Locally Automated Restoration in SDN Disaggregated Networks

NICOLA SAMBO

Scuola Superiore Sant'Anna, Via G. Moruzzi 1, 56124 Pisa, Italy

nicola.sambo@sssup.it

Compiled February 5, 2020

Multi-vendor interoperability and disaggregation are attracting the interest of network operators as a way to avoid vendor lock-in, thus opening the market and, possibly, reducing costs. NETCONF, in concert with YANG data models, has been identified as the Software Defined Networking (SDN) configuration protocol for these networks. Currently, several YANG models describe devices in a vendor-neutral way; however, there is a lack of YANG models describing functions. Moreover, the centralization of the control plane may suffer from scalability issues during critical situations (e.g., link failures) given that several restoration requests will arrive to the SDN controller close in time.

This paper investigates an innovative paradigm of restoration for disaggregated SDN networks named *Delegated Restoration*, which operates in a hybrid centralized-distributed manner. Before a failure occurs, the backup lightpath is centrally computed by the SDN controller and based on this computation the controller informs the network devices (switches and transponders), through NETCONF, of the reconfigurations to perform in case of failure. Simulations show that Delegated Restoration can reduce the restoration time with respect to a fully centralized approach, making it a candidate for impacting the operation, administration and maintanance of next-generation networks.

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

---

1. INTRODUCTION

During the last few years, network operators and service providers have shown great interest in the inter-operability of network elements across vendors and in the disaggregation of the software from the hardware [1–3]. Indeed, current deployed networks are typically locked into a single vendor, thus upgrades (e.g., the introduction of transponders supporting higher bit rates) must be performed by that same vendor. Similarly, the control and management system — including any control operation (e.g., transponder configuration) and operation in response to faults [4] — is vendor dependent.

Inter-operability among network elements from different vendors allows operators to have more options when building out their networks, leading to lower costs and the ability to adopt best-of-breed technology. The shortest-term solution for inter-operability is the so-called *open-line system*, where reconfigurable optical add&drop multiplexers (ROADMs) are provided by one vendor and transponder pairs by other vendors. With this approach, an operator can adopt the best transponder technology for its network, independent from the choice of line system. A demonstration has been shown in [5], where transponders from eight different vendors operated in the same testbed. Moreover, disaggregating the software from the hardware enables an operator to choose the management system without being tied to a vendor. A relevant limitation to achieving a vendor-neutral and disaggregated network lies in the agreement on data models among companies. Indeed, control and management software should operate on or manipulate hardware parameters in a vendor-neutral way so that any operator can implement/adopt/buy the software independent of the supplier. The agreement on how to describe parameters of ROADMs, amplifiers, equalizers, transponders, etc. is thus a fundamental step for the deployment of these networks.

This important discussion/work is carried on within several consortiums and projects involving the major companies of the world: OpenROADM [6], OpenConfig [7], Telecom Infra Project [8], and also within the Internet Engineering Task Force (IETF) [9]. A first agreement has been made with regard to the selected Software Defined Networking (SDN) protocol for network configuration, i.e., the Network Configuration (NETCONF) protocol [10], which is based on the YANG data modeling language [11]. With this assumption, network devices are described through YANG data models. Thus, the identification of YANG data models is a fundamental step shared by operators and vendors. Discussions are still ongoing within the aforementioned projects and consortiums. Examples of YANG data models can be downloaded from [6].

Up until now the effort has been mainly focused on the description of hardware *configuration parameters* and *state parameters* (the latter can be utilized for monitoring purposes). However, the programmability of a network also requires the support of several advanced functionalities such as recovery and transmission adaptation (i.e., the reconfiguration of modulation format and coding based on actual network physical layer conditions) [12]. Currently, there is a lack of YANG models describing network functionalities. A first attempt has been made within the netmod group of the IETF [13], proposing a draft where network functions are associated with specific network states (e.g., physical layer degradations) with an approach based on a Finite State Machine (FSM). Leveraging this IETF YANG model for FSM, a field trial has been performed at Telecom Italia Mobile (TIM), Torino, Italy [14] demonstrating automatic transmission parameter adaptation through FSM. In that work, each state in the FSM is associated with a specific level of bit error rate (BER) and, when physical layer conditions change such that a BER threshold level is crossed, a state transition is triggered, causing the reconfiguration of the modulation format. This demonstrated approach is capable of responding to physical-layer degradations but not link failures. In general, network functionalities and reliability still require investigation.

