CHALMERS
UNIVERSITY OF TECHNOLOGY

P4-based Telemetry Processing for Fast Soft Failure Recovery in Packet-Optical Networks

Downloaded from: https://research.chalmers.se, 2025-01-14 12:06 UTC

Citation for the original published paper (version of record):
Cugini, F., Natalino Da Silva, C., Scano, D. et al (2023). P4-based Telemetry Processing for Fast Soft Failure Recovery in Packet-Optical Networks. Optical Fiber Communication Conference and Exhibition. http://dx.doi.org/10.23919/OFC49934.2023.10117408

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

P4-based Telemetry Processing for Fast Soft Failure Recovery in Packet-Optical Networks

Filippo Cugini,[1,*] Carlos Natalino,[2] Davide Scano,[3] Francesco Paolucci,[1] and Paolo Monti[2]

[1] *CNIT, Pisa, Italy*
[2] *Electrical Engineering Department, Chalmers University of Technology, Gothenburg, Sweden*
[3] *Scuola Superiore Sant'Anna, Pisa, Italy*

[*] *filippo.cugini@cnit.it*

Abstract: A novel framework for in-network P4 processing of distributed multi-layer telemetry data is presented, enabling effective soft failure detection and recovery strategies enforced in just a few microseconds. © 2023 The Author(s)

1. Introduction

Telemetry is an attractive technology to provide comprehensive and accurate awareness of optical network performance, potentially leading to remarkable traffic engineering optimization or accurate failure detection [1]. However, current telemetry systems typically suffer from substantial scalability issues due to the large amount of data processed at the centralized telemetry collector [2]. To overcome these limitations, distributed or peer-to-peer telemetry schemes have been introduced. For example, the transmitter and receiver of an optical connection may directly exchange telemetry data for performing selected operations (e.g., identification of the most suitable operational mode) without involving the centralized telemetry collector or the Software-Defined Networking (SDN) Controller [3]. Furthermore, the work in [4] showed that hardware offloading improves scalability by accelerating the telemetry data processing at the centralized collector. The work in [5] proposed exchanging telemetry for fast remote detection of failures between packet-optical nodes directly connected via multiple dedicated optical connections. However, the need to have direct connections severely limits the applicability of this solution to a few practical deployments. So far, no distributed telemetry system has been proposed for network scenarios with non-directly connected packet-optical nodes.

In this work, we propose and experimentally implement a framework for in-network P4 processing telemetry data generated by remote, not necessarily directly connected packet-optical nodes. Telemetry includes both optical Quality of Transmission (QoT) and packet-layer Quality of Service (QoS), thus providing a comprehensive view of available resources and outage probability due to impairments. We also present and validate a strategy that, upon detecting a potential QoT degradation, computes and enforces recovery routes directly at the P4 transmitting node. Results on simulations parameterized by a network testbed demonstrate the selection of the best alternative path and the effectiveness of the proposed framework at limiting packet loss due to soft failures.

2. Distributed telemetry system with P4 processing of telemetry data

The work focuses on packet-optical nodes equipped with P4 programmable packet switching technologies and coherent pluggable modules supporting high-speed optical transmission. These nodes can potentially provide significant cost savings, particularly in metro networks, due to the removal of standalone transponders and O/E/O conversions. Furthermore, they enable tight integration between the optical and the packet layers and open the way to implementing innovative in-network processing functions.

Fig. 1 shows a network of four packet-optical nodes interconnected through optical line systems. The connectivity between N1 and N2 is served by two unidirectional lightpaths, i.e., L_{12} and L_{21}. Alternative routes between N1 and N2 are also available, passing through N3 or N4. Each node knows about the QoT of the lightpaths terminated at its coherent receivers (e.g., received OSNR, pre-FEC BER, RX power). In addition, each node has access to the packet level QoS parameters (e.g., in/out packet/bit rates, queue occupation). For example, N2 monitors QoT and QoS of the optical connections originated at N1, N3 and N4, e.g., N2 can immediately detect a (soft) failure affecting L_{12}.

