Demonstration of Packet-Optical Intent-Based Survivability Using Mininet-Optical

Rossano P. Pinto[1], **Celso H. Cesila**[1], **Kayol S. Mayer**[2], **Andrés F. Escallón-Portilla**[2],
Dalton S. Arantes[2], **Darli A. A. Mello**[2], **Christian E. Rothenberg**[1]
(1) DCA, (2) DECOM, University of Campinas, Avenida Albert Einstein 400, Campinas, 13083-852, SP, Brazil.
rossano@dca.fee.unicamp.br

Abstract: We demonstrate packet-optical intent-based networking with survivability intents in a Mininet-Optical testbed. The intent agent negotiates intents with users based on path availability and allocates end-to-end connectivity services.

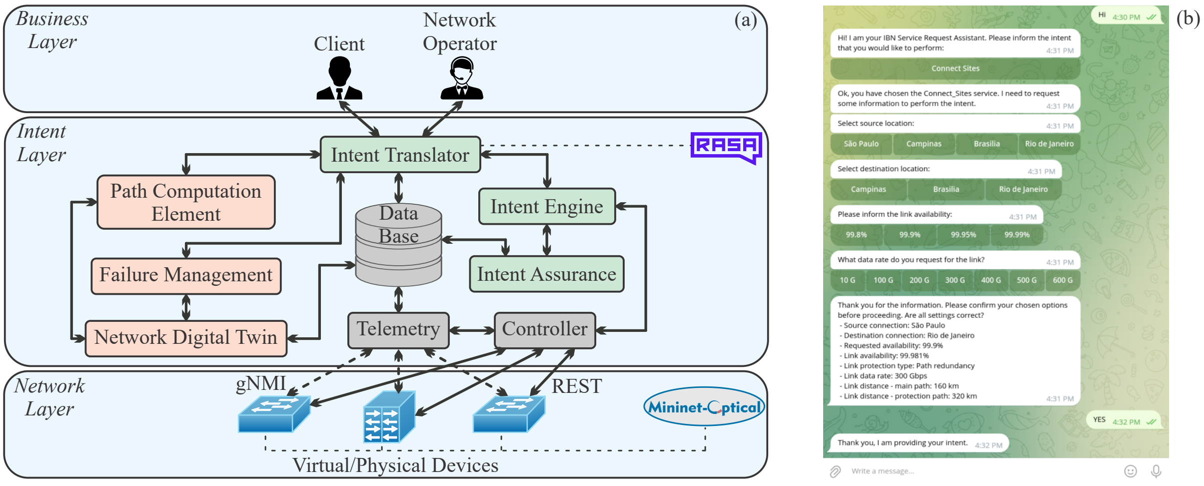
OCIS codes: (060.1155) Software-defined optical networks, (060.4257) Networks, network survivability. (060.4261) Networks, protection and restoration

1. Overview

Intent-based networking (IBN) [1, 2] is one of the most prominent technologies enabling automation in optical networks. Following software-defined networking (SDN) [3] principles, IBN is based on centralized control with a business-centric view that increases abstraction and improves automation of manual processes. IBN can be characterized as an abstraction layer on top of the network infrastructure where operators can control parameters irrespective of the environment [4]. Ideally, users should be able to express their network intents using natural language and have them seamlessly translated into network configurations [5]. IBN becomes of paramount importance in scenarios of increasing network complexity, where management tasks such as resource allocation and network survivability cannot be adequately addressed through traditional tools.

This demonstration presents network survivability as an additional intent in an IBN framework. In traditional core networks, connection survivability has been quantified using several metrics [6], being the existence or not of an additional dedicated protection path the most common differentiation mechanism. However, this strategy has several limitations, e.g., neglecting the failure susceptibility of the working and protection paths. A second common approach is to allocate connections based on their asymptotic availability. The limitation of this approach is that the interval availability, measured within a service-level agreement (SLA) duration, only approaches the asymptotic availability in SLAs lasting many years or even many decades. Therefore, a more suitable approach, in-line with business intents, is to allocate survivability resources based on the interval availability requested by the user, also considering the risk of not fulfilling the contract and potential non-compliance penalties [7, 8]. In this demonstration, an IBN-based framework abstracts the entire resource-allocation decision and implementation, translating the required user availability into a set of complex procedures that are transparent to the user.

The deployed IBN solution, shown in Fig. 1(a), is composed of three layers: Network, Intent, and Business Layers. The **Intent Layer** mediates all the processes to install, manage, and monitor the network behavior to



Hi 4:30 PM

Hi! I am your IBN Service Request Assistant. Please inform the intent that you would like to perform: 4:31 PM

Ok, you have chosen the Connect_Sites service. I need to request some information to perform the intent. 4:31 PM

Select source location: 4:31 PM

Select destination location: 4:31 PM

Please inform the link availability: 4:31 PM

What data rate do you request for the link? 4:31 PM

Thank you for the information. Please confirm your chosen options before proceeding. Are all settings correct?
- Source connection: São Paulo
- Destination connection: Rio de Janeiro
- Requested availability: 99.9%
- Link availability: 99.981%
- Link protection type: Path redundancy
- Link data rate: 300 Gbps
- Link distance - main path: 160 km
- Link distance - protection path: 320 km

4:31 PM

YES 4:32 PM

Thank you, I am providing your intent. 4:32 PM

Fig. 1. (a) IBN architecture deployed with a Network Layer, Intent Layer, and Business Layer. (b) Chatbot with RASA conversational AI for implementing intents with natural language.

satisfy the user intent. From top to down, the **Intent Translator** service interacts with the user to collect the desired intent characteristics using natural language. Intents are received based on a chatbot implemented in Telegram that uses conversational AI based on RASA [9] . In this demonstration, the intent is composed of source location, destination location, link availability, and data rate.