Reliability in SDN networks is crucial given the centralization of the control plane. Indeed, in SDN networks, a central controller holds the overall view of the network and performs configurations. In case of failures or degradations, the SDN controller can be overloaded due to the processing of several instantaneous dynamic restoration requests, thus introducing delays in the recovery. One way to speed up the recovery is the adoption of 1+1 protection [15]. However, it requires the occupation of extra spectrum and additional transponders/modules; typically, it is implemented by operators only for high-priority traffic. Another alternative is distributed (rather than centralized) dynamic restoration, typically implemented through Generalized Multi Protocol Label Switching (GMPLS) [16]. However, it often may require that path computation intelligence be distributed in the control plane. Additionally, dynamic restoration based on GMPLS would not permit a full migration toward NETCONF. In the literature, researchers have also proposed hybrid centralized/distributed solutions, e.g., based on a Path Computation Element and GMPLS [17–20]. Another example of hybrid restoration is proposed in [21], where a central controller pre-computes restoration alternatives (e.g., protection paths considering the sharing of resources) and instructs physical nodes of the recovery actions to take upon failure. In this scheme, the restoration resources to be used (e.g., the wavelength) are not specified by the controller, and the scheme relies on a custom signaling system or GMPLS. Overall, we see that reliability in SDN networks considering the current trends of disaggregation and NETCONF compatibility is a topic that still requires investigation and solutions to overcome scalability issues.

In this paper, an innovative approach named *Delegated Restoration* is investigated for SDN-based disaggregated networks operating with the NETCONF protocol. This scheme was first proposed in [22]. According to the proposed approach, the SDN controller pre-computes restoration lightpaths (path and spectrum resources) and informs the agents of the involved devices of the actions to perform for rerouting a lightpath in case of link failure (the agents are the interfaces between the network devices and the SDN controller). Thus, when a link failure occurs, the agents already know the reconfiguration settings and can react to failures by simply applying pre-loaded instructions to accomplish the rerouting. This method also overcomes the problem of SDN controller unavailability (e.g., controller failure) during a failure. NETCONF and YANG are used to install instructions. Such a restoration scheme is i) scalable, limiting the involvement of the SDN controller during failures; ii) efficient, relying on centralized path computation and not requiring any path computation intelligence to be distributed in the control plane nor the allocation of dedicated protection resources. In this paper, the operation of Delegated Restoration is explained in detail and simulations are provided to evaluate the decrease in recovery time with respect to a fully centralized approach. Simulations show the high potential of the proposed method.

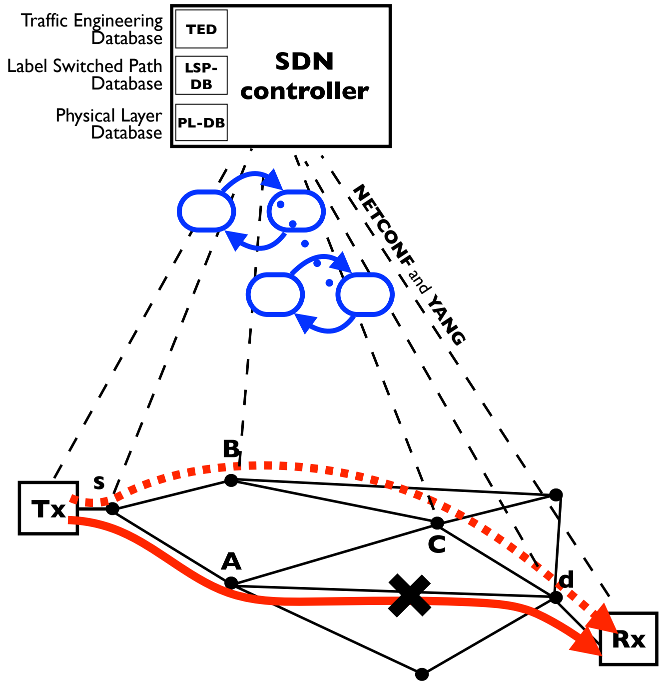


Fig. 1. Considered network architecture.

2. DELEGATED RESTORATION

An SDN-based optical network, as in Fig. 1, is assumed to be controlled with an out-of-band control plane based on NETCONF and YANG. ROADMs and transponders are equipped with power monitors per channel (e.g., the Lumentum White Box [23]). Flexible transponders are assumed to support multiple modulation formats, code, bit rate and symbol rate values. The SDN controller performs routing and spectrum assignment (RSA) by relying on the following databases: the Traffic Engineering Database (TED) storing traffic engineering information; the Label Switched Path Database (LSP-DB) storing lightpath information; and the physical layer database (PL-DB) storing physical layer information required to perform Quality of Trans-