On the one hand, with traditional SDN-based mechanisms, N2 has no direct knowledge of the QoT status of the lightpath in the reverse direction (L_{21}). Upon a soft failure occurrence on L_{21} (e.g., amplifier malfunctioning, aging), N2 must be rapidly notified by N1. There are effective mechanisms for hard failures, e.g., routing protocol notifications and Bidirectional Forwarding Detection (BFD). On the other hand, since soft failures imply higher pre-FEC BER with sporadic post-FEC problems, these mechanisms do not operate appropriately since they are based only on up/down status information. Alternative notification methods through a centralized SDN system might not be scalable and sufficiently fast to avoid data losses. Furthermore, even if N2 correctly detects the soft

failure, it has no reliable means to quickly identify alternative directions with good end-to-end QoT and QoS. Indeed, N2 has no visibility on the performance of non-directly connected links (i.e., N3-N1 and N4-N1), which, in turn, might also suffer from soft failures and/or might not have sufficient available bandwidth. Also, in this case, leveraging on SDN intervention might not provide timely mitigation.

The paper introduces a framework that relies on the direct, in-network exchange of Multi-Layer Network Telemetry (MLNT) reports among packet-optical nodes to address these shortcomings. MLNT reports provide detailed real-time performance (i.e., QoT and QoS) of those links that are of interest for a specific node. For example, by retrieving telemetry data from N1, N3, and N4 (Fig. 1), N2 learns about the status of the outgoing directions on the links that are (in this case) up 1-hop away. N2 can then decide on the most suitable end-to-end B-A connection (i.e., direct L_{21} as well as either lightpaths L_{23}-L_{31} or L_{24}-L_{41} crossing N3 and N4, respectively). The MLNT reports are processed in-network by the P4 ASIC, operating at wire speed. Thanks to its stateful capabilities, P4 operations can then be performed on the retrieved MLNT data. In the proposed framework, the SDN Controller pre-computes the alternative route(s) to be enforced upon a QoT threshold violation for each active lightpath in the network. This is done by considering the actual QoT and QoS performance of each alternative route to make sure their QoT threshold is met, and QoS is appropriate. Then, the SDN Controller compiles the decisions in the form of custom P4 rules to be deployed at the P4 nodes. In the end, the algorithm enforces (e.g., through P4Runtime or Thrift) the flow rules and thresholds to be considered for real-time operations.

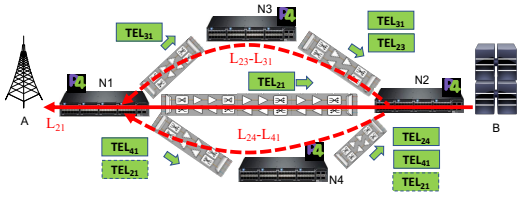

Fig. 1: Reference network scenario

if $Q_{L_{21}} \geq T_{L_{21}}$ **then**
  $do\_nothing()$
else if $Q_{L_{23}} \geq T_{L_{23}} \wedge Q_{L_{31}} \geq T_{L_{31}} \wedge max(B_{L_{23}}, B_{L_{31}}) \geq max(B_{L_{24}}, B_{L_{41}})$ **then**
  $use\_route(L_{23}, L_{31})$
else if $Q_{L_{24}} \geq T_{L_{24}} \wedge Q_{L_{41}} \geq T_{L_{41}} \wedge max(B_{L_{24}}, B_{L_{41}}) > max(B_{L_{23}}, B_{L_{31}})$ **then**
  $use\_route(L_{24}, L_{41})$
end if

Algorithm 1: P4 rules for QoT/QoS-aware re-routing

Alg. 1 illustrates the details of the P4 rules deployed by the SDN Controller at N2 to recover from a potential soft failure on L_{21}. Let Q and T be the set of current QoT measurements and thresholds for each lightpath, respectively, and B be the residual (free) bit rate at each lightpath. First, the script checks if L_{21} is above its QoT threshold, in which case no action is necessary. Otherwise, the script selects the node (N3 or N4) to be used to re-route the traffic. This is done by first checking if all links of a potential alternative path meet the QoT threshold and then selecting the alternative route with the highest residual bit rate. This way, when the QoT threshold is exceeded, the P4 ASIC can immediately steer the traffic toward the pre-defined alternative route. The proposed distributed telemetry framework provides detection of failures, including soft ones, and evaluates the performance of suitable end-to-end alternative routes in real time.

