CHALMERS
UNIVERSITY OF TECHNOLOGY

Optimizing telemetry forwarding for distributed failure recovery in packet-optical networks

Citation for the original published paper (version of record):
Lechowicz, P., Natalino Da Silva, C., Cugini, F. et al (2025). Optimizing telemetry forwarding for distributed failure recovery in packet-optical networks. Journal of Optical Communications and Networking, 17(2): 152-162. http://dx.doi.org/10.1364/JOCN.534559

(article starts on next page)

Optimizing telemetry forwarding for distributed failure recovery in packet-optical networks

PIOTR LECHOWICZ[1,2,*], CARLOS NATALINO[1], FILIPPO CUGINI[3], FRANCESCO PAOLUCCI[3], AND PAOLO MONTI[1]

[1] *Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden.*
[2] *Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wrocław, Poland*
[3] *CNIT, Pisa, Italy.*
[*] *piotr.lechowicz@chalmers.se*

Compiled February 18, 2025

Fast network recoverability from hard and soft failures is crucial for network operators to deliver uninterrupted services. Streaming telemetry has been studied as a solution for enabling fast and accurate failure detection in optical networks. However, significant delay incurs when relying on a centralized entity (e.g., software-defined network controller) to collect, process, and act on telemetry data. Programmable switches (e.g., P4-based) allow telemetry data to be processed at line speed, enabling local on-device (distributed) decisions. These devices can be used to deploy quick and local mitigation to failures while a global solution is being computed on a longer time scale. However, designing network-wide streaming telemetry with distributed decisions remains an open challenge. In this work, we specify the joint optimization of packet-optical networks with on-device failure recovery, considering multiple aspects of the problem. The problem is modeled using linear programming and solved for multiple network realizations. The solutions can be used to program each switch in the network to detect failures and quickly recover the traffic. Results show that the proposed model decreases the required number of register entries to store telemetry data while assuring high recoverability and a minimized number of wavelengths.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

Current packet-optical networks use high-speed lightpaths in the optical layer to support transmission in the packet layer. Even short interruptions in the optical layer transmission incur significant data losses in the upper layers. If the data losses violate some service level agreement (SLA), these short interruptions can lead to significant monetary losses for network operators. Recovering from them is costly and challenging because failures have a strong random component, originate from different layers, and are hard to predict. Therefore, preparing the network to mitigate failures in a timely, cost-effective, and scalable fashion is essential.

Nowadays, networks are prepared to handle hard failures that cause total and immediate service disruption. Handling failures can be achieved through planning and provisioning of spare backup resources to be used upon detecting a hard failure. Soft failures, on the other hand, progressively degrade the quality of a service, eventually leading to its disruption. However, large traffic losses may occur between the start of a soft failure and the moment it triggers a service disruption. Traditionally, soft failures require data monitoring from multiple layers transmitted to and analyzed at a central location. Upon detecting a soft failure, the controller triggers a mitigation strategy by reconfiguring all involved devices. Centralized mitigation means that there are several round-trips between the controller and the devices involved from the beginning of the soft failure to its mitigation.

More recently, on-device processing capabilities (e.g., P4-based [1]) have enabled a new category of failure detection and mitigation schemes [2, 3]. They are distributed and multi-layer. Additionally, they can significantly reduce the time required to detect soft failures and, consequently, data losses, thanks to their ability to process telemetry data at line speed, i.e., at the same speed as the traffic flows being processed by the device. For instance, in [3], we have shown that a multi-layer, on-device, distributed detection and recovery mechanism can reduce the restoration time by several seconds in a software-defined networking (SDN)-like architecture. This results in lower traffic losses compared to a centralized mechanism for detection and recovery. However, the work in [3] focused on a simple scenario with a single connection, showcasing only the scheme's feasibility, the time incurred to detect and mitigate a failure, and its ability to reduce traffic losses.

Given that it is possible to efficiently detect and mitigate (soft) failures by leveraging programmable devices, the next logical step is to investigate how to realize this in a network-wide scenario. This can be achieved by deploying programmable devices (e.g., switches and transceivers) and defining a telemetry forwarding graph, i.e., deciding where to monitor the network and where to send the measurements. Following this intuition, devices receive telemetry data from the devices supporting a particular lightpath, store the information in register entries, and forward them to a defined set of nodes using in-band or out-of-band telemetry. Then, the transmitting devices process telemetry data received, determining if a (soft) failure affects the lightpath of interest. If a (soft) failure is detected, the device can quickly adopt a short-term mitigation while the centralized controller computes a longer-term mitigation.

In this paradigm, optimizing a network-wide scenario introduces challenges related to the presence of multiple (often conflicting) aspects. Given a traffic matrix (i.e., logical topology) comprised of the data rate requirements between node pairs, multiple objectives must be optimized concurrently. Firstly, the routing and spectrum allocation (RSA) problem determines how the logical topology will be mapped onto the physical topology by defining the lightpaths to be established while optimizing the resource usage in the optical layer. Secondly, alternative recovery paths (to be used upon failure detection) need to be selected based on the RSA solution. Thirdly, based on the solution of the two previous aspects, the probe placement needs to be computed. Probes are characterized by programmable devices that collect multi-layer telemetry data, store it in their register entries, and forward it to the appropriate node(s). Fourthly, a telemetry forwarding graph needs to be computed. The telemetry graph establishes which programmable devices will receive, store in their register entries, and forward telemetry data so that it reaches its destination. Note that the probe placement and telemetry forwarding graph need to be carefully designed to ensure the required programmable device resources are within their capacity. Finally, the RSA and the recovery paths selected will determine the recoverability of the network, i.e., the ability of the network to recover upon a given intensity of failure.

In this work, we specify the joint problem of designing a packet-optical network that takes advantage of programmable devices to enable distributed failure recovery. The problem is solved by an integer linear programming (ILP) model that computes the optimal routing, alternative recovery paths, telemetry forwarding graph, and probe placement, while optimizing the network recoverability performance. Each connection is protected by two logical alternative recovery paths, assuming best-effort data rate availability in the case of a failure. The proposed solution allows the mitigation, at line speed, of hard and soft failures affecting the optical layer. The numerical experiments investigate the tradeoffs regarding the required number of wavelengths, register entries, and recoverability from failures in the physical topology. Results show that the model can reduce the register entries required by nearly 8% while incurring only a few extra wavelengths in the most congested link.

2. LITERATURE REVIEW

The problem specified in this paper and the optimization model proposed to solve it are at the intersection of several areas of the literature. In the following, we review the literature closely related to this work.

One of the key aspects of this work is the adoption of telemetry. In-band network telemetry (INT) has been gaining attention from academia and industry as an improvement for traditional network monitoring and a step towards self-driven networks [4]. Tan *et al.* [5] survey recent achievements in network telemetry, describe various existing solutions such as INT, in situ operation administration and maintenance (IOAM), alternate marking-performance measurement (AM-PM), active network telemetry (ANT), and point out technical challenges and possible future directions. Telemetry can be applied to measure various operating network parameters, namely, available bandwidth [6], anomaly detection [7], congestion control [8], throughput, fault localization [9], queue depth [10], load balancing [11], loss, delay [12], path tracing, estimation of tail latency, health monitoring, resource accounting and planning, attack detection, and others [13]. However, the use of telemetry as an enabler for packet-optical network-wide recovery has not been studied in the literature.

A critical aspect when adopting network telemetry is the definition of the forwarding graph, i.e., what information, where, and when should be measured to decrease potential bandwidth overhead. Hohemberger *et al.* [14] study orchestration aided by telemetry aiming to avoid performance degradation of business network flows. Orchestration based on telemetry data has been further investigated in [15], where INT fields are inserted only into a portion of packets. However, defining a network-wide forwarding graph that enables distributed failure recovery has not been studied in the literature.