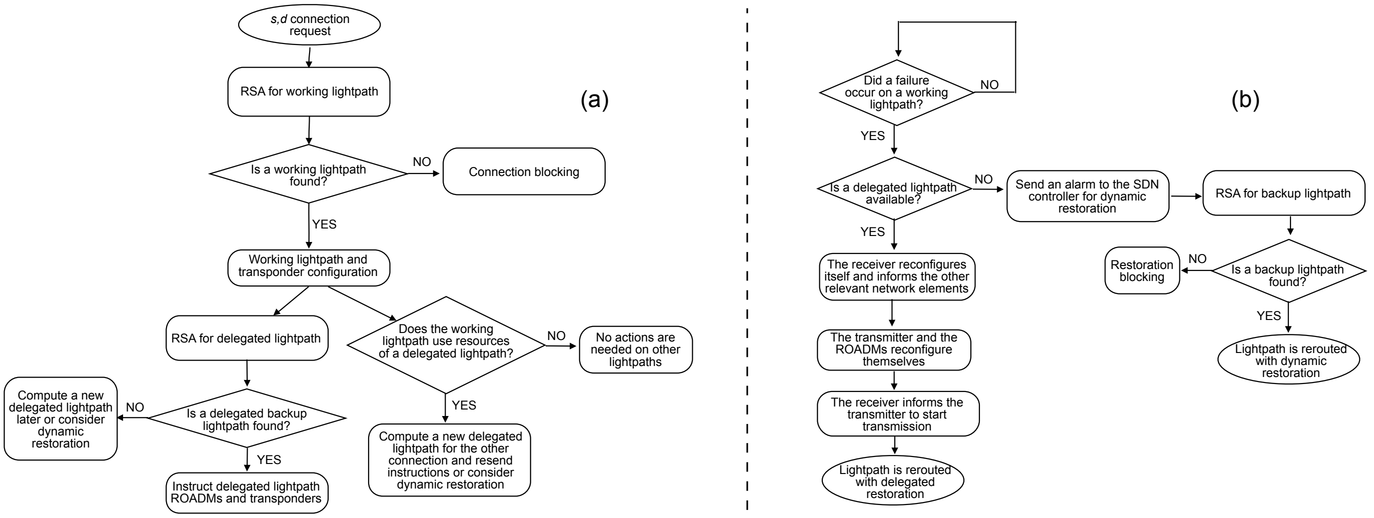


Fig. 2. Flow charts describing Delegated Restoration: (a) upon connection request; (b) upon failure.

mission (QoT) estimation. These databases are exploited for the RSA of any lightpath, including any *delegated lightpath* (i.e., the precomputed lightpath for recovery).

The resources for the delegated lightpath are not pre-reserved; rather the network devices that would be involved in the restoration are informed of how this lightpath should be configured if a failure occurs (as will be detailed below), in order to speed up recovery without involving the SDN controller at the time of failure. It may happen that a working lightpath needs to use the resources identified for the delegated lightpath of another connection. In this scenario, another delegated lightpath can be computed or traditional dynamic restoration can be utilized in case this other connection fails.

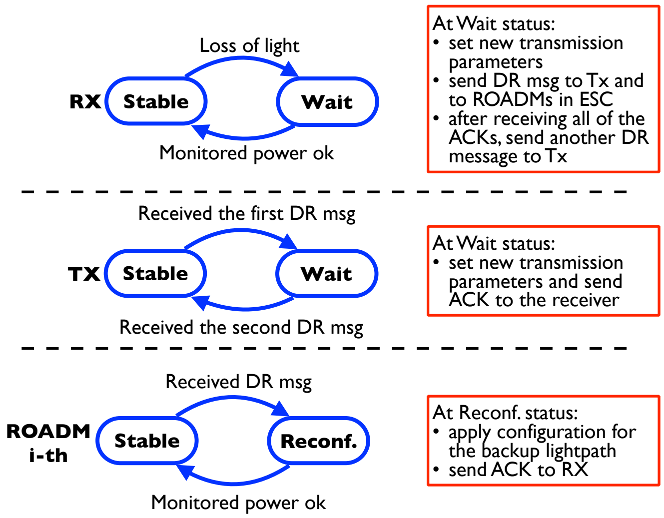
Delegated Restoration is described through the flow charts shown in Fig. 2. Upon connection request (Fig. 2a), RSA is performed for the working path. If no working path with at least one frequency slot satisfying the continuity constraint is found, the connection is blocked. A frequency slot is defined with a central frequency and a bandwidth through the ITU-T *n* and *m* parameters, respectively [24]. Based on this computation, the SDN controller sets up the working lightpath by configuring via NETCONF the ROADMs and the transponders at the transmitter and receiver ends. Throughout the paper, we assume that paths do not require any intermediate regeneration.

