AI-assisted In-band Network Telemetry Framework for Fast Network Failure Detection in Optical Core Networks

Badr Mochizuki
(The Kyoto College of Graduate Studies for Informatics)

Abstract

This paper introduces the key challenges for building an AI-assisted framework for network failure detection and localization in optical core networks based on in-band network telemetry data. The main idea consists in feeding telemetry data as input for the AI module in order to achieve real-time network failure detection and localization.

1. Introduction

The aim of this paper is to discuss how to leverage In-band Network Telemetry (INT) [1] - that is network state information inserted into production flow packets as header fields - for both detecting. in real-time when a failure occurs. In particular, this paper considers exploiting the low latency of INT in order to maintain high availability of the optical core data communication networks. Three key challenges can be identified: the development of a framework for dynamic orchestration of INT metadata collection without significantly degrading the overall network performance in terms of bandwidth and quality of service; the specification of a lightweight method for real-time detection of silent failures (subtle failures that become harmful over time) from the collected INT information; the realization of fast and accurate network failure localization. The design of the INT metadata collection framework will be of particular interest because it will serve a basis for both network failure detection and localization.

In the rest of this paper, we introduce background information in section 2 and summarize the main research trends in network failure detection and localization in section 3. In Sections 4 and 5, we discuss the key challenges and its implementation strategies for AI-assisted dynamic orchestration of INT metadata collection in order to realize failure detection and localization in optical core networks. Finally, conclusions are presented in Section 6.

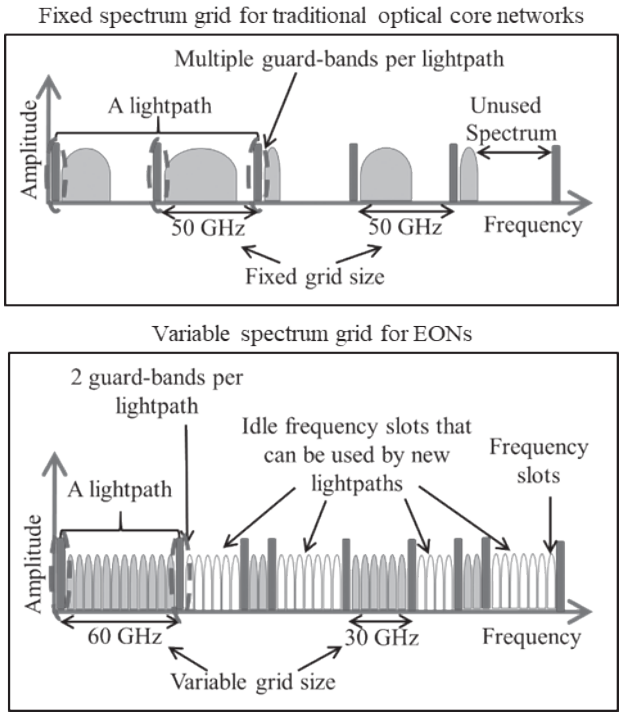

Fig. 1: Spectrum grids for traditional optical core networks and EONs.

2. Background

A. Optical core networks

In optical core networks, light is used to transmit data over optical fibers at the speed of light, hence allowing ultrafast transmission and making it ideal for core networks that span over long distances [2].

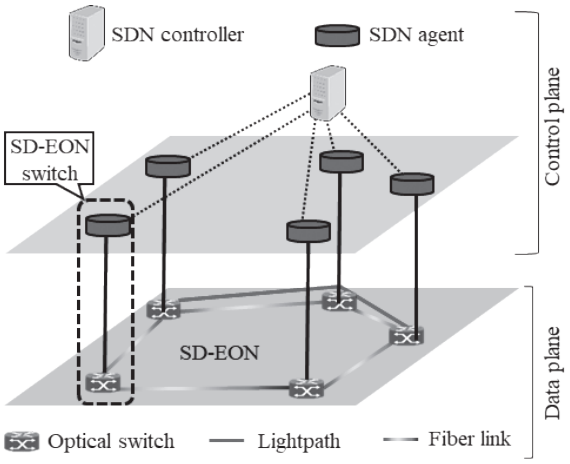

Fig. 2: Software defined EONs.