Another aspect when adopting telemetry is probe planning, which aims to reduce bandwidth overhead. This problem has been studied in [16–18]. However, none of these works optimize the design in terms of routing. Castro *et al.* [16] propose an ILP model and a heuristic to minimize the number of probing cycles so that the whole graph is covered, assuming that probing packets cannot exceed a specified limit (with a probing packet limit of up to 1500 Bytes and size of telemetry item varying between 2 and 20 Bytes). In [17], a simulated annealing-based meta-heuristic is proposed to optimize the placement of probing flows for monitoring network function virtualization (NFV) service chains. In [18], the amount of transmitted data is minimized through distributed network monitoring, where multiple telemetry sinks are created for a single NFV service chain.

Soft failure detection is mentioned as one of the possible telemetry applications [3, 5, 19, 20]. However, to the best of our knowledge, probe design has not been studied in conjunction with logical topology mapping to increase recoverability from hard failures. Moreover, the mentioned papers do not consider restoration planning such that bandwidth overhead and memory are reduced for storing and forwarding required telemetry data. Therefore, to fill the literature gap, we investigate the probe planning problem to maximize recoverability from soft and hard failures while minimizing the required number of P4 node register entries.

3. NETWORK ARCHITECTURE WITH ON-DEVICE PROCESSING OF TELEMETRY DATA

In this work, we consider a network equipped with packet-optical devices containing programmable switching technologies (e.g., P4), which allows for tighter integration between the packet and optical layer [3]. Optical connections are established between source and destination nodes with a given data rate and are supported by an optical (primary) lightpath.

Programmable packet switches can process telemetry data

at line speed, retrieve local monitoring data such as Quality of Transmission (QoT) or quality of service (QoS) indicators of lightpaths, and store such indicators in their memory through register entries. Nodes can make autonomous decisions based on the collected data. For instance, a node can re-route the traffic using a predefined logical recovery (backup) path without communicating with a centralized controller, thus enabling fast soft failure recovery. The decision to re-route the traffic can be taken by the source node of a connection, driven by the measured quality of supporting lightpaths included in the logical primary and recovery paths. A single lightpath quality is measured at its destination node, stored in register entries, and forwarded to its source node. If the logical path is composed of more than one lightpath (logical hop), that information is stored and forwarded by the source node of each lightpath in the logical path, accumulating the information stored in nodes closer to the connection's source node. However, the number of registers available in each programmable switch is defined by hardware, and their number is limited. Thus, minimizing the number of register entries is crucial due to its limited number and additional processing overhead when the number of register entries increases.

An illustrative example of a network topology supporting lightpaths and their storing and forwarding of telemetry data is depicted in Fig. 1. We consider a packet-optical network, where the physical (optical) network topology has nodes denoted as set V_p. Let us denote 5 logical nodes $\{v_1^l, \ldots, v_5^l\} \in V_l$, which are a subset of physical nodes $V_l \subseteq V_p$. Logical nodes are connected with lightpaths denoted as $l_{ij} = (v_i^l, v_j^l)$ where v_i^l is its source node and v_j^l is its destination node. Each lightpath traverses one or more optical fibers in the physical topology. Two logical (recovery) connections using the logical paths p_1 and p_2 are established in the network. Path p_1 is composed of lightpaths $\{l_{12}, l_{23}, l_{34}\}$, while path p_2 of $\{l_{12}, l_{23}, l_{35}\}$. The QoT of each lightpath is measured and stored in the register entry at its destination node, e.g., l_{34} QoT is stored in v_4. The source node requires QoT of all lightpaths to decide whether to re-route the traffic. The telemetry information is propagated back along the logical path and stored in each passing node (i.e., v_3^l, v_2^l, and v_1^l). Thus, the number of register entries required for a single logical path is larger in nodes closer to the source node. Minimizing the required number of stored indicators in register entries can be achieved by either decreasing the number of hops of the logical (recovery) path or by increasing the sharing of the same lightpaths for multiple connections. In the case of Fig. 1, both logical paths share two common lightpaths, i.e., l_{12} and l_{23}. The indicators regarding l_{12} and l_{23} need to be stored for both paths p_1 and p_2. However, as the indicators of common lightpaths are the same for both paths, they can be stored and transmitted only once from nodes v_3^l and v_2^l. To be more specific, an indicator for lightpath l_{12} is stored/forwarded only once between nodes v_2^l and v_1^l. Similarly, an indicator for lightpath l_{23} is stored/forwarded only once from node v_3^l to v_2^l and from node v_2^l to v_1^l. This means only one register entry instead of two is required for lightpath l_{23} in nodes v_1^l, v_2^l, and v_3^l. The sharing of register entries for common path segments can substantially lower the number of register entries required for monitoring the network. Moreover, sharing register entries has a cascading effect, i.e., the savings are higher when the sharing occurs on longer paths. In the example of Fig. 1, 13 register entries are required when considering register entry sharing, while a solution without sharing would require 18 register entries.

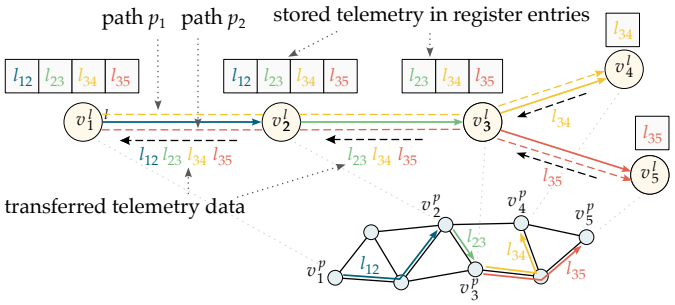


Fig. 1. An illustrative network topology with its lightpaths and the stored telemetry at programmable switches.

Fig. 2 presents the logical-physical topology mapping, telemetry forwarding, and alternative path selection example in a packet-optical network topology. The logical topology is composed of 5 nodes $\{v_1^l, \ldots, v_5^l\}$ (Fig. 2(a)). Fig. 2(b) presents the physical topology equipped with programmable switches denoted as $\{v_1^p, \ldots, v_6^p\}$, with vertical dashed lines showing the mapping between logical and physical nodes. Links in the logical topology refer to optical lightpaths $\{l_{12}, \ldots, l_{53}\}$ realized in the physical topology. Each lightpath comprises one or more physical edges, i.e., optical fibers. For simplicity, we do not present continuity constraints. However, different colors in the figure refer to the assigned wavelength. Fig. 2(c) shows the paths used by connections represented as an ordered set of physical topology nodes. For instance, lightpath l_{13} is set up in two physical links, the first one between nodes v_1^p and v_6^p, and the second one between nodes v_6^p and v_3^p.

A. In-band network telemetry

Let us assume that the l_{13} is our connection of interest for which we would like to collect telemetry data. The telemetry data can be, e.g., QoT and QoS, characterized as optical signal-to-noise ratio (OSNR) and unused/idle data rate, respectively. If the QoT drops below a predefined acceptable threshold, the connection can no longer be transmitted due to a soft failure occurrence. To detect this soft failure, one may want to predefine some other connections in the logical topology to form logical recovery paths for which QoT and QoS are monitored. After detecting a soft failure in the connection of interest, it is possible to rapidly switch to one of the recovery paths if its QoT and QoS levels allow for that.