After the setup of the working lightpath, the SDN controller pre-computes a delegated lightpath for that connection, again exploiting the databases in Fig. 1 and accounting for QoT. An alternative route, a new portion of spectrum, and transmission parameters such as the modulation format and the symbol rate are computed for the delegated lightpath. We assume that path computation for the delegated lightpath is based on the shortest path that is fully link-disjoint from the working path, but other algorithms could be applied. Two strategies for spectrum assignment for the delegated lightpath are compared in the simulations (i.e., first-fit and last-fit). If a delegated lightpath is not found (e.g., due to the lack of a frequency slot satisfying the continuity constraint), the new connection is not blocked (i.e., the working path is still established). A delegated lightpath can be recomputed later when the network state changes; alternatively, classical dynamic restoration can be utilized if a failure impacts this connection.

Fig. 1 shows an example where the working path is s-A-d and the delegated path is s-B-C-d. After the delegated backup path is computed, a cornerstone step of Delegated Restoration takes place: the SDN controller informs the involved network devices along the backup path of the local reconfigurations to perform for recovering this lightpath in case of failure. The involved network devices include: transmitter, receiver, and ROADMs at s, B, C, and d. Instructions are sent through NETCONF and YANG to the relevant agents by means of a FSM and following the YANG model for a FSM proposed in the IETF draft [13]. After this occurs, the relevant devices along the backup route are aware of the required actions to perform if a failure occurs. Thus, there is no need to involve the SDN controller during the failure, assuming the pre-computed delegated lightpath is still available.

When a failure occurs (Fig. 2b), if a delegated lightpath is available, then Delegated Restoration is applied according to the instructions in the FSMs, as described below. If a delegated lightpath is not available, then dynamic restoration can be applied; to this purpose an alarm is sent to the SDN controller which can perform RSA for the backup path based on the current network state. If no backup path with a frequency slot satisfying the continuity constraint is found, restoration blocking is experienced.

When a failure is detected (e.g., through loss of light) and a delegated lightpath is available, the receiver self-reconfigures the transmission parameters for the delegated lightpath (e.g., new central frequency) and informs the transmitter about this reconfiguration. In this paper, to enable such a step of receiver-transmitter synchronization, an enhanced supervisory channel (ESC) is also proposed. ESC runs over the control plane and logically connects selected network elements such as transponder pairs and transponders with ROADMs. Thus, multiple ESCs are established for a given connection. An ESC differs from a typical optical supervisory channel, which runs on a wavelength between 1504.5 and 1517.5 nm and which is used only between ROADMs [25]. ESC is exploited by Delegated Restoration to trigger state transitions in the FSM at specific network elements. To this purpose, a lightweight message, named Delegated Restoration (DR) message, is exchanged through the ESC. The DR message does not carry backup path information and it is only exploited to inform a network element that a state



At Wait status:
- set new transmission parameters
- send DR msg to Tx and to ROADMs in ESC
- after receiving all of the ACKs, send another DR message to Tx

At Wait status:
- set new transmission parameters and send ACK to the receiver

At Reconf. status:
- apply configuration for the backup lightpath
- send ACK to RX

Fig. 3. FSM installed at each network element of the backup path to implement Delegated Restoration.

transition in the FSM has to be performed.

Moreover, similar to the communication between receiver and transmitter, the receiver informs the ROADMs along the backup path about reconfiguration. These nodes, after reconfiguration, send an acknowledgement (ACK) to the receiver along the ESC. After the receiver receives an ACK from each device along the backup path (i.e., each ROADM and the transmitter), it sends another DR message to the transmitter indicating that it is OK to start transmission

The operations involved during Delegated Restoration are modelled with a FSM, as shown in Fig. 3, which shows the installed FSM at each involved device in the backup path: transmitter, receiver and ROADMs. Under normal conditions, all of the FSMs are in the "stable" state. When a failure occurs in the working path, the receiver detects the loss of light, which triggers a state transition to the "wait" state. Thus, the receiver reconfigures itself according to the transmission parameters associated with the backup lightpath such as the central frequency. Moreover, the backup path may require a different modulation format, symbol rate, and coding than the working path, e.g., it may be longer, and thus more affected by physical layer impairments. At this point, the synchronization between the receiver and the transmitter needs to be done. To this purpose, the agent of the receiver informs the agent of the transmitter about the state transition. Thus, a DR message is sent from the receiver to the transmitter. The reception of this message at the transmitter triggers its transition to the "wait" state, and the transmitter reconfigures itself to the new required transmission parameters. In this state, the transmitter does not send any data since the working path failed and the new path is not configured yet. The transmitter sends an ACK to the receiver along the ESC.

Moreover, the receiver sends the DR message to the relevant ROADMs using ESCs. Each ROADM receiving this message applies the local reconfiguration for the backup lightpath and sends an ACK to the receiver. After receiving ACKs of all of these reconfigurations, the receiver sends a DR message to the transmitter, which is informed that the backup path has been set up; thus transmission can now take place along the backup path.