3. Implementation and Demonstration

The proposed solution was implemented in a network testbed reproducing Fig. 1. N2 is a P4 Tofino switch. N1, N3, and N4 are Mellanox/Nvidia switches equipped with Spectrum1/2 P4 ASIC, SONiC Operating system, and 100G optical pluggable modules. A soft failure is introduced by forcing in-line attenuation following the pattern of signal overlap reported in [6]. In N1, an application running in SONiC reads the QoT (i.e., RX power) and QoS (e.g., actual bit rates) values of L_{21} from the Redis database and relies on a thrift connection to write those values within dedicated flow rules. MLNT packets, generated from clones of specific management flows, match those flow rules, and their content is updated accordingly. That is, MLNT packets are filled in using in-network operations that are efficiently handled at wire speed.

Fig. 2 shows the two considered P4 implementations. MLNT headers may be sent periodically (out-of-band,

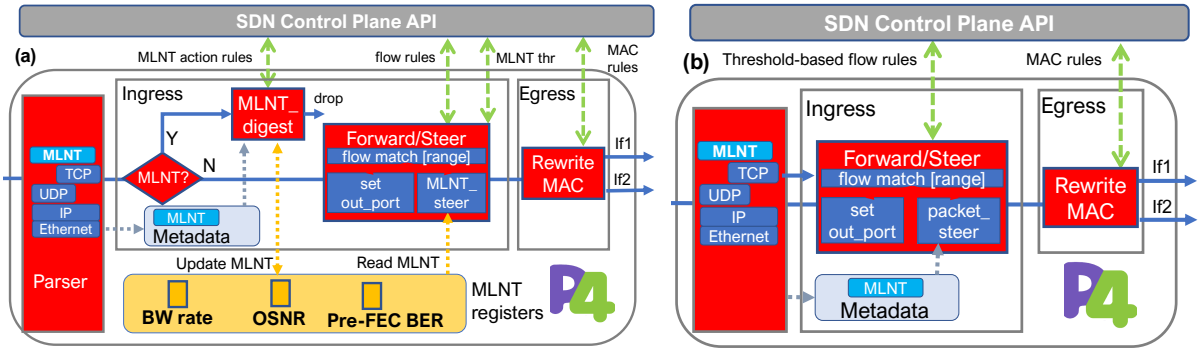

Fig. 2: P4 design of (a) out-of-band and (b) in-band MLNT-based steering pipelines.

O-MLNT) or continuously inside each traffic packet (in-band, I-MLNT). In the O-MLNT case (Fig. 2(a)), MLNT messages are processed separately, and the embedded QoT/QoS metrics are stored in dedicated P4 registers, while data traffic is steered based on the current register values (i.e., MLNT_steer action). In the I-MLNT case (Fig. 2(b)), the MLNT headers are embedded in each packet and matched as metadata against a steering flow table, performing instant forwarding relying on range-based matching (i.e., packet_steer action) without requiring any P4 register. When N2 receives the MLNT data, it stores the retrieved values in specifically configured P4 registers (O-MLNT) or processes it in a per-packet fashion (I-MLNT). When pre-defined thresholds are exceeded, L_{21} traffic flows are rerouted according to pre-installed rules (i.e., either on L_{23}-L_{31} or L_{24}-L_{41}). Results show that the time from which a MLNT packet triggering the rerouting enters N2 and the time at which the first data packet is forwarded along an alternative direction (e.g., L_{23}-L_{31}) is only around 2 µs in both I-MLNT and O-MLNT deployments. The MLNT report format is derived from INT reports [4]. Each MLNT packet occupies between 60 and 100 bytes. The overall bandwidth consumed by MLNT packets depends on the generation rate. For example, 0.08 Mb/s is obtained in the case of a MLNT packet every 10ms. Even higher rates (e.g., 1 pck/ms) would still occupy less than 0.001% of a 100 Gb/s connection. In these measurements, we limited the forwarding of MLNT data to 1-hop distances. Future studies will investigate the impact and benefit of encompassing additional/different nodes. Given the processing of data at the HW level, no performance issues are experienced, even at higher rates.

