In-network Control for Flow Steering

A. Dimoglis^1, **F. Alhamed**^{2}, **A. Dalgkitsis**^{1}, **A. Sgambelluri**^{2}, **F. Paolucci**^{3}, **C. Papagianni**^{1}, **P. Grosso**^{1}

1: Informatics Institute, University of Amsterdam, Amsterdam, Netherlands; 2: Scuola Superiore Sant'Anna, Pisa, Italy; 3: CNIT, Pisa, Italy

E-mail: {a.dimoglis, a.dalgkitsis, c.papagianni, p.grosso}@uva.nl, {faris.alhamed, andrea.sgambelluri}@santannapisa.it, francesco.paolucci@cnit.it

ABSTRACT

In recent years monitoring networks with high level of accuracy is becoming crucial to meet performance and security requirements of data flows. Concurrently, automated communication service assurance calls for increased network visibility and faster control loops. To this end, we demonstrate a new proposition that employs in-network control and monitoring of traffic flows using In-band Network Telemetry. Our solution enables rapid network reaction times through decisions taken at the data plane.

Keywords: Network Telemetry, Dataplane Programmability, Congestion Control, Trusted Communication

1. Innovation

As network technologies evolve, operators increasingly require ubiquitous network intelligence to support comprehensive network automation, across different timescales, in order to cope with the scale of operations, and meet stringent Service Level Agreements (SLAs). This in turn, necessitates an equally pervasive monitoring system. According to the IETF, Network telemetry is a technology for gaining network insight and facilitating efficient and automated network management.

With the development of programmable data planes in-band Network Telemetry (INT) emerged, providing significant advantages, such as flexible programming, real-time and in-depth network visibility. This is being accomplished by inserting network state information directly into the data packet headers. Thus, the network measurement process is now being managed by the data plane without requiring intervention by the control pane. However, the P4 data plane framework for INT [1] and other existing frameworks yield a substantial transmission overhead, which grows linearly with the number of hops, as well as with the number of telemetry variables. Per-flow Aggregation (PFA) entails a lightweight form of INT, where the telemetry values are spread across the packets of a single flow, instead of compounding all the telemetry values in a single packet [2].

Motivated by the need for faster control loops for network automation, we demonstrate a new proposition that entails in-network control and monitoring of traffic flows using INT. Our solution enables rapid network reaction times through decisions taken at the data plane. Specifically, we show flexible and dynamic flow steering though the network, enabled by (i) ultra-fast identification of changes in the network state, in terms of detection of incipient congestion or degradation in the trustworthiness of the flow path, and (ii) exchanging control messages through the in-network communication channel, allowing the network to react immediately and steer the traffic flows via alternative (trust- and QoS-equivalent) routes, with no CPU involvement. The approach aims to significantly reduce reaction times by the infrastructure elements and improve the accuracy of measurements.

2. Demonstrator Description and Operation

Fig. 1 provides an overview of the network topology and the loosely coupled set of components that constitute the proposed framework. The main parts of our proposed solution can be summarized in the following points:

Pervasive Telemetry. It allows network elements to report metrics with a per-packet rate. We utilize the INT approach on a fully P4-programmable data-plane. The metadata are appended to the traffic, conveying crucial service metrics to a collection point [3].

Telemetry Collector. The telemetry data is sent to the Telemetry Collector, a P4-programmable switch that performs in-network aggregation at wire speed. The Telemetry Collector is responsible for the collection, aggregation, filtering, and correlation of the data. In essence it facilitates:

- the real-time identification of bottlenecks or unsolicited changes in the trust level of the flow path;