Table 1. State attributes in the YANG model

id	description
1	Stable
2	Wait
3	Reconf.

When the transmitter receives the DR message from the receiver, it returns to the "stable" state. Similarly, when the ROADMs and the receiver monitor a proper channel power level at the ports of the backup lightpath, their state returns to "stable". The SDN controller can be informed about the accomplishment of Delegated Restoration so that it can tear down the working path. Moreover, the SDN controller can also reprogram the FSMs of the rerouted connection to inform it of a new backup path. (In general, recalculating a delegated restoration lightpath can be done periodically as the network spectrum usage changes).

A delegated lightpath may not be available when a failure occurs due to one of the following: a delegated lightpath was not found when the connection was first established; a working lightpath of another connection has used resources of the delegated lightpath; or, multiple concurrent failures have occurred. In these scenarios, dynamic restoration can be exploited. (We assume that there is enough time for the SDN to inform the relevant equipment of the unavailability of a delegated lightpath, so that delegated restoration is not attempted when a failure occurs.) In this case, the device detecting the failure simply sends an alarm to the SDN controller, which performs RSA for the backup path based on the current network state, and enacts reconfiguration of the relevant network elements.

The spectrum assigned to a delegated lightpath accounts for spectrum availability and also considers the spectrum assigned to other delegated lightpaths. In this paper, a dedicated (though not reserved) portion of spectrum is assigned to each delegated lightpath. However, delegated restoration can optionally adopt the sharing of backup resources (e.g., different connections having link-disjoint working paths can share the same delegated lightpath). In this case, with respect to *dedicated* delegated lightpath, fewer delegated lightpaths would be present in the network. This can impact the number of connections that need to rely on dynamic restoration after a failure; thus resource sharing among the delegated lightpaths can have an impact on the recovery delay, which can be investigated in future work.

A discussion is warranted on disaggregation and vendor neutrality. FSM is a generic model that can describe several functionalities (some use cases are reported in [13]). In the case of Delegated Restoration, the involved network devices at the data plane include ROADMs and transponders. Thus, vendor neutrality of Delegated Restoration is enabled if vendor-neutral YANG models for ROADMs and transponders are adopted. Regarding the switching technology, the configuration should be performed based on the standard ITU-T n and m parameters [24] (describing the central frequency and the bandwidth, respectively). Regarding transponders, the process of converging on an agreement for a vendor-neutral data model is taking longer because several parameters are involved (e.g., including at the Optical Transport Network layer) and each vendor typically adopts a proprietary solution for the transmission technique (e.g., coding). An example of a transponder YANG model can be found in [26].

```
+--rw current-state?   leafref
   +--rw states
      +--rw state [id]
         +--rw id              state-id-type
         +--rw description?   string
         +--rw transitions
            +--rw transition [name type]
               +--rw name           string
               +--rw description?   string
               +--rw actions
                  +--rw action [id]
                     +--rw id              action-id-type
                     +--rw description?   string
                     +--rw execute
                  +--rw next-state?   leafref
```

Fig. 4. FSM YANG model.

Table 2. List of actions

id	description
1	transmission parameter reconfiguration
2	send DR message to the transmitter to set current-state id=2
3	send DR message to the ROADMs to set current-state id=3
4	ROADM reconfiguration
5	send DR message to the transmitter to start transmission setting current-state id=1

3. FINITE STATE MACHINE YANG MODEL FOR DELEGATED RESTORATION

Instructions to network devices are given in the form of a FSM through the NETCONF protocol. To this purpose the YANG model for FSM proposed in the IETF draft [13] is here refined to support Delegated Restoration. The FSM YANG model is shown in Fig. 4. A list of states is envisioned. The attributes of “state” include an identifier and a descriptor, whose possible values are summarized in Tab. 1, reflecting Fig. 3. One of these states is the “current-state”. As shown in the FSMs of Fig. 3, only one transition is admitted from each state.

The transition at the receiver from the “stable” state is triggered by the loss of light and implies a list of “actions”; i.e., the actions with identifiers equal to 1, 2, and 3 in Tab. 2. The attribute “execute” actually implements the action. Then, a next state is achieved (i.e., “next-state”; in this case the receiver moves to the state with id=2). The DR message sent to the transmitter is simply a NETCONF `edit-config` message [10], which changes the value of the “current-state” attribute at the transmitter, setting its value to id=2. Setting the “current-state” to “Wait” (id=2) in the FSM at the transmitter directly triggers the reconfiguration of transmission parameters suitable for the backup lightpath. After reconfiguring the parameters, the acknowledgement is implemented with the standard NETCONF `ok` message [10] sent to the receiver over ESC.