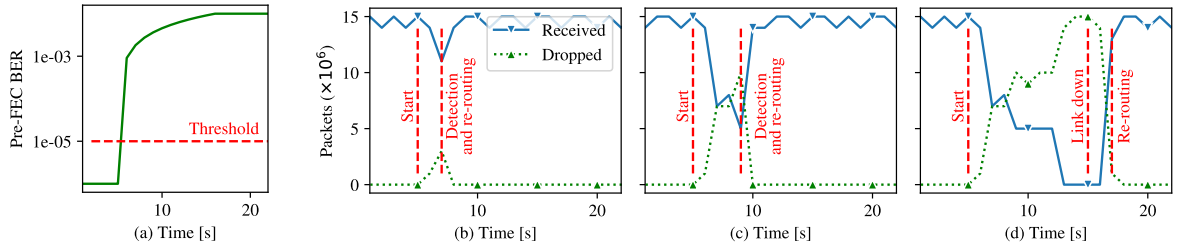

Fig. 3: Simulation results: (a) Pre-FEC BER; packets received/dropped at N1 with (b) proposed framework, (c) telemetry-based SDN; and (d) SDN-based hard failure detection.

The performance of the proposed framework was measured using an NS-3 simulation parametrized with the data recorded in the network testbed just described. Fig. 3 shows a snapshot of network operations over 25 s of simulation. Fig. 3(a) shows the evolution of pre-FEC BER over time and the threshold after which Pre-FEC BER recovery becomes challenging. The Pre-FEC BER degradation starts at 5 s, gradually increasing each second until the link is down (i.e., after 15 s). For consistency, L_{24}-L_{41} is the alternative path used in all cases. Fig. 3(b)-(d) show the number of packets received/dropped as a result of the presence of a soft failure with the proposed framework and with two benchmark solutions, i.e., telemetry-based SDN with a monitoring cycle of 5 s, and SDN-based hard failure detection. Among the three approaches, the one based on in-network MLNT performs the best with clear advantages in terms of fewer dropped packets. The figure also confirms the earlier claim that conventional SDN-based mechanisms might struggle to detect a soft failure quickly unless very frequent monitoring cycles are used, with a price to pay for increased overhead. For example, a 5 s cycle means 20 bytes/s traffic from the node to the SDN Controller per lightpath. If we were to reduce the monitoring cycle to 1 second further to approximate the results of the in-network MLNT, the overhead would increase linearly by a factor of 5. More importantly, these telemetry packets would need to be centrally processed at the SDN Controller, leading to substantial compute overhead. On the other hand, this process is distributed, thus more scalable, when using in-network telemetry.

4. Conclusions

This paper introduces a novel framework that leverages in-network P4 processing to enable distributed telemetry and fast failure recovery. Two P4 implementations of the framework show equivalent performance in terms of switching time upon detecting a (soft) failure. Simulation results parameterized using a network testbed show a substantial reduction in packet losses during the transient period of the soft failure.

Acknowledgment. This work has been partially supported by the B5G-OPEN H2020 Project (101016663), by the TeraFlow H2020 Project (101015857), and by the Chalmers' ICT-AoA seed project Auto5G.

References

1. R. Casellas *et al.*, J. Opt. Commun. Netw. **14**, C23–C37 (2022). DOI: 10.1364/JOCN.451516.
2. C. Natalino, *et al.*, in *Proc. of OFC,* (2022), p. Th3D.4.
3. F. Paolucci *et al.*, J. Opt. Commun. Netw. **14**, 606–620 (2022). DOI: 10.1364/JOCN.456666.
4. F. Alhamed *et al.*, in *Proc. of ONDM,* (2022). DOI: 10.23919/ONDM54585.2022.9782868.
5. F. Cugini *et al.*, J. Opt. Commun. Netw. **15**, A1–A10 (2023). DOI: 10.1364/JOCN.470118.
6. A. Vela *et al.*, J Light. Technol. **35**, 4595–4604 (2017). DOI: 10.1109/JLT.2017.2747223.