Let p_1 denote the currently active path for connection l_{13} in Fig. 2(d). Three alternative recovery paths may be selected for l_{13}, namely, $p_2 = \{l_{12}, l_{23}\}$, $p_3 = \{l_{15}, l_{53}\}$, and $p_4 = \{l_{14}, l_{45}, l_{53}\}$. Monitoring the QoT and QoS of each alternative path requires storing that information in register entries and forwarding that information to the node of interest, i.e., the source node of p_1. For path p_4, node v_3^l must monitor the QoT (i.e., store in a register entry) of l_{53} and forward that to node v_5^l. Node v_5^l needs to monitor lightpath l_{45} and forward the information about lightpath l_{53} to node v_4^l. Thus, node v_4^l needs to use two register entries. Finally, node v_4^l monitors lightpath l_{14}, and forwards information about l_{45} and l_{53} to v_1^l, thus requiring 3 register entries. QoT data is collected at the node of interest v_1^l, and stored in its registers, requiring three entries. To summarize, monitoring the quality of path p_4 requires 9 register entries: 3 in v_1^l, 3 in v_4^l, 2 in v_5^l, and 1 in v_5^l. Therefore, it seems reasonable to balance the number of alternative recovery paths and the data that needs to be stored in

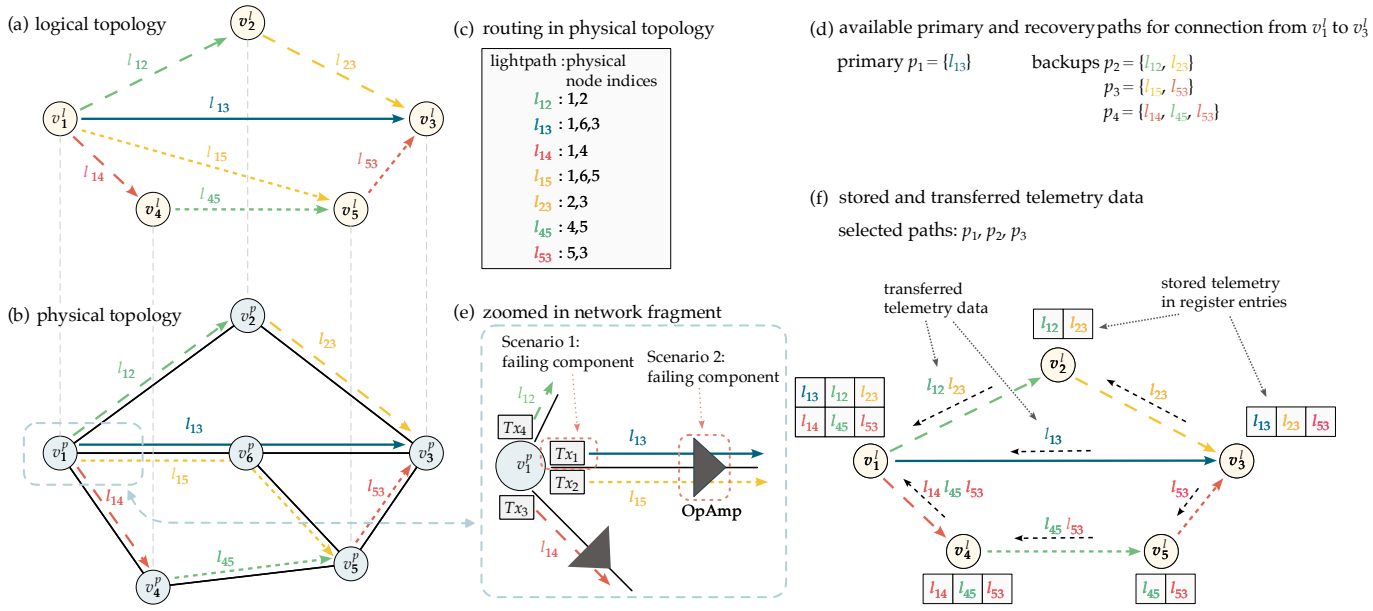


Fig. 2. Logical-physical topology mapping, telemetry forwarding, and recovery path selection example.

register entries. Assuming that the network operator may want to keep track of two alternative recovery paths for soft failure mitigation, it is reasonable to use p_2 and p_3 as they have a lower number of hops when compared to p_4 and, as a consequence, require a lower number of register entries in the nodes.

Fig. 2(e) shows two possible failure scenarios. In the first scenario, transceiver Tx_1 malfunctions in node v_1^p, affecting connection l_{13} that uses path p_1. After noticing a QoT degradation, node v_1^p may temporarily switch the traffic to path p_3 composed of links $\{l_{15}, l_{53}\}$ if sufficient QoT and QoS are met by path p_3. However, if the failing component is an optical amplifier, as in scenario 2, path p_3 is also affected as both paths share the same physical link where the malfunctioning device is located. To account for such cases, network operators may select longer, node- and link-disjoint recovery paths to ensure that they can be used to recover from connection-specific as well as link/node failures. Fig. 2(f) shows the processed telemetry data for the scenario when path p_1 is selected as a primary path and p_2 and p_4 as recovery paths. Such a solution requires a total of 16 register entries across all network nodes.

The planning of primary and recovery paths will not only define the ability of the network to recover from soft failures but also define how many register entries will be used at each node. In an operating network loaded with various connections, a given node is likely required to monitor several connections other than the ones it originates or terminates. Therefore, it is essential to carefully consider the number of register entries when designing the recovery paths.

B. On-device soft-failure detection and re-routing

In the proposed INT framework, network nodes can apply a re-routing policy after detecting soft failures without delegating that decision to a centralized network controller. To this end, pre-planned logical recovery paths are selected for each connection, and the telemetry data is forwarded to the corresponding nodes. Each network node is programmed with two dedicated rules for each primary lightpath that: *(i)* define the QoT threshold that triggers re-routing once crossed, and *(ii)* define the alternative route that should be applied when a soft failure is observed.

Algorithm 1 presents the policy deployed in node v_1^l for connection R using lightpath l_{13}, considered in Fig. 2. Based on the planning phase, two recovery paths are selected for that connection, namely path $p_2 = \{l_{12}, l_{23}\}$ and $p_4 = \{l_{14}, l_{45}, l_{53}\}$. Information about lightpaths included in the primary and two recovery paths for connection is programmed in the network node and is denoted as R_p, R_{b1}, and R_{b2}, respectively. In addition to the routing information, network nodes store QoT thresholds T_l of lightpaths comprising paths $l \in R_L : R_L = R_p \cup R_{b1} \cup R_{b2}$. For each lightpath $l \in R_L$, the node collects the required information, such as QoT measurement Q_l and residual (free) data rate B_l. Node v_1^l observes the QoT level of the primary lightpath (line 1). When the quality of a primary lightpath drops below an acceptable threshold, the network node swiftly investigates which of the recovery paths is currently the most suitable. If both recovery paths demonstrate sufficient QoT (line 2), the algorithm selects the path with the higher residual capacity for re-routing (lines 3-6). If only one of the recovery paths meets the QoT requirement, the algorithm selects and re-routes onto that path (lines 7-10). Based on such applied re-routing policies, programmable devices can immediately re-route the traffic when a failure is observed without forwarding the data and waiting for a decision to/from a centralized network controller, thus minimizing the disruption caused by the soft failure.

4. NETWORK MODEL AND PROBLEM FORMULATION

We model the packet-optical network as a directed graph $G(V_l, V_p, E)$ where V_l is a set of logical nodes communicating in the network, $V_p \supseteq V_l$ is a set of optical nodes, and E is a set of optical fibers that interconnect optical nodes. Let D denote the set of traffic requests, where each request $d \in D$ is characterized by its source $d_s \in V_l$ and destination $d_d \in V_l$ nodes, and data rate $d_r \in \mathbb{R}^+$.