The DR messages sent to the ROADMs changes the value of their “current-state” attribute to id=3. This directly triggers the reconfiguration of each of these ROADMs in accordance with the pre-computed backup lightpath (action with identifier equal to 4). When the agent at the receiver receives all the NETCONF `ok` messages as a consequence of the NETCONF `edit-config` implementing the DR messages, it confirms that the backup path has been established and the transmitter reconfigured. A final action, sending a DR message to the transmitter to indicate it can start transmission, is then performed (id=5). Upon receipt of this DR message, the state of the transmitter transitions to “Stable”. (Note that additional state transitions still need to occur; i.e., when the monitored channel power level is proper at the relevant input port of the ROADMs, each of the ROADMs transitions from “Reconf” to “Stable”. Similarly, the receiver transitions from “Wait” to “Stable” when a sufficient power level is detected. However, these transitions are triggered by physical light, not by a DR message.)

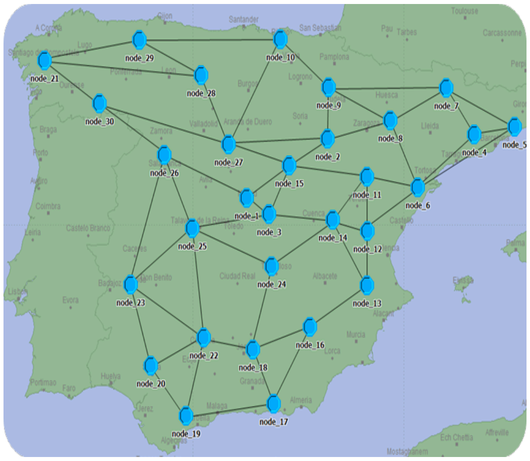


Fig. 5. Spanish network topology used in the simulations.

4. SIMULATION RESULTS

Delegated Restoration is assessed by simulations using a custom built event-driven C++ simulator. The Spanish backbone network topology used in the simulations (shown in Fig. 5) consists of 30 nodes and 56 bidirectional links. Lightpath requests follow a Poisson process. The average holding time of each lightpath is $1/\mu = 24$ hours, while the average inter-arrival time $1/\lambda$ is varied to change the traffic load, which is expressed as λ/μ. All requested lightpaths are established using 37.5 GHz bandwidth.

Path computation for the working path selects the path with the maximum number of frequency slots satisfying the continuity constraint among all of the shortest paths, in terms of hops, connecting the source-destination pair [27]. Spectrum assignment for the working lightpath is first fit (lowest ITU-T n). A lightpath request is blocked when no path with at least one frequency slot satisfying the continuity constraint is found.

If dynamic restoration is required, each time it is applied, the path with maximum number of frequency slots satisfying the continuity constraint among all of the shortest paths, in terms of hops in the new topology (i.e., removing the failed link from the original topology), is selected; first-fit spectrum assignment is applied.

Path computation for the delegated lightpath returns the shortest path, in terms of hops, that is link-disjoint from the working path. In the simulations, this computation is performed just once for a connection, when it is first established. Two spectrum assignment strategies were tested for the delegated lightpath: first fit (FF) and last fit (LF). The former selects the first available frequency slot (lowest ITU-T n), the latter selects the last available frequency slot (highest ITU-T n), both satisfying the continuity constraint. The unavailability of a delegated lightpath does not cause a connection request to be blocked.

In the simulations, different connections do not share delegated lightpaths. Moreover, if a new working lightpath uses resources of the delegated lightpath of another connection, the FSM associated with that delegated lightpath is assumed to be uninstalled and, if the connection fails, traditional dynamic restoration is attempted. In this case, an alarm is sent to the SDN controller, through a NETCONF `notification` message when a failure occurs; the SDN controller then performs RSA for the backup lightpath as stated above.

Delegated Restoration is compared with a fully centralized approach that always utilizes traditional dynamic restoration: when a link failure is detected, a NETCONF `notification` message is sent to the SDN controller to inform it about the failed lightpath; the SDN controller then performs RSA for the backup lightpath and enacts the necessary rerouting. RSA at the SDN controller requires a time T_{RSA}. When the SDN controller receives several NETCONF notification messages at one time, these messages experience a queue for processing which depends on T_{RSA}. Conversely, with Delegated Restoration, when a delegated lightpath is available, lightpaths are recovered without requesting any processing from the SDN controller during the failure.

Delegated and centralized restoration are compared in terms of: i) control plane recovery delay, defined as the time between the failure and the time retransmission on the backup path can begin (averaged over all of the link failures and all of the restored connections); ii) recovery blocking probability, defined as the ratio between the unrecovered lightpaths and the lightpaths impacted by the failure (averaged over all of the link failures). A lightpath experiences blocking during restoration when no path satisfying the frequency slot continuity constraint is found (i.e., the delegated lightpath is unavailable, and dynamic restoration fails).