With all features collected, the **Path Computation Element** (PCE) selects all feasible lightpaths between source and destination locations. The PCE is implemented as an application of a **Network Digital Twin** (NDT) [10] that mirrors the network status and provide suitable QoT estimation for unestablished connections. The feasible connections are returned to the Intent Translator, which forwards them to the **Failure Management** module, which keeps a record of all failures that occurred in the network and uses the previous history to individually adjust the failure and repair rates of all network elements (i.e. fibers, reconfigurable add-drop multiplexers, transponders). The Failure Management module interacts with the NDT to store an updated list of active network elements. The Failure Management module selects the working and (eventual) protection paths that satisfy the requested connection availability, considering a minimum risk of not fulfilling the contract (e.g., availability of 99.99% at risk lower than 5%). The solution returned by the Failure Management module is sent back to the client for confirmation. Once the proposed solution is confirmed, it is forwarded to the **Intent Engine**, which passes all the necessary settings to the **Controller** that executes the appropriate commands on the physical devices. Finally, the client/operator is informed about the intent installation success or failure.

In case of success, the **Intent Assurance Engine** (IAE) is configured to monitor the intent, ensuring its validity and fulfillment in real time. The IAE uses streaming telemetry to collect operational data from the network elements, including reconfigurable add-drop multiplexers (ROADMs), terminals, and packet switches. In the event of a working lightpath failure, the Controller switches the signal to an alternate path, pre-computed during the setup phase. In general, protection switching can be implemented in either the P4-packet switch (DC1, DC2) or in a ROADM (R1-R4). This demonstration uses optical layer switching implemented in ROADMs. The packet switch pipeline is programmed using the P4 language as part of a network slice provisioning. P4 switches can offer several features for network slicing [11], e.g., those provided by a DCSG (disaggregated cell site gateway) [12] in the context of 5G networks, including rate limiting and packet prioritization towards slice QoS (quality of service).

Demonstration details. The focus of this demonstration is showcasing the implementation of IBN with survivability intents. The Network Layer, shown in Fig. 1(a), consists of an experimental network setup based on the Mininet-Optical emulator [13]. As depicted in Fig. 2, the experimental testbed includes two packet switches represented by BMv2 (behavioral model version 2) [14] P4 software switches (DC1, DC2). Hosts H1 and H3 are located on DC1, and hosts H2 and H4 are on DC2. Two transponders (T1 and T2) interface the packet and the optical networks, with 0-dBm transmitter power on the line side. The optical network consists of four ROADMs (R1, R2, R3, R4) with broadcast and select (B&S) architecture. Optical links R1-R2, R1-R4, and R2-R4 are 160 km long, and R1-R3 and R3-R4 are 240 km long, all composed of cascaded 80-km fiber spans. In-line amplifiers operate with 17.6-dB gain. Pre-amplifier and booster gains are set to 17.0 dB and 17.6 dB, respectively. A single REST server is instantiated to allow ROADMs and terminals configuration and management by the Controller. We also instantiate a NETCONF server per ROADM to monitor input and output powers. A gNMI server per ROADM provides link state monitoring. In coordination with the main Controller, a fast-response agent/controller enables network monitoring and triggers ROADM reconfiguration in case of network failure.

The demonstration features the IBN interaction via the chatbot application (Fig. 1(b)) and the real-time processing of the desired intent. Traffic is generated between hosts H1 and H2 over optical link R1-R4, as shown in Fig. 2(a). The demonstration presents real-time lightpath monitoring and failure detection in two scenarios: (a) without protection and (b) dedicated protection over R1-R2-R4. The choice of protecting or not the connection is taken by the Failure Management module based on the availability target selected using the chatbot. After the connection is established, a hard failure is introduced in the main path (R1-R4), triggering a real-time response in case of protection (Fig. 2(b)). Fig. 2(c) shows CLI captions indicating a fast-response local agent/controller (red), a Mininet-Optical CLI (black), and the traffic generator producer/consumer (orange). Fig. 2(d) exhibits the flow throughput for the working (yellow) and protection paths (green) in case of a working path failure.