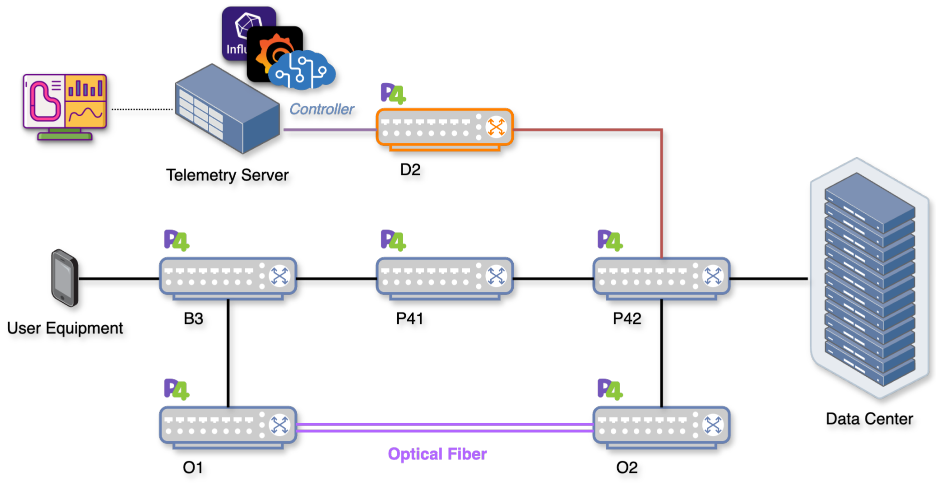
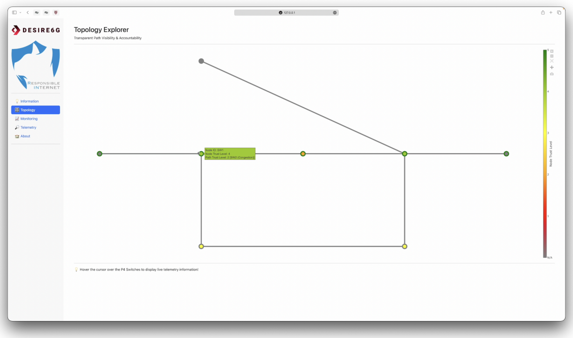


Figure 1: Experimental demonstration topology.



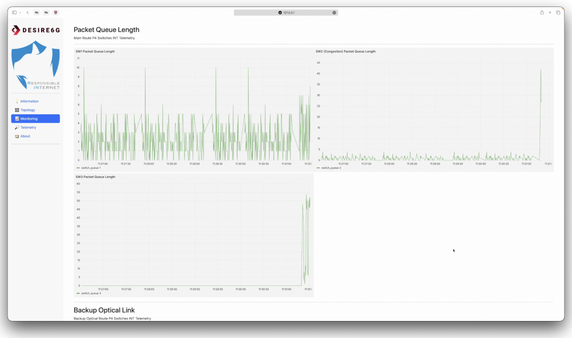


Figure 2: Dashboard: Topology explorer.

Figure 3: Dashboard: Telemetry monitor.

- real-time or near real-time control loops, utilizing predefined control policies installed in the switch, or by feeding information to the controller, which dynamically makes decisions for near real-time closed loop operations. In this demo, we show case the former.

Telemetry Server. The telemetry information are being forwarded from the collector switch to a telemetry server embedded into layer 2 data packets. Then the data are extracted and stored into a database.

Dashboard. The gathered telemetry information are displayed through a graphical user interface called (*Dashboard*) that provides:

- the *Topology exploration* section 2, which is a topology graph that indicates in real-time the state of the user data flows, and all elements in the network via aggregated information;
- the *Monitoring* section 3, that displays the collected telemetry in real-time with statistical analysis tools;
- an *Information* section with further instructions and assistance on how to operate the software.

2.1 Evaluation metrics

We are collecting different types of performance, security and routing metrics to showcase our demo in action. The information is being embedded into the data packets either using PFA INT or traditional INT (embedding metadata on each hop) based on the type of metric. These include:

1) Path Tracing: The path is traced using unique identifiers for every switch along the path. This approach ensures that the user is aware of the path that is being used at any given time. The information is collected using Probabilistic PFA, as explained in [2].

2) Trust: The trustworthiness of a path is calculated as the minimum trust of the nodes participating in the path. Each node is assigned a trust level based on several device-specific metrics, such as geo-location, running firmware, operator, interactions with other nodes, and similar criteria [4], [5].