Network operators are updating their equipment on a regular basis in order to cope with the continuous increase in internet bandwidth. The network traffic is becoming increasingly heavy and bursty as the popularity of cloud computing, Internet of Things (IoT) and 5G mobile communications increases. However, adding new network equipment is costly and has physical limitations. As a promising cost-effective solution, Elastic Optical Networks (EON) have been proposed. EONs are optical circuit switching networks where first a lightpath is established, then data is transmitted all optically and finally the lightpath is released [3]. Here, lightpaths are established and released dynamically. Please note that data is transmitted through optical channels, called spectrum grids, in the optical fibers [4]. EON can support efficiently higher data bit rates than the traditional optical core networks, by dividing a wavelength into multiple frequency slots (FSs). Note that FSs have to be allocated adjacently (spectrum contiguity constraint) [5].

Figure 1 shows that the spectrum grid size is fixed in traditional optical core networks and is variable in EON. In this figure, the grid size for traditional optical core networks is fixed to 50 GHz. Therefore, if a lightpath needs 70 GHz, two grids of size 50 GHz are used, and the remaining 30 GHz cannot be used by other lightpaths. Hence the grid is not efficiently utilized in traditional optical core networks. Moreover, two guard-bands per grid is always needed even if a lightpath utilizes multiple grids. On the other hand, in EON, the grid size is variable. Here, the optical spectrum can be allocated per FS unit, and only two guard-bands are needed per lightpath. As a result, the optical spectrum can be utilized more efficiently in EON.

B. Software Defined EONs

Recently, due to the expansion of datacenters and the exponential increase in internet traffic, optical core networks can no longer be controlled with the traditional management framework. Software defined networking (SDN) has been proposed as a promising solution because network operators can program their network devices much more freely. In SDN, the data plane and the control plane are physically separated, and a centralized control plane can add, delete or modify forwarding rules on SDN switches [6]. This network architecture is illustrated in Fig. 2 for the case of software defined EON denoted as SD-EON.

In the data plane of SD-EONs, packets are sent by the optical switches on the fiber links following the predefined packet forwarding rules. On the other hand, the control plane consists of an SDN controller that translates network management policies into packet forwarding rules. These rules can be expressed as flow entries which are installed on flow tables at the SD-EON switches. The interaction between the SDN controller and SD-EON switches is performed using communication protocols such as OpenFlow [7].

C. In-band Network Telemetry

In-band network telemetry (INT) is the basis for a variety of applications such as virtual/

augmented reality, health monitoring and self-driving cars. INT is also a relatively new network monitoring framework that operates at the data plane. By using programmable network devices, it is possible to collect in real-time INT metadata such as network devices internal states (e.g., switch ID, queue occupancy) and network performance metrics (e.g., latency). This can be achieved by encapsulating information inside the packet as headers fields, which are interpreted by network devices as instructions to collect INT metadata. Finally, the collected header fields are extracted and reported to a monitoring system, which in turn analyses the INT metadata and reacts if problems are detected in the network. Consequently, INT can significantly enhance network-wide visibility, and network problems such as network failure can be detected in a timely manner [1].

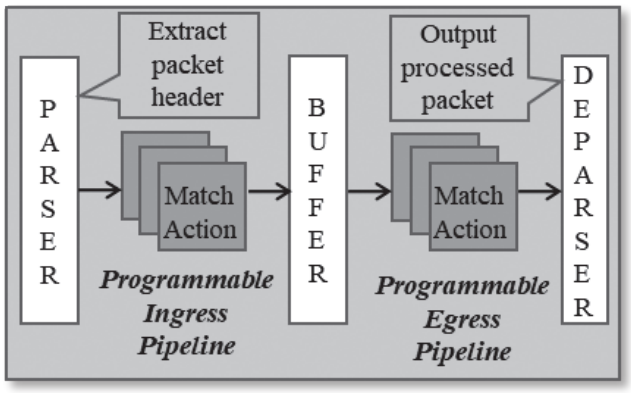

Fig. 3: P4 forwarding model.