Algorithm 1. Rules for soft-failure detection and re-routing of connection R deployed in node v_1^l from Fig. 1.

Data: $R_p \leftarrow \{l_{13}\}$
$R_{b1} \leftarrow \{l_{12}, l_{23}\}$
$R_{b2} \leftarrow \{l_{14}, l_{45}, l_{53}\}$

if $\exists_{l \in R_p} Q_l < T_l$ **then**
  if $\forall_{l \in R_{b1}} Q_l \geq T_l \wedge \forall_{l \in R_{b2}} Q_l \geq T_l$ **then**
    if $\min_{l \in R_{b1}} B_l \geq \min_{l \in R_{b2}} B_l$ **then**
      use_route(R_{b1})
    else
      use_route(R_{b2})
  else if $\forall_{l \in R_{b1}} Q_l$ **then**
    use_route(R_{b1})
  else if $\forall_{l \in R_{b2}} Q_l$ **then**
    use_route(R_{b2})

A. Problem Formulation

In this paper, we propose the routing assignment and probe design (RAPD) problem, which encompasses optimizing five different aspects of a packet-optical network. The first aspect is the primary path. For each traffic request $d \in D$, we need to solve the RSA problem, i.e., find a path in the optical topology. The optical (primary) path is always a transparent lightpath connecting the source and destination nodes of d, with data rate according to d_r. The second aspect is the selection of recovery paths. The solution needs to select two logical paths created for other traffic requests. The recovery paths can be composed of one or more lightpaths. The third aspect is the selection of nodes that will act as probes. Failure recovery of a connection requires the source node to monitor the network status along the primary and recovery paths. Thus, telemetry data should be collected, stored, and forwarded in each node along the paths. The fourth aspect is the computation of the telemetry forwarding graph formed by the probes. The fifth aspect is the recoverability of the network against link/node failures, characterized by the investigation of the node- and path-disjointness of the primary and recovery paths.

B. Routing assignment and probe design (RAPD) ILP model

The problem is multi-objective but is presented as a single-objective optimization with a weighted sum. In the following, we define the (input) sets, constants, and variables.

Sets:

$G(V_l, V_p, E)$ a directed network graph where V_l is a set of logical nodes, $V_p \supseteq V_l$ is a set of physical nodes, and E is a set of optical fibers that interconnect optical nodes.

L set of optical lightpaths (demands) forming logical topology, each demand $l \in L$ has $l_s \in V_l$ as the source node and $l_d \in V_l$ as the destination node.

$L^+(v)$ $= \{l \in L : l_s = v\}$, set of lightpaths originating in node $v \in V_l$.

$L^-(v)$ $= \{l \in L : l_d = v\}$, set of lightpaths terminating in node $v \in V_l$.

$P(v, w)$ a set of k-shortest paths in the physical topology between node $v, w \in V_p$; Let $P(l) = P(l_s, l_d)$.

Constants:

α weight determining the impact of a number of register entries in programmable devices on the objective function.

γ weight determining the impact of a number of network capacity units (wavelengths) used in the highest congested link on the objective function.

φ weight determining the impact of recoverability, i.e., protected lightpaths (demands) with physically edge-disjoint paths, on the objective function.

c_e^p $= \{0, 1\}$, 1 if path $p \in P$ contains edge $e \in E$; 0, otherwise.

Variables:

Related to routing in the optical layer.

x_p^l $\in \{0, 1\}$; 1, if the path $p \in P$ is used by demand associated with lightpath $l \in L$.

f_e $\in \mathbb{N}$; the number of network capacity units (wavelengths) used in link $e \in E$.

f $\in \mathbb{N}$; the number of network capacity units (wavelengths) used in any (highest congested) link in the network.

Related to routing in the logical layer.

$\hat{y}_o^l$ $\in \{0, 1\}$; 1 if lightpath $l \in L$ is using another lightpath $o \in L$ associated with another demand on its first recovery path; 0, otherwise.

$\hat{\hat{y}}_o^l$ $\in \{0, 1\}$; 1 if lightpath $l \in L$ is using another lightpath $o \in L$ associated with another demand on its second recovery path; 0, otherwise.

$\hat{x}_e^l$ $\in \{0, 1\}$; 1 if link $e \in E$ is used by lightpaths associated with another demand that is selected as a first recovery path for demand associated with lightpath $l \in L$; 0, otherwise.

$\hat{\hat{x}}_e^l$ $\in \{0, 1\}$; 1 if link $e \in E$ is used by lightpaths associated with another demand that is selected as a second recovery path for demand associated with lightpath $l \in L$; 0, otherwise.

Related to edge order in recovery paths from the node-link formulation in logical layer.

$\hat{\delta}_v^l$ $\in \mathcal{N}$; the weight of node $v \in V_l$ for the first recovery path of demand associated with lightpath l. Nodes that are not used on the path associated with demand d have a weight of 0, while the remaining ones have increasing weights along the selected path.

$\hat{\hat{\delta}}_v^l$ $\in \mathcal{N}$; weight of node $v \in V_l$ for the second recovery path of demand associated with lightpath l. Nodes that are not used on the path associated with demand d have a weight of 0, while the remaining ones have increasing weights along the selected path.

$\hat{k}_v^l$ $\in \{0,1\}$; 1 if node $v \in V_l$ is in the first recovery path used by demand associated with lightpath l (required for calculating register entries).

$\hat{\hat{k}}_v^l$ $\in \{0,1\}$; 1 if node $v \in V_l$ is in the second recovery path used by demand associated with lightpath l (required for calculating register entries).

Related to calculating the number of register entries required.

r_l^v $\in \{0,1\}$; 1 if node $v \in V_l$ requires info in the register entry about lightpath $l \in L$.

$r_l^{src,v}$ $\in \{0,1\}$; 1 if node $v \in V_l$ requires info in the register entry about lightpath $l \in L$ associated with demand originating in that node.

$r_l^{tr,v}$ $\in \{0,1\}$; 1 if node $v \in V_l$ requires info in register entry about lightpath $l \in L$, to forward it to some other node (demand associated with lightpath l is originating in some other node).

r_v $\in \mathcal{N}$; number of register entries required in node $v \in V_l$.

r $\in \mathcal{N}$; number of register entries.

Related to protection/recoverability in the physical layer.

$\hat{q}_l$ $\in \{0,1\}$; if demand associated with lightpath l is protected from physical failure on its first recovery path.

$\hat{\hat{q}}_l$ $\in \{0,1\}$; if demand associated with lightpath l is protected from physical failure on its second recovery path.

q $\in \mathcal{N}$; amount of recovery paths for demands that are physically link disjoint from their primary lightpaths (first and second recovery paths are counted separately as two separate protections).