Performance is measured through 1,000 experiments of 10,000 connection requests; in each experiment a single link failure is randomly generated among all network links with a uniform distribution among the links. The duration of each experiment, T, is 100 times longer than the average connection holding time. At time 0.01×T, the single link failure is generated; at time 0.04×T, the failed link is repaired. A non-revertive scheme is assumed, where the connection remains on the backup path even after the failure is repaired. Plots are reported with a confidence interval of 95 % (note that the confidence intervals were typically very small).

Fig. 6 shows the control plane recovery delay versus traffic load with $T_{RSA} = 20$ ms. This value of processing time has been selected since it is comparable with the time required by the Gaussian-Noise-model estimation tool to perform QoT computation [5]. The centralized solution experiences a higher control plane recovery delay as compared to Delegated Restoration because centralized restoration requests always rely on the SDN controller, thus they experience a processing queuing delay. Delegated Restoration reduces recovery delay because some, or all, of the failed lightpaths typically can use delegated backup lightpaths, without requesting any processing from the SDN controller; the controller is only involved when a delegated lightpath is unavailable. The availability of a delegated lightpath permits prompt reaction to failures because devices are already

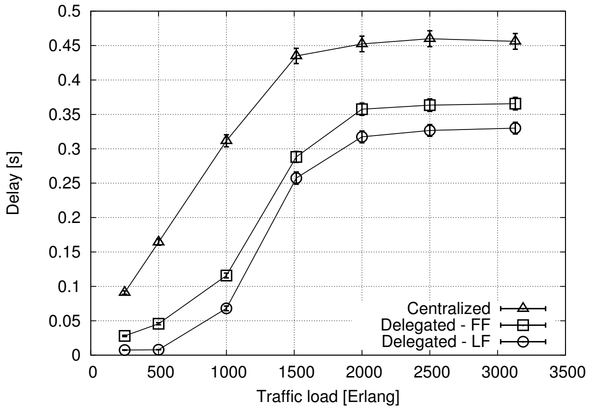


Fig. 6. Control plane recovery delay vs. traffic load with $T_{RSA} = 20$ ms.

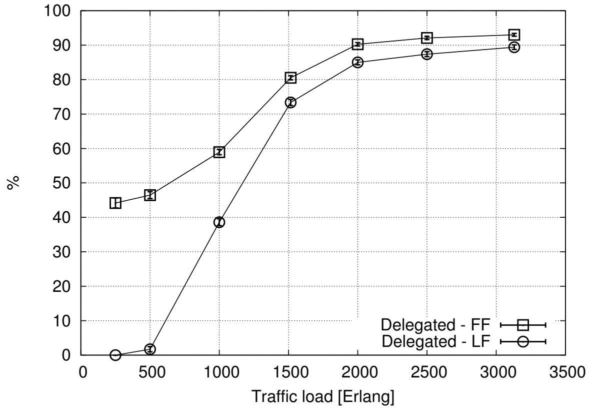


Fig. 7. Percentage of connections affected by the failure requiring SDN invocation when Delegated Restoration is adopted with $T_{RSA} = 20$ ms.

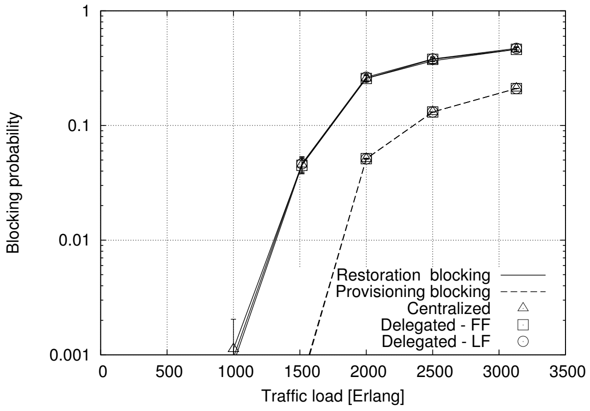


Fig. 8. Restoration and provisioning blocking probability vs. traffic load.

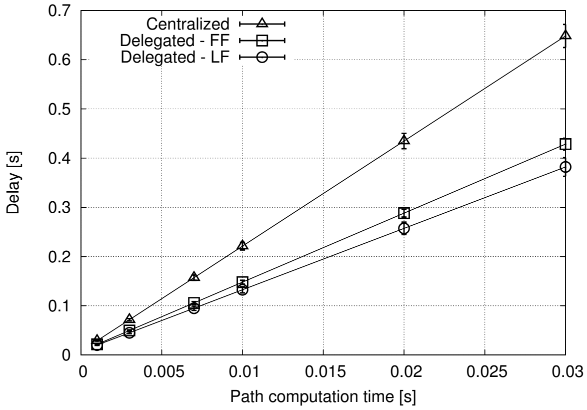


Fig. 9. Control plane recovery delay vs. T_{RSA} at 1515 Erlang.

instructed about the reconfigurations.

With the centralized approach, the control plane recovery delay increases with traffic load because more lightpaths are impacted by the failure; thus, more restoration requests have to be processed by the SDN controller and a longer queue is experienced. With Delegated Restoration, recovery delay increases with traffic load as well, albeit at a slower rate, because more working lightpaths use resources of other delegated lightpaths, making them unavailable; thus more connections need to send a request to the SDN controller to compute a backup lightpath upon failure.

As shown in Fig. 6, Delegated Restoration with LF reduces the recovery delay as compared to Delegated Restoration with FF. Indeed, since the working paths are provisioned with FF spectrum assignment, the adoption of LF for the delegated lightpath reduces the probability that a working lightpath uses the resources of a delegated lightpath (as shown in Fig. 7) since they are preferentially assigned from opposite ends of the spectrum. Fig. 7 shows the percentage of connections affected by the failure that rely on the central controller when Delegated Restoration is adopted with $T_{RSA} = 20$ ms. As anticipated, LF reduces the number of connections relying on the SDN controller after the failure as compared to FF because fewer working lightpaths use resources of delegated lightpaths.

Fig. 8 shows that Delegated Restoration and the centralized approach experience the same performance in terms of both provisioning and restoration blocking probability (the lines are on top of each other in the figure for the two schemes). Indeed, the RSA strategy for the working path is the same for both Delegated Restoration (either FF or LF) and the centralized approach, thus resulting in similar provisioning blocking probability. There may be some differences because both schemes are assumed to run in a non-revertive manner, where connections remain on their backup path even after a failure is repaired, and the backup paths may not be the same for the two schemes. The restoration blocking probability is similar as well because Delegated Restoration will resort to dynamic restoration if the delegated lightpath is unavailable at the time of failure. Thus, Delegated Restoration can speed up the recovery delay without impacting restoration blocking probability.

Fig. 9 shows the control plane recovery delay vs. T_{RSA} at 1515 Erlang, as a function of SDN path computation time. The centralized solution experiences a higher control plane recovery delay as T_{RSA} increases because the queuing time at the SDN controller increases. Delegated Restoration, which reduces the number of connections that rely on the SDN controller upon failure, is less impacted by the increase in T_{RSA}, though the control plane recovery does increase as well (i.e., at a slower rate).

The simulations assumed that a delegated lightpath is calculated just once for a connection; i.e., when the connection request is first received and established. When a working path uses a resource of the delegated lightpath of another connection, the connection makes use of dynamic restoration if impacted by the failure, thus increasing the recovery delay. To dampen this effect, it is possible to have the SDN periodically recalculate the delegated lightpath for a connection, to take into account the current network state. This can be investigated in future work.

5. CONCLUSIONS

In this paper, an innovative approach named *Delegated Restoration* has been investigated for SDN-based disaggregated networks operating with the NETCONF protocol. The SDN controller pre-computes restoration lightpaths and instructs the agents of the involved devices of the reconfigurations to apply in case of link failure. In this manner, the agents can react to failures by simply applying pre-loaded instructions without interrogating the SDN controller. For this reason, this method also overcomes the problem of SDN controller unavailability during a failure, which is an event currently being investigated by operators. NETCONF and YANG are used to install the instructions, which are given in the form of a finite state machine.

Delegated Restoration is scalable because it limits the involvement of the SDN controller during the failure and it achieves faster rerouting time with respect to a fully centralized solution. Moreover, Delegated Restoration is efficient in terms of spectrum utilization because it relies on centralized path computation. Rerouting is enacted in a distributed way within the network but without requiring any path computation intelligence distributed in the control plane or the allocation of extra resources (as in 1+1 protection). Simulations have shown the benefits of Delegated Restoration in terms of recovery time with respect to a fully centralized solution, highlighting its potential to impact recovery in next-generation networks.

There are several directions for future work to investigate; for example, Delegated Restoration with shared resources, or the implementation of Delegated Restoration in a network that requires regeneration. Future studies can also assume different service classes, associating a priority to the resources pre-computed for delegated lightpaths that can be impacted by the setup of new working lightpaths. Also, as noted above, the impact of periodically updating the calculated designated lightpath can be investigated.

6. ACKNOWLEDGEMENT

This work has been supported by the Italian Ministry for Education University and Research (MIUR) under the PRIN project FIRST (Fiber Infrastructