www.geant.org
Why working on INT (data plane programming)?

Control and debug more complex networks:
• Larger "networks", more protocols (LAN, WAN), more virtualization
• Rising security threats (DDoS e.g.)
• More bits per wave per unit time (Tbps soon)
• Still "slow monitoring" style and analysis

Started with an "inovative" approach
• Programmability outside the "locked" router/switch and packet format:
  Programming language for controlling packet forwarding planes in networking devices: Programming Protocol-Independent Packet Processors
• New silicon P4 enabled (FPGA, Tofino, ...) for line speed
In-Band Network Telemetry (INT) short summary

INT can report any info from switch (id, queue depth, temperature, ingress or egress interface, cpu load e.g.), packet (timestamps e.g.) and any info computed in DataPlane (28 bytes overhead or more / packet)

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
INT implemented in P4

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 **software** and **Intel/Barefoot Tofino** chipsets (Arista, Cisco, EdgeCore, Accton, ...), **Broadcom**
- **SmartNIC/FPGA cards** (Agilio, Xilinx, Intel, Pensando, Netcope, ...)

```c
#include <e1model.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(); }
```
What INT can tell about your packets and be used for?

- **Packet route** doesn’t follow the expected path
- Notification of flow **path changes**
- **Latency spikes**
- When and where packets **microbursts** happen
- Location of **congestion** and packet **drops**
- Identify **elephant flows**
- **Self-healing** networks

- **Monitor SLA**
- **Debug flow performance issues**
- **Automation of network management**

**Latency-sensitive applications**

**Bandwidth-hungry applications**
Example: INT-based High Precision Congestion Control

HPCC: use INT as precise feedback

- 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 configuration mode

Testing INT on own UDP flow, tagging with timestamps to measure its IPDV, loss,…

End user INT enabled - source

End user INT enabled - sink

LAN

GÉANT

NREN 1

NREN 2

LAN

Edgecore Wedge100BF-32X
Tofino (Barefoot/INTEL)

Arista 7170-32c

BMv2 virtual P4 software switch

P4 on DPDK implementation

FPGA

www.geant.org
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.

- Packet inter-arrival time and losses between Source nodes (PSNC, FBK) and Sink node (CESNET)
- UDP packets generated with constant rate 1k to 300k pps
5 minutes of the INT monitored flow from PSNC to CESNET

Packet rate (1.8K pack/s)
Flow of about 1 Mbps

Reordering (and loss) (none)

Packet inter-arrival time (555 μs average - 1 packet is ~5 μs – 1 Gb LAN)
10 ms of inter-arrival packet time, 1 UDP flow of 260 K pps
(~3.3 µs average) FBK to CESNET (1 packet is ~5 µs – 1 Gb LAN)
INT Platforms: Lessons Learned

A solution is available works for many use cases

- **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 (development ongoing)**:
  - Promising performance up to ~10G
  - Needs careful selection of NICs
... and "Big Data" management to become the "new normal"
Collecting INT data

- Initially stored plain INT header content
- Currently INT data is pre-processed by INT exporter, computing time differences

INT report (InfluxDB line protocol)
```
int.telemetry.srcip=10.0.0.1,dstip=10.0.0.2,srcport=1111,dstport=80
origdst=157656960000000000000000000,dtstp=15765696000000010000000,seq=625
```

INT report contains packet information:
- source and destination IP addresses
- source and destination ports
- source and destination timestamps
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 packet is monitored:

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

Means OK for "our" INT testbed

- Performance has also to scale 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)

Control Plane level

Data Plane level

INT exporting → INT collection → INT preprocessing → INT storing → INT visualisation

INT analytics

Control Plane or Application systems

Data Plane level

P4 language
eXpress Data Path
Data Plane Development Kit
extended Berkeley Packet Filter
Remote Direct Memory Access

Data Plane level

More generality and flexibility, bigger server clusters

Many limitations, hardware special requirements

Data filtering
Key events finding
Data aggregation
Time resolution

Kafka Streaming cluster
ElasticSearch cluster
InfluxDB Enterprise cluster
Spark cluster

Data aggregation

Time resolution

Remote Direct Memory Access
Summary
Summary

• **INT** (and Data Plane Programming) (using P4) offers a powerful magnifying glass for monitoring, debugging and to provide information to control plane, in real time

• However is not business-as-usual, requires specific expertise and may require specific HW

• Strict time synchronization between nodes is important

• Using an INT/P4 tool may require handling of large amount of "raw" data, also to be used for analytics and more. It implies the development of further insight, knowledge and specific tools and equipment to scale.
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)
• Evaluate an implementation on DPDK
• **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
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


  For INT use cases, see section D, page 22
References

INT public web page: https://wiki.geant.org/display/NETDEV/INT

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.
Thank you

Any questions?

gn4-3-wp6-t1-dpp@lists.geant.org
int-discuss@lists.geant.org

www.geant.org