D. Data plane programming using P4 language

In traditional SDNs, it is possible to modify the behavior of the control plane but not that of the data plane. Concretely, routing policies can be flexibly defined, but the corresponding actions cannot be flexibly executed. In order to resolve this issue, a programming language for the data plane named P4 (Programming Protocol-independent Packet Processors) has been proposed by the inventors of OpenFlow in [8]. P4 can express how incoming packets are processed by the SDN switches, without modifying the control plane. It is currently widely used in research because it is an open-source and permissively licensed language.

P4 uses the concept of match-action pipelines as illustrated in Fig. 3. Packet forwarding in SDN switches is performed by table lookups and its corresponding actions. Here, the parser, the ingress pipeline, the egress pipeline and the deparser can be modified by P4 programming. An incoming packet is first handled by the parser that will extract the packet header, and the packet payload is buffered. The extracted header is then passed to the ingress pipeline that will process the packet header according to the programmed actions and lookup keys. Furthermore, the ingress pipeline determines an egress port and a queue into which the packet is placed. Next, the packet is passed to the egress pipeline which can be programmed to process the packet in a specific manner. Finally, the packet's payload and headers are assembled and forwarded to the egress port by the deparser.

3. Related Research

There is extensive research in the literature about network failure detection and localization [9-11]. However, most of the proposed methods are implemented in the control plane and hence present a centralized solution to be implemented at the network controller. The main disadvantage of this approach is the high latency required for gathering network information at the network controller, localizing the network failure and pushing failure recovery instructions to the network devices.

Furthermore, in recent years, there have been several research work about using INT as a network monitoring framework. While most of the proposed research work has focused on proposing methods for using INT to debug network events such as congestion or traffic bursts, little research work has addressed the challenges about how to design an autonomous network monitoring framework as well as network failure detection and localization using

INT. Since network monitoring using INT requires collecting networking information regularly, data overhead is introduced in the network every time INT metadata is encapsulated into packets. Hence, it is important to consider the tradeoff between network overhead and real-time network monitoring. A promising solution that is aware of the above-mentioned tradeoff is to apply machine learning techniques to dynamically and autonomously collect INT metadata. In addition, since INT is used in the data plane, a distributed approach to network failure detection and localization is required, and the previously proposed centralized approaches cannot be applied, hence new approaches have to be considered.

In the next section, we discuss two key challenges for AI-assisted dynamic orchestration of INT metadata collection in order to realize failure detection and localization in SD-EONs.

4. Key Challenges

Key Challenge 1: How to dynamically orchestrate the collection of INT metadata?

Using INT to monitor the network state is of particular significance because it is a powerful tool that allows detecting network anomalies in real-time. INT metadata are collected and delivered to a network monitoring application. Each monitoring application is specialized in identifying a specific network anomaly (e.g. malicious network attacks, congestion or network failures) and reacts to it.

In this section, the main purpose is to orchestrate dynamically the process of INT metadata collection in order to maximize network status visibility. This is particularly challenging because of three main problems. First, if INT metadata are incorporated in all packets, a significant bandwidth overhead will be generated, which in turn will affect the network performance by introducing delays in communications. Second, the amount of INT metadata that can be embedded into packets is limited due to the maximum transmission unit (MTU). Third, each monitoring application may require different INT metadata to be collected at different rates, therefore some INT metadata may need to be collected at different rates (more or less frequently) than others.

The first and second problems can be solved by formulating an optimization problem that maximizes the number of collected INT metadata. For the third problem, a machine learning model can be designed and trained to identify which INT metadata is important to be collected based on the monitoring application's needs.

This INT collection framework will constitute the basis for the next key challenge and will have a decisive impact on performance in terms of delay and network visibility.

Key Challenge 2: How to detect and localize network failures in real-time?

Network failures can range from silent failures -where packets are dropped without notification- to complete power loss of network devices, causing network service unavailability. Silent failures can be caused by flaws in the software or errors in network configuration. While silent failures do not cause immediate network service unavailability, damage can be inflicted over time such as severe performance degradation, thrashing memory, flaky I/O, and eventually network service unavailability. In addition, silent failure detection and localization can be time consuming because no explicit notification is provided when the failure occurs.