3) Congestion: The congestion level is being evaluated based on device-specific metadata which are available by the switches. Those are the queue length and the queue delay. The former one refers to the length of the buffer that temporarily stores the packets into the switch. The latter is the delay that it takes for a single packet to go through that buffer. The congestion information is being gathered by using traditional INT. This way, no reconstruction of the telemetry data is needed, so that the congestion of the path is identified as soon as possible.

2.2 Packet Header Structure

The structure of the *Telemetry Headers* is depicted in Fig. 4. Telemetry data such as the *Path Tracing* and *Trust* related information is embedded to the packet as TCP Options, while an additional *Congestion* header is added after in the following.

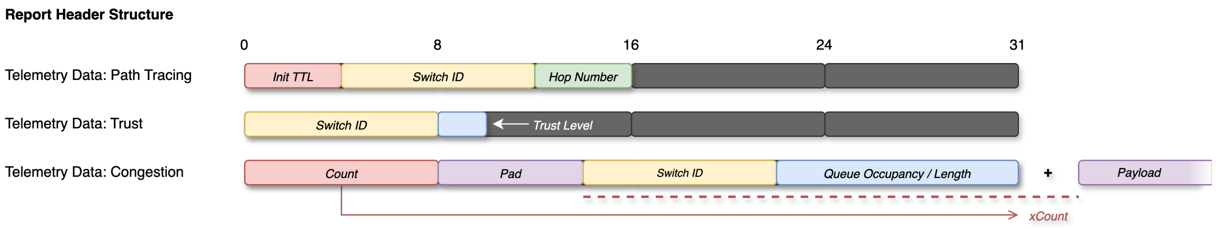


Figure 4: Telemetry and Trigger header structures overview.

Similarly, the header structure of the INT-triggered control message, consists of the target *Switch ID*, the *Port ID*, along with flow identification metadata, for redirecting the flow to the desired path.

2.3 ARNO Infrastructure

The experimental environment used for the demonstration is called the ARNO Testbed and it is co-hosted by CNIT and Scuola Superiore Sant'Anna in Pisa, Italy. The testbed includes several x86 servers that run network functions, service agents, applications, and emulated programmable network switches. A commercial packet router and a pair of commercial OTN optical transport ROADMs, encompassing muxponder cards at 100G with coherent DSP and Ethernet tributaries at 10Gb/s, are deployed to exploit tight scheduling effects and multi-layer communication.

The probing traffic is generated with a Spirent N4U traffic generator/analyzer, with a pair of Ethernet interfaces at 10GB/s, injecting and receiving traffic with specific characteristics.

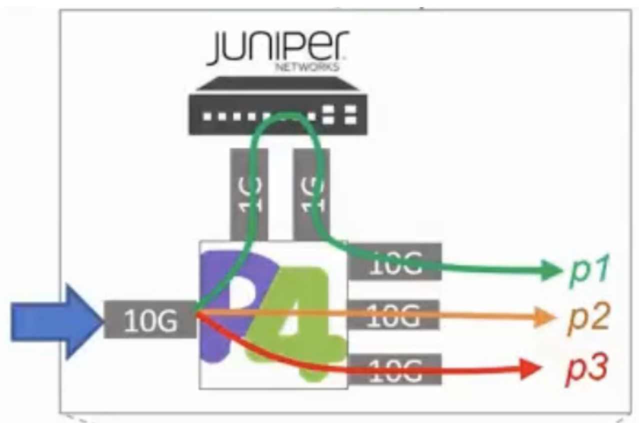


Figure 5: Usage of the packet routing scheduling to realize traffic congestion.

The programmable switches are running the Nikss software, providing an efficient implementation of fully configurable P4 switches, able to transmit the packets at wire speed. The switches run on general-purpose CPUs. Therefore, each switch operates on a dedicated server to achieve maximum efficiency.

3. Demonstration Scenarios

We assume a client-server application scenario where the server is sending TCP traffic to the client through the programmable data-plane. As the data packets are traversing through the network, the switches are embedding telemetry data into the packet headers, employing either traditional INT or lightweight INT with PFA, depending on the measured metric. Once the traffic flow packets reach the last node in the path, the telemetry data are being extracted and sent to the telemetry collector, while traffic is forwarded as expected to the end-user.