2. Innovation

The demonstration showcases several innovative features. i) We present an experimental validation testbed of a novel method for network configuration and monitoring leveraging state-of-the-art IBN on a packet-optical emulation platform. ii) We demonstrate how real-time monitoring and intent assurance mechanisms can be used to offer protection in a closed-loop automated approach. iii) We use a natural language Intent Translator and an Intent Engine to express, negotiate, deploy, and monitor intents. iv) A network digital twin and a failure management module store records of failure events and quantify the risk associated with not fulfilling SLAs. v) State-of-the-art streaming telemetry is employed to detect failures and trigger network reconfiguration.

```
*** r4.check_switch_rule: removing switch rule (4, 1): 1
*** r4.remove_switch_rule: [4, 1]: 1
*** t1.turn_on <ch1:191.35THz> on port 2
*** t2 receiving <ch1:191.35THz> at port 2: Success! gOSNR: 22.908080 dB OSNR: 24.987137 dB
*** t2.turn_on <ch1:191.35THz> on port 2
*** t1 receiving <ch1:191.35THz> at port 2: Success! gOSNR: 22.908080 dB OSNR: 24.987137 dB
*** t1.turn_on <ch1:191.35THz> on port 2
*** t2 receiving <ch1:191.35THz> at port 2: Success! gOSNR: 22.908080 dB OSNR: 24.987137 dB
*** t2.turn_on <ch1:191.35THz> on port 2
*** t1 receiving <ch1:191.35THz> at port 2: Success! gOSNR: 22.908080 dB OSNR: 24.987137 dB
64 bytes from 10.1.1.2: icmp_seq=457 ttl=64 time=3.46 ms
```

```
WORKING PATH DOWN - switching to protection path
    DC1 - t1 -- r1 -- r2 -- r4 -- t2 - DC2
WORKING PATH UP - switching back
    DC1 - t1 -- r1 -- r4 -- t2 - DC2
```

```
------------------------------------------------------------
Client connecting to 10.1.1.2, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.1.1.1 port 40118 connected with 10.1.1.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  19.4 MBytes  16.2 Mbits/sec
```

```
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.1.1.2 port 5001 connected with 10.1.1.1 port 40118
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.2 sec  19.4 MBytes  16.0 Mbits/sec
```

Fig. 2. Experimental setup for packet-optical IBN with survivability intents. (a) Network Layer emulated in Mininet-Optical comprising two P4 switches (DC1 and DC2) with two hosts each (H1, H2, H3 and H4), four ROADMs (R1, R2, R3, and R4), and two transponders (T1 and T2). (b) Protection switching in case of a failure in the R1-R4 link. (c) Command line captions indicating: a fast-response local agent/controller (red); Mininet-Optical CLI (black); traffic generator - producer/consumer (orange). (d) Real-time working and backup path throughput upon protection switching.

3. OFC Relevance

Network automation and IBN are active discussion topics within the wireless and data networking communities and are gaining traction in optical networking. Although initial works on IBN applied to optical networks have already been proposed, only a few implementation attempts have been published to demonstrate its main concepts. The present demonstration showcases an IBN solution including most of its features, including AI-based natural language processing and the translation of business intents related to SLAs. Furthermore, our demonstration is implemented using the Mininet-Optical framework, which is a recent emulation tool available to the optical network community. Finally, we expect that the demo will be of interest to academia and industry, including network operators and equipment manufacturers concerned with exploring the possibilities of advanced customer-operator automated interaction.

Acknowledgement. This work is supported by Padtec S/A.

References

[1] A. Leivadeas et al., "A survey on intent based networking," IEEE Commun. Surv. Tutor., pp. 1–32, (2022).
[2] L. Velasco et al., "Intent-based networking and its application to . . . ," J. Opt. Commun. Netw. **14**, pp. A11–A22, (2022).
[3] D. Kreutz et al., "Software-defined networking: A comprehensive survey," Proc. IEEE, **103**, pp. 14–76, (2015).
[4] E. Zeydan et al., "Recent Advances in Intent-Based Networking: A survey," Proc. of VTC2020-Spring, pp. 1–5, (2020).
[5] M. Bezahaf et al., "To all intents and purposes: Towards flexible intent expression," Proc. of NetSoft, pp. 31–37, (2021).
[6] R. B. R. Lourenço et al., "On the exponential assumption for the time-to-repair in . . . ," Proc. of ICTON, pp. 1–4, (2012).
[7] D. A. A. Mello et al., "Interval availability estimation for protected connections . . . ," Comput. Netw. **55**, 193–204, (2011).
[8] D. A. A. Mello et al., "Specification of SLA survivability requirements . . . ," Proc. of OFCC/NFOEC, paper JThB76, (2006).
[9] *Rasa*, Available: `https://github.com/RasaHQ/rasa`. [Accessed: 07-Nov-2022].
[10] K. S. Mayer et al., "Demonstration of ML-assisted soft-failure . . . ," J. Light. Technol. **40**, pp. 4514–4520, (2022).
[11] R. Casellas et al., "Enabling network slicing across disaggregated optical . . . ," Proc. of OFC, paper Tu3H.3, (2019).
[12] J. A. G. Atrio et al., "Disaggregated cell site gateway," Tech. spec., Telecom Infra Project, pp. 1–31, (2019).
[13] B. Lantz et al., "Demonstration of software-defined packet-optical network . . . ," Proc. of OFC, paper M3Z.9, (2020).
[14] *BMv2*, Available: `https://github.com/p4lang/behavioral-model`. [Accessed: 09-Nov-2022].