Objective:

$$\min \alpha r + 12\gamma f - 6\varphi q \quad \textbf{(1)}$$

Constraints:

$$\sum_{p \in P(l)} x_p^l = 1 \qquad \forall l \in L \quad \textbf{(2)}$$

$$f_e = \sum_{l \in L} \sum_{p \in P(l)} x_p^l c_e^p \qquad \forall e \in E \quad \textbf{(3)}$$

$$f_e \leq f \qquad \forall e \in E \quad \textbf{(4)}$$

$$\sum_{o \in L^+(v)} \hat{y}_o^l - \sum_{o \in L^-(v)} \hat{y}_o^l = \begin{cases} 1 & \text{, if } v = l_s \\ -1 & \text{, if } v = l_d \\ 0 & \text{, otherwise} \end{cases} \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(5)}$$

$$\sum_{o \in L^+(v)} \hat{\hat{y}}_o^l - \sum_{o \in L^-(v)} \hat{\hat{y}}_o^l = \begin{cases} 1 & \text{, if } v = l_s \\ -1 & \text{, if } v = l_d \\ 0 & \text{, otherwise} \end{cases} \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(6)}$$

$$\sum_{o \in L^+(v)} \hat{y}_o^l \leq 1 \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(7)}$$

$$\sum_{o \in L^-(v)} \hat{y}_o^l \leq 1 \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(8)}$$

$$\sum_{o \in L^+(v)} \hat{\hat{y}}_o^l \leq 1 \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(9)}$$

$$\sum_{o \in L^-(v)} \hat{\hat{y}}_o^l \leq 1 \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(10)}$$

$$\hat{y}_o^l + \hat{\hat{y}}_o^l \leq 1 \qquad \forall l, o \in L \quad \textbf{(11)}$$

$$\hat{y}_l^l = 0 \qquad \forall l \in L \quad \textbf{(12)}$$

$$\hat{\hat{y}}_l^l = 0 \qquad \forall l \in L \quad \textbf{(13)}$$

$$\hat{y}_o^l + \sum_{p \in P(o)} x_p^o c_e^p - 1 \leq \hat{x}_e^l \qquad \forall l, o \in L, \forall e \in E \quad \textbf{(14)}$$

$$\hat{\hat{y}}_o^l + \sum_{p \in P(o)} x_p^o c_e^p - 1 \leq \hat{\hat{x}}_e^l \qquad \forall l, o \in L, \forall e \in E \quad \textbf{(15)}$$

$$\hat{q}_l + \sum_{p \in P(l)} x_p^l c_e^p + \hat{x}_e^l \leq 2 \qquad \forall l \in L, \forall e \in E \quad \textbf{(16)}$$

$$\hat{\hat{q}}_l + \sum_{p \in P(l)} x_p^l c_e^p + \hat{\hat{x}}_e^l \leq 2 \qquad \forall l \in L, \forall e \in E \quad \textbf{(17)}$$

$$q = \sum_{l \in L} (\hat{q}_l + \hat{\hat{q}}_l) \quad \textbf{(18)}$$

$$\delta_{l_s}^l = 1 \qquad \forall l \in L \quad \textbf{(19)}$$

$$\hat{\delta}_{l_s}^l = 1 \qquad \forall l \in L \quad \textbf{(20)}$$

$$|V_l| \sum_{o \in L^-(v)} \hat{y}_o^l \geq \delta_v^l \qquad \forall l \in L, \forall v \in V_l \setminus l_s \quad \textbf{(21)}$$

$$|V_l| \sum_{o \in L^-(v)} \hat{\hat{y}}_o^l \geq \hat{\delta}_v^l \qquad \forall l \in L, \forall v \in V_l \setminus l_s \quad \textbf{(22)}$$

$$\delta_{o_d}^l \geq \delta_{o_s}^l + 1 - (1 - \hat{y}_o^l)(|V_l| + 1) \qquad \forall l, o \in L \quad \textbf{(23)}$$

$$\hat{\delta}_{o_d}^l \geq \hat{\delta}_{o_s}^l + 1 - (1 - \hat{\hat{y}}_o^l)(|V_l| + 1) \qquad \forall l, o \in L \quad \textbf{(24)}$$

$$|V_l| \hat{k}_v^l \geq \delta_v^l \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(25)}$$

$$\hat{k}_v^l \leq \delta_v^l \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(26)}$$

$$|V_l| \hat{\hat{k}}_v^l \geq \hat{\delta}_v^l \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(27)}$$

$$\hat{\hat{k}}_v^l \leq \hat{\delta}_v^l \qquad \forall l \in L, \forall v \in V_l \quad \textbf{(28)}$$

$$r_l^{src,v} = 1 \qquad \forall l \in L, v = l_s \quad \textbf{(29)}$$

$$r_o^{src,v} \geq \hat{y}_o^l \qquad \forall l, o \in L, v = l_s \quad \textbf{(30)}$$

$$r_o^{src,v} \geq \hat{\hat{y}}_o^l \qquad \forall l, o \in L, v = l_s \quad \textbf{(31)}$$

$$r_l^{tr,v} = 1 \qquad \forall l \in L, v = l_d \quad \textbf{(32)}$$

$$|V_l| r_o^{tr,v} \geq \delta_{o_d}^l - \delta_v^l + 1 - (1 - \hat{y}_o^l)(|V_l| + 1) - (1 - \hat{k}_v^l)(|V_l| + 1) \qquad \forall l, o \in L, \forall v \in V_l \setminus l_s \quad \textbf{(33)}$$

$$|V_l| r_o^{tr,v} \geq \hat{\delta}_{o_d}^l - \hat{\delta}_v^l + 1 - (1 - \hat{\hat{y}}_o^l)(|V_l| + 1) - (1 - \hat{\hat{k}}_v^l)(|V_l| + 1) \qquad \forall l, o \in L, \forall v \in V_l \setminus l_s \quad \textbf{(34)}$$

$$2r_l^v \geq r_l^{src,v} + r_l^{tr,v} \qquad \forall v \in V_l, \forall l \in L \quad \textbf{(35)}$$

$$r_v = \sum_{l \in L} r_l^v \qquad \forall v \in V_l \quad \textbf{(36)}$$

$$r = \sum_{v \in V_l} r_v \quad \textbf{(37)}$$

The objective function in Eq. (1) is a weighted function accounting for three key terms. The first represents the total number of register entries required across all network nodes (αr). The second one represents the wavelengths required in the highest congested link ($12\gamma f$). The third one represents the number of protected connections, defined here as the number of physically protected recovery paths ($-6\varphi q$). The model selects a primary

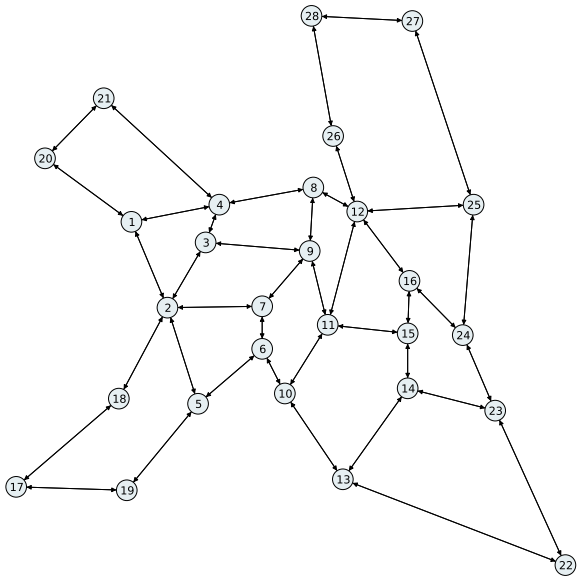


Fig. 3. European network topology used as the physical topology.

path of a single logical hop for each demand. Two logical recovery paths are also selected. In the considered scenario, they have at least two logical hops (assuming there are no two lightpaths between the same node pair), so each demand requires at least 12 register entries (without sharing them). Thus, without sharing register entries, there are approximately $12|L|$ register entries required in the network. There are at most $|L|$ wavelengths in the highest congested link. All primary lightpaths are at most protected by $2|L|$ physically disjoint lightpaths. Therefore, the normalization factors used in the objective function are 12/12=1, 12/1=12, and 12/2=6 for register entries, wavelengths, and protected connections, respectively.