In this section, two sub problems are considered, namely lightweight failure detection and accurate network failure localization. In the first sub-problem, the main challenge consists in devising a method that can detect early indicators of silent failures based on the collected INT metadata. This can be achieved by embedding and collecting an exhaustive amount of INT metadata in order to detect potential indicators of failures such as high queue occupancy or high packet processing time. However, the bandwidth

overhead caused by those INT metadata may degrade the network performance and consequently increase the failure detection time which in turn will delay the failure localization time. Therefore, the main difficulty consists in detecting silent failures with minimum INT metadata in order to realize real-time detection. In the second sub problem, the goal is to localize precisely the silent failure. While it is possible to identify in which path a failure occurred, it is difficult to determine the exact link or network device that is malfunctioning. Rerouting through alternative link-disjoint paths might be a temporary fix, however this could trigger unwanted congestion or failures in other parts of the network. Therefore, it is important to identify and locate precisely the failure immediately after it occurs.

5. Implementation strategies

Theme 1: Orchestration of INT metadata collection using machine learning

In this section, guidelines for solving the problems presented in key challenge 1 are discussed. The proposed framework can be realized by solving two problems: Maximizing the number of collected INT metadata and the identification of which INT metadata is important to be collected.

In the first problem, let's consider a communication network N with D network devices, a set of M monitoring applications, and a set of I *INT* metadata. These INT metadata have dependencies that are spatial – that is the INT metadata has to be collected from a specific network device - or temporal if the INT metadata has to be collected within a time limit. A spatial dependency s at device d for INT metadata i and monitoring application m is denoted as $s^i_{d,m}$. In addition, a temporal dependency t for INT metadata i and monitoring application m is denoted as t^i_m. Therefore, this problem can be formulated as a Mixed Integer Linear Programming model. The maximization of the number of collected INT metadata can be achieved by solving an optimization problem where the number of spatial and temporal dependencies are maximized. The objective function can be expressed as:

$$\text{Maximize} \sum_{m\in M}\sum_{i\in I}\sum_{d\in D} s^i_{d,m} + t^i_m$$

The above optimization problem should consider several constraints including the network flow capacity cannot be exceeded; an INT metadata should be collected at most by one network flow at a given network device.

In the second problem, it is assumed that each monitoring application may give different importance to an INT metadata depending on the monitoring application needs. In addition, this importance may change dynamically depending on the network state and time. As a solution to this problem, a learning mechanism can be used to guide the INT collection process based on the importance of an INT metadata that could be quantified in weights. Furthermore, Machine learning models such as deep learning methods could be used in the learning mechanism in order to infer those weights. The inputs for the learning mechanism would include not only the collected INT metadata (e.g. switch ID, ingress/egress port ID, packet processing time or queue length) but also specific information related the network flow or the needs of the monitoring applications that will analyze the collected INT metadata. Next, in order to keep an up-to-date network visibility, the weights representing the importance of each INT metadata must be updated periodically over time. Moreover, some weights may be updated more frequently than others depending on how often a monitoring application requires an INT metadata.

Two main advantages are provided with this framework: bandwidth overhead is reduced by maximizing the number of collected INT metadata at a time interval; and an accurate network-wide visibility can be achieved by identifying which INT metadata is important.

Finally, the collected INT metadata by the above-mentioned framework will serve as a basis for lightweight detection and accurate localization of silent failures.

Theme 2: Lightweight detection and accurate localization of silent failures

In this section, the approach for solving the problems indicated in Key Challenge 2 is presented. Here, the development of a monitoring application at each network device is proposed and consists of two modules: lightweight failure detection module and accurate failure localization module.