We use a P4-programmable switch as telemetry collector, depicted in 1, to facilitate ultra-fast identification of bottlenecks or unsolicited changes in the trust level of the flow path. The telemetry collector retrieves and processes telemetry data for each traffic flow, such as the buffer size for each switch in the flow path, the trust level of the path along with path tracing info. The collector uses this information to:

1) Trigger the network to take action in case of service degradation, and
2) Inform the telemetry collector about the state of the network, performing necessary computations and processing of the data at wire-speed.

The real-time QoS and trust properties of the active path are depicted in the Dashboard, as retrieved by the proposed framework, via the telemetry server. These include the real-time representation of the achieved QoS, color-coded as green, yellow, or red, depending in the level of congestion, and the trust level for all switches in the path indicated by color-coded nodes.

Scenario I: QoS Degradation. In the first scenario, we introduce background traffic in the packet router that will eventually lead to congestion in the data path. Specifically, we are causing congestion by sending multiple flows of 130 Mbps. We can observe via the Dashboard that the queue length increases in the switch P41 1 - that indicates a bottleneck, and the active path color is changing in the Topology Explorer graph 2. Once congestion is detected by the collector using predefined thresholds for metrics such as queue size or delay as an indicator of anticipated SLA violation, the telemetry collector will request network service adaptation. This is achieved through sending the INT-triggered control message with *Steering Flag ON* in the reverse path, prompting the use of the optical by-pass switches O1 and O2 1, which is the pre-defined backup path that adheres to the QoS and Trust requirements of the flow. The change of the path is then visible through the Dashboard, reflecting also the real-time QoS properties.

Scenario II: Trust Degradation (Interactive). In the second scenario, the attendee selects a switch along the active path whose trust level is reduced through the Command Line Interface, emulating the adaptation in the trustworthiness of the node caused e.g., by a change in its firmware, etc. We can observe through the Topology section of the Dashboard, that the trust level indicator color changes for the particular switch according to the scale 2. Once the degradation in the trust level of the data path is detected, the telemetry collector will request network service adaptation, in a similar fashion as described or the previous scenario. As a result, the traffic is steered to the optical backup path that adheres to the trust requirements of the flow. All events will be visible through the Dashboard, reflecting the trustworthiness of the new active path.

4. Conclusion & Future Work

We demonstrate communication service assurance by employing ultra-fast control loops. The use of INT allows the early detection of bottlenecks or unsolicited changes in the trust level of data flow path at the telemetry collector, a P4-programmable switch. Using predefined control policies installed in the switch, a specifically designed control message is sent to the respective network nodes, leading to immediate actions such as traffic steering, with no CPU involvement.

As future work, we are planning to integrate machine learning in order to predict congestion based on *a priori* patterns and implement a mechanism to verify the authenticity of the received metrics.

Acknowledgment

This work was supported by the European research project DESIRE6G (101096466) and the Dutch Research Council NWO project CATRIN (NWA.1215.18.003).

REFERENCES

[1] T. P. A. W. Group. In-band network telemetry dataplane specification v2.1. [Online]. Available: https://p4.org/p4-spec/docs/INT_v2_1.pdf

[2] K. Papadopoulos, P. Papadimitriou, and C. Papagianni, "Deterministic and probabilistic p4-enabled lightweight in-band network telemetry," *IEEE Transactions on Network and Service Management*, 2023.

[3] A. Sgambelluri, A. Pacini, L. Valcarenghi, A. Dimoglis, C. Papagianni, F. Cugini, and F. Paolucci, "Pervasive Telemetry Solutions for 6G Systems and end-to-end Accelerated Services Monitoring," 2023.

[4] S. K. Dhurandher and V. Mehra, "Multi-path and message trust-based secure routing in ad hoc networks," in *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. IEEE, 2009, pp. 189–194.

[5] N. Torkzaban, C. Papagianni, and J. S. Baras, "Trust-aware service chain embedding," in *2019 Sixth International Conference on Software Defined Systems (SDS)*. IEEE, 2019, pp. 242–247.