Constraints in Eq. (2) ensure that for each demand, exactly one path in physical topology is selected. Equations (3) and (4) evaluate the required number of wavelengths on each link and in the most congested link, respectively. Flow conservation of selected two recovery paths in logical topology for associated primary demands is satisfied by Equations (5) and (6). Equations (7) – (10) prevent loops in logical topology.

Constraints (11) – (13) ensure wavelength disjointness in the logical layer. Eq. (11) ensures disjointness between two recovery paths, while two later ones disjointness of each recovery path with the primary path. The number of disjoint recovery paths from the primary path in the physical layer is evaluated in two steps. Firstly, logical recovery paths are mapped to physical edges (Constraints (14) and (15)). Secondly, Equations (16) – (18) check if each recovery path shares at least one common edge with its corresponding primary path.

To calculate the number of register entries required, it is necessary to restore the order of links in the routing using node-link formulation. To this end, for each recovery path of each lightpath, an auxiliary graph is created where the selected path source node has weight 1 in Eq. (19) and (20). The nodes' weights not used in the selected path are set to 0 in Eq. (21) and (22). At the same time, the nodes' weights that belong to the path are bounded from the top by the number of nodes in the network. All remaining nodes have weights higher than the previously selected node, as set by Eq. (23) and (24). In particular, if the connection is using lightpath o as a recovery path for lightpath l, then $1 - \hat{y}_o^l$ is 0, forcing $\hat{\delta}_{o_d}^l$ to be higher than $\hat{\delta}_{o_s}^l$. Otherwise, the right-hand side of the equation becomes a negative number. By comparing the weights of two nodes, it is possible to determine if a given node/edge is located before or after another selected node. Finally, a binary variable is created to determine if the node belongs to the path, which is required in the further part of the ILP, Eq. (25) and (28).

The evaluation of the number of register entries is divided into two steps: determining which nodes require an entry about the lightpath that it is originating in that node (source lightpath register entry) or need an entry for a lightpath to pass it further to some other node (transitive lightpath register entry). Source lightpath register entries for each node are defined in Constraints (29) – (31). Equations (32) – (34) allow us to find transit register entries for a given node. In more detail, Eq. (33) and (34) force $r_o^{tr,v}$ to be 1 if node v requires register entry about lightpath o. It is required if that lightpath belongs to specific demand recovery paths, the node belongs to that path, and the edge (lightpath) is either after node v in that path or it is an edge that directly goes into node v. If the lightpath does not belong to demand recovery paths, then the $1 - \hat{y}_o^l$ part of Eq. (33) is 1 and the right-hand side becomes a negative value. If node v does not belong to the path, then $1 - \hat{k}_v^l$ is 1 and the right-hand side is a negative value. If the lightpath is not located after node v in the path, or it is not directly preceding node v, then $\hat{\delta}_{o_d}^l - \hat{\delta}_{o_v}^l$ is negative.

Source nodes of primary/recovery paths are excluded from the constraints as they are accounted for in the source register entries. Edge (lightpaths) positions in the path can be obtained by checking if the weight of the edge's destination is higher or equal to the weight of the node (if it is equal, it is the edge that is going into that node). Note that primary paths have only one logical hop, so the destination node requires the same register entry as its source node. Finally, Eq. (35) checks if a node requires a register entry for lightpath regardless of whether it is the source or transitive register entry, and the number of register entries required is evaluated in Equations (36) and (37).

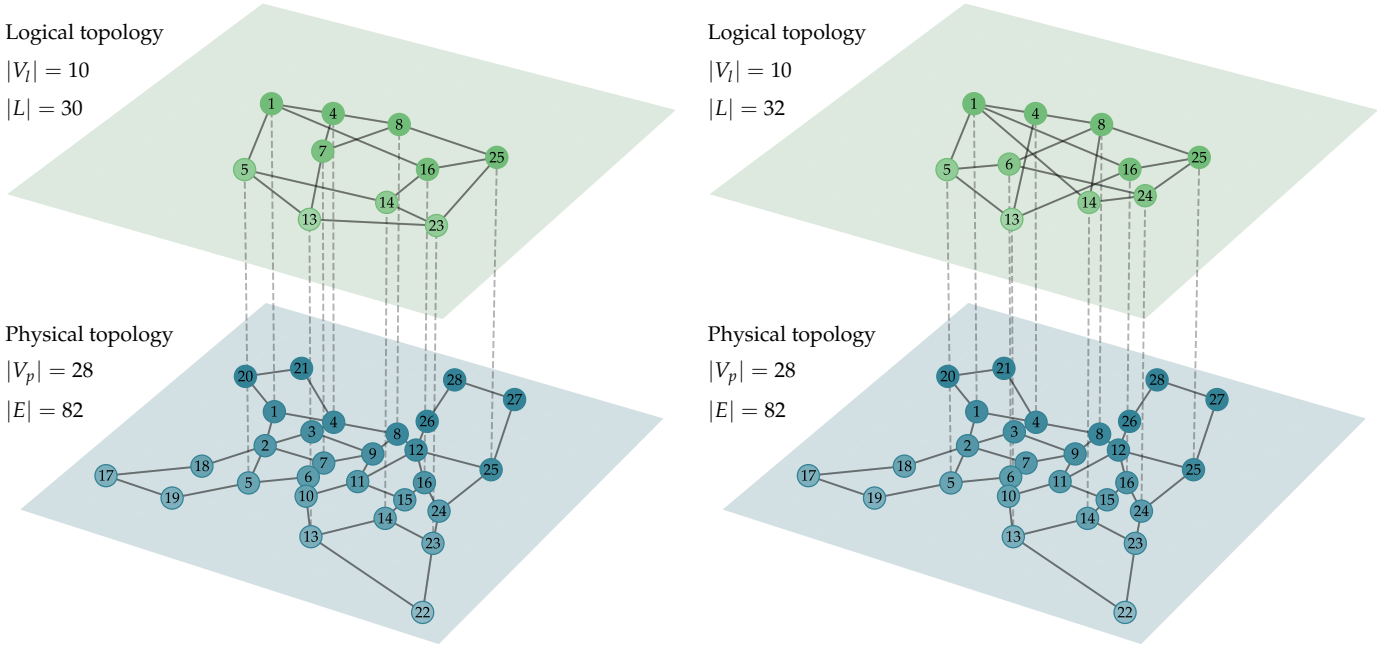
If needed, the number of register entries at each node can be bound to a maximum value by introducing the following constraint:

$$r_v \leq R_{max} \qquad \forall v \in V_l, \tag{38}$$

where R_{max} is the maximum number of available register entries.

The number of variables in the model is equal to $|L||P| + 2|L|^2 + 2|L||E| + 7|L||V_l| + |E| + |V_l| + 2|L| + 3$. As $|E| \geq |V_l|$, thus the number of variables is bounded by $O(|L|(|L| + |P| + |E|))$. The number of constraints is $2|L|^2|E| + 2|L|^2|V_l| + 13|L||V_l| + 5|L|^2 + 2|L||E| + 7|L| + 2|E| + |V_l| + 3$, thus it is bounded by $O(|L|^2|E|)$.

The optimization problem described by the presented ILP model is NP-hard. It can be proven by considering a restricted version of the corresponding decision problem with a single connection and single recovery path. The restricted version is equivalent to the survivable routing problem of a logical ring (SRRP). The decision problem for SRRP has been proven to be $\mathcal{NP}$-complete in [21]. If the corresponding decision problem for our problem is NP-complete, then the optimization problem is NP-hard.



(a) Example 1.

(b) Example 2.