A) Lightweight failure detection

This module will analyze the INT metadata collected by the INT framework designed in theme 1. Specifically, INT metadata such as switch ID, ingress and egress port ID are considered. This module will create locally a "Path table" to store all active paths to all reachable destinations, where each table entry corresponds to a path to a single destination. Therefore, there may be multiple table entries for a single destination if there are multiple feasible paths to that destination. In addition, each table entry may hold switch ID, ingress and egress port ID for each intermediate network device of the specified path. Furthermore, each table entry will be stored temporarily by setting an aging time in order to limit the memory usage at each network device. Here, the table entries are updated before the aging time has elapsed based on the INT metadata provided by the INT collection framework. If the lightweight detection module does not receive the required INT metadata from the INT collection framework before the aging time has elapsed, then all the timed-out path entries are deleted and a failure is detected at those paths. Please note that the aging time will be changed dynamically based on the frequency of INT metadata collection that is regulated by the machine learning mechanism used in the framework presented in theme 1.

B) Accurate Failure localization

After a failure is detected, a timed-out path is passed to the failure localization module which is then executed in order to determine the precise location of the failure along the timed-out path. This module can localize failures through the use of backward probing in three steps. First, an INT probe packet is sent through the timed-out path. Second, at each intermediate device, the status of the egress port is checked. If the status is UP, the probe packet will be forwarded to the next intermediary device without embedding any information in the probe packet header. Otherwise, if the egress port status is DOWN, the switch ID and egress port ID are embedded in the probe packet's header. Third, the probe packet is sent through the same ingress port back to the source device. Please note that no additional information is embedded in the header when the probe packet is sent backwards to the source. Finally, the failure localization module will pass the switch ID and the port ID where the failure occurred to a failure recovery module.

6. Conclusion and Future work

The strategies mentioned in section 5 will be confirmed through practical experiments using the network simulator Mininet [12], and INT data collection procedures will be implemented in P4. Furthermore, the P4 program will be written for the V1Model architecture implemented on P4.org's bmv2 software switch, using the experimental setup provided in the P4 tutorial [13]. As a future work, a network failure recovery module will be implemented in order to complement failure detection and localization.

[References]

[1] L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, "In-band network telemetry: a survey", *Computer Networks*, vol. 186, pp. 1-24, 2021.

[2] B. C. Chatterjee, N. Sarma, P. P. Sahu, and E. Oki, "Limitations of Conventional WDM Optical Networks and Elastic Optical Networks for Possible Solutions", *Routing and Wavelength Assignment for WDM-based*

Optical Networks, Springer, Cham, vol. 410, pp. 101-115, 2017.
[3] O. Gerstel, M. Jinno, A. Lord, and S. J. B. Yoo, "Elastic Optical Networking: A New Dawn for the Optical Layer?", *IEEE Communications Magazine*, vol. 50, no. 2, pp. S12–S20, 2012.
[4] B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial", *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1776-1800, 2015.
[5] M. Recalcan, F. Musumeci, M. Tornatore, S. Bregni, and Achille Pattavina, "Benefits of Elastic Spectrum Allocation in Optical Networks with Dynamic Traffic", *IEEE Latin America Transactions*, vol. 13, no. 11, pp. 3642-3648, 2015.
[6] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, future of programmable networks", *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
[7] Open Networking Foundation, https://www.opennetworking.org
[8] P. Bosshart et al., "P4: Programming Protocol-Independent Packet Processors", *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 88–95, 2014.
[9] PC Fonseca and ES Mota, "A survey on fault management in software-defined networks", *IEEE Communications Surveys & Tutorials*, vol. 19, no 4, pp. 2284-2321, 2017.
[10] F. Musumeci, C. Rottondi, G Corani, S. Shahkarami, F, Cugini, and M. Tornatore, "A Tutorial on Machine Learning for Failure Management in Optical Networks", *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4125-4139, 2019.
[11] J. Ali, G. Lee, B. Roh, D.K. Ryu, and G. Park, "Software-Defined Networking Approaches for Link Failure Recovery: A Survey", *Sustainability*, vol. 12, no. 10, 2020.
[12] Mininet: http://mininet.org/
[13] P4 Tutorial: https://github.com/p4lang/tutorials

◆著者紹介

望月　バドル　Badr Mochizuki

京都情報大学院大学　講師
福井大学大学院工学研究科修了　工学博士
奈良先端科学技術大学院大学情報科学研究科修了　工学修士
元 CNRS 研究所（フランス）　研究員
Al Akhawayn 大学（モロッコ）　工学士