Fig. 4. Examples of physical-to-logical topology mapping and 3-vertex logically-connected topology.

5. NUMERICAL EXPERIMENTS

In this section, we assess the performance of the proposed model. We run experiments on the European network topology (Fig. 3) comprising 28 nodes and 82 links, identified as *nobel-eu* in the SNDlib dataset [22]. We randomly select a subset of nodes for each experiment that communicate among themselves. To allow for physically-disjoint primary and recovery paths, we select at least 3-connected nodes. The number of communicating logical nodes is set to $|V_p| = \{6, 7, \ldots, 11\}$. Once the communicating nodes are selected, we randomly generate demands between them to form a 3-vertex logically connected topology. Fig. 4 shows two examples of such mapping. Both mappings form a logical topology with 10 logical nodes. However, Fig. 4(a) has a logical topology with 30 logical links, while Fig. 4(b) has 32 logical links. For each node pair in the physical layer, we precompute 8 shortest paths. The ILP instances are solved using CPLEX solver version 22 with a 7-day running time limit.

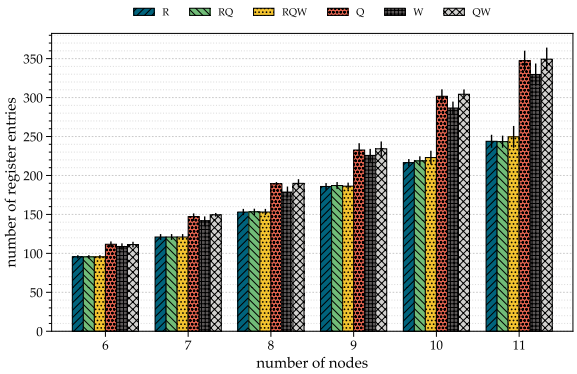
The results presented in this section are averaged over 10 randomly selected logical topologies and mappings, as illustrated in Fig. 4. The recoverability level on both recovery paths refers to the ratio of demands physically disjoint on both recovery paths to all the demands. The recoverability level on at least one recovery path refers to the ratio of demands physically disjoint on at least one recovery path to all the demands.

Each of the logical topologies is solved with the following 6 scenarios (weights):

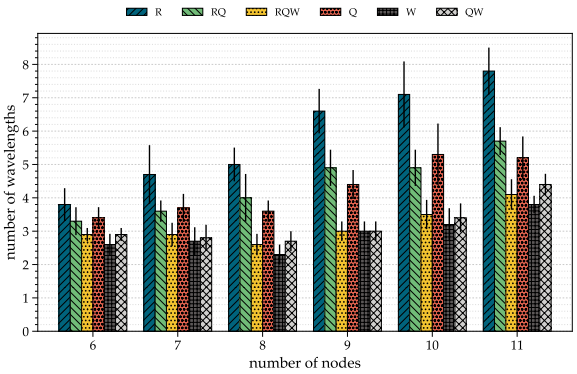
- R - minimize the number of register entries, ($\alpha = 1$, $\gamma = 0$, $\varphi = 0$)
- RQ - minimize the number of register entries first, and then maximize the recoverability level ($\alpha = 1$, $\gamma = 0$, $\varphi = 0.01$)
- RQW - minimize the number of register entries first, then maximize the recoverability level, and then minimize the number of wavelengths in the most congested link ($\alpha = 1$, $\gamma = 0.01$, $\varphi = 0.01$)
- W - minimize the number of wavelengths in the most congested link ($\alpha = 0$, $\gamma = 1$, $\varphi = 0$)
- Q - maximize recoverability level ($\alpha = 0$, $\gamma = 0$, $\varphi = 1$)
- QW - maximize recoverability level first, and then minimize the number of wavelengths in the most congested link ($\alpha = 0$, $\gamma = 0.01$, $\varphi = 1$)

Fig. 5 shows the results of the experiments. Fig. 5(a) shows the number of register entries required for various scenarios as a function of number of nodes in logical topology. The scenario that optimizes the number of register entries (R) reduces this metric by up to 28.32% for 11 nodes compared to scenarios without register entries minimization in the objective function (i.e., between the average value of R, RQ, and RQW and the average value of Q, W, and QW). Optimizing for additional metrics (i.e., QW) does not significantly impact the number of register entries. For instance, RQW only requires 0.46% more register entries than R. However, we can observe that when the number of register entries is not optimized, this metric significantly increases.

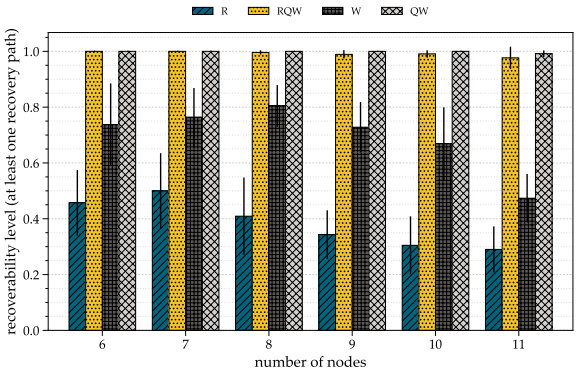
Fig. 5(b) shows the number of wavelengths required in the most congested link. Intuitively, when this term is not considered objective, such as in R, RQ, and Q scenarios, its number is much higher than in the remaining ones that focus on it. Scenario R requires 80.10% more wavelengths on average than scenario W. However, adding wavelength awareness as a small weight to the objective allows for a decrease in that value to 2.64%



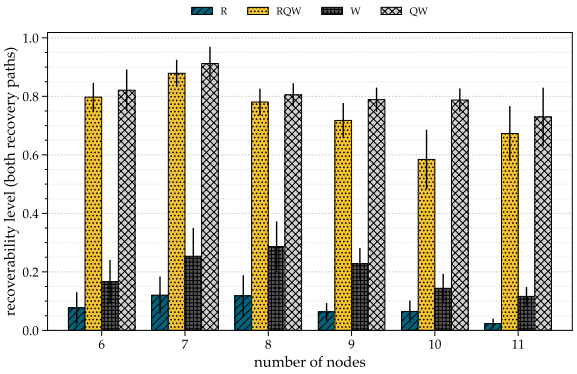
(a) Number of register entries.



(b) Number of wavelengths in the most congested link.



(c) Number of lightpaths that are protected by at least one physically disjoint path.



(d) Number of lightpaths that are protected by both physically disjoint paths.

Fig. 5. Results for the various logical topologies mapped to the European physical network topology with logical communicating nodes varying from 6 to 11. Legends represent the metric(s) optimized by the objective function in the priority order. R: number of register entries; Q: recoverability level; W: number of wavelengths in the most congested link.

(difference between RQW and W). At the same time, RQW allows a decrease in the number of register entries by 7.73% on average when compared to W. It is worth noting that QW requires a different number of wavelengths than RQW, but the different weights for the recoverability level factor can justify it. In the case of 11 nodes, the average increase in the number of wavelengths is 0.25 between the RQW and W scenario, and the average decrease in the number of register entries is 80.75.

Fig. 5(c) and 5(d) show recoverability level as a ratio of connections protected from physical link failures to all connections with at least one recovery path connection or on both recovery paths, respectively. RQW and QW scenarios guarantee physical protection on at least one recovery path for all the cases. The QW scenario physically protects connection with two disjoint recovery paths more often than the RQW strategy, as its weight for the recoverability level is higher than in the RQW strategy. The difference in recoverability ratio with at least one recovery path between the R and RQW strategy is 58.44%, showing that by slightly increasing the number of register entries, one can achieve a much better recoverability level. This difference is even larger regarding recoverability level with two recovery paths, where the RQW strategy protects 62.07% more connections.

The average running time to solve instances was 0.57, 5.94, 34.53, 57.85, and 55.50 minutes for 6, 7, 8, 9, 10, and 11 nodes, respectively. Considering the first quartile as $Q1$ and the third quartile as $Q3$, these averages ignore times outside of the range $Q1 - 1.5 \times (Q3 - Q1)$ and $Q3 + 1.5 \times (Q3 - Q1)$. It is important to note that the computation complexity of instances varies significantly, as well as suffer influences from other concurrently running processes. In our case, some of the instances reached a 7-day running time limit. We also solved instances for a non-uniform traffic profile where a few high-degree nodes have multiple demands between them. The results show a similar trend and are not included in this paper due to space constraints.

6. CONCLUSION

In this work, we envision an architecture where optical nodes are equipped with programmable devices (e.g., P4-enabled pluggables or switches). This architecture allows the deployment of on-device, at line speed, (soft) failure detection and mitigation. Soft failure detection works by processing telemetry data mea-

sured and transmitted by the nodes. The mitigation leverages spare capacity on existing connections to minimize the disruption caused by the soft failure. We propose an ILP model for joint routing assignment, alternative recovery path selection, telemetry forwarding graph, and probe placement based on the proposed architecture and intuition. The ILP model optimizes optical network metrics such as the number of wavelengths in the most congested link, recoverability ratio, and metrics related to the programmable devices such as the number of register entries required. Six scenarios were designed to show the impact of optimizing certain metrics on the designed solution. The solution computes the physical paths to serve the optical demands and the logical recovery path in case of soft failures. The results reveal trade-offs to be considered among the three primary metrics analyzed. By increasing the number of required wavelengths in the most congested link by 11.57%, it is possible to reduce the number of register entries by 14.79%.

This work showcases how on-device programmability can be exploited to provide lightweight and fast soft-failure detection and mitigation. A centralized controller can solve the proposed model, with the solution being deployed in the programmable devices. Then, programmable devices can execute the soft failure detection and mitigation at line speed without the intervention of a centralized controller.

The complete realization of the vision proposed herein depends on a few elements that are out of the scope of this paper. For instance, the computation of thresholds to detect soft failures is a challenging problem. Potential solutions are the use of empirically defined values or the use of AI/ML. Alternatively, using on-device AI/ML models could allow for dynamically adjusted thresholds. Moreover, hard failures may disrupt the recovery paths of several connections, requiring recomputing the recovery paths for those connections.

FUNDING

This work was supported by the ECO-eNET project, which received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under grant agreement No. 10113933. The JU receives support from the European Union's Horizon Europe research and innovation programme. This work has also been supported by the EU B5G-OPEN Project (G.A., 101016663).

REFERENCES

1. The P4.org Applications Working Group, "In-band Network Telemetry (INT) Dataplane Specification, Version 2.1," https://github.com/p4lang/p4-applications/blob/master/docs/INT_v2_1.pdf [Accessed: 11.13, 2024].
2. J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004).
3. F. Cugini, C. Natalino, D. Scano, F. Paolucci, and P. Monti, "P4-based telemetry processing for fast soft failure recovery in packet-optical networks," in *Optical Fiber Communications Conference and Exhibition (OFC),* (2023), p. M1G.2.
4. J. Hyun, N. Van Tu, and J. W.-K. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *IEEE/IFIP Network Operations and Management Symposium (NOMS),* (2018), pp. 1–7.
5. L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, "In-band network telemetry: A survey," Comput. Networks **186**, 107763 (2021).
6. N. S. Kagami, R. I. T. da Costa Filho, and L. P. Gaspary, "CAPEST: Offloading network capacity and available bandwidth estimation to programmable data planes," IEEE Transactions on Netw. Serv. Manag. **17**, 175–189 (2020).
7. S. Nam, J. Lim, J.-H. Yoo, and J. W.-K. Hong, "Network anomaly detection based on in-band network telemetry with rnn," in *IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia),* (2020), pp. 1–4.
8. Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, "HPCC: High precision congestion control," in *Proceedings of the ACM Special Interest Group on Data Communication,* (Association for Computing Machinery, New York, NY, USA, 2019), SIGCOMM '19, p. 44–58.
9. Y. Lin, Y. Zhou, Z. Liu, K. Liu, Y. Wang, M. Xu, J. Bi, Y. Liu, and J. Wu, "NetView: Towards on-demand network-wide telemetry in the data center," in *IEEE International Conference on Communications (ICC),* (2020), pp. 1–6.
10. T. Pan, E. Song, Z. Bian, X. Lin, X. Peng, J. Zhang, T. Huang, B. Liu, and Y. Liu, "INT-path: Towards optimal path planning for in-band network-wide telemetry," in *IEEE Conference on Computer Communications (INFOCOM),* (2019), pp. 487–495.
11. N. Katta, A. Ghag, M. Hira, I. Keslassy, A. Bergman, C. Kim, and J. Rexford, "Clove: Congestion-aware load balancing at the virtual edge," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies,* (Association for Computing Machinery, New York, NY, USA, 2017), CoNEXT '17, p. 323–335.
12. A. Karaagac, E. De Poorter, and J. Hoebeke, "Alternate marking-based network telemetry for industrial wsns," in *IEEE International Conference on Factory Communication Systems (WFCS),* (2020), pp. 1–8.
13. R. Ben Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, "PINT-: Probabilistic in-band network telemetry," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication,* (2020), pp. 662–680.
14. R. Hohemberger, A. G. Castro, F. G. Vogt, R. B. Mansilha, A. F. Lorenzon, F. D. Rossi, and M. C. Luizelli, "Orchestrating in-band data plane telemetry with machine learning," IEEE Commun. Lett. **23**, 2247–2251 (2019).
15. S. Tang, S. Zhao, X. Pan, and Z. Zhu, "How to use in-band network telemetry wisely: Network-wise orchestration of sel-int," IEEE/ACM Transactions on Netw. **31**, 421–435 (2023).
16. A. G. Castro, A. F. Lorenzon, F. D. Rossi, R. I. T. d. C. Filho, F. M. V. Ramos, C. E. Rothenberg, and M. C. Luizelli, "Near-optimal probing planning for in-band network telemetry," IEEE Commun. Lett. **25**, 1630–1634 (2021).
17. D. Bhamare, A. Kassler, J. Vestin, M. A. Khoshkholghi, and J. Taheri, "Intopt: In-band network telemetry optimization for nfv service chain monitoring," in *IEEE International Conference on Communications (ICC),* (2019), pp. 1–7.
18. R. Hohemberger, A. F. Lorenzon, F. Rossi, and M. C. Luizelli, "Optimizing distributed network monitoring for nfv service chains," IEEE Commun. Lett. **23**, 1332–1336 (2019).
19. L. Tan, W. Su, W. Zhang, H. Shi, J. Miao, and P. Manzanares-Lopez, "A packet loss monitoring system for in-band network telemetry: Detection, localization, diagnosis and recovery," IEEE Transactions on Netw. Serv. Manag. **18**, 4151–4168 (2021).
20. K. Zhang, L. Liu, L. Tan, Y. Zhang, W. Gao, and W. Zhang, "Efficient clustered network telemetry based on failure awareness," in *Asia-Pacific Network Operations and Management Symposium (APNOMS),* (2022), pp. 1–4.
21. A. Sen, B. Hao, and B. H. Shen, "Survivable routing in wdm networks," in *Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications,* (2002), pp. 726–731.
22. S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0–survivable network design library," Networks. **55**, 276–286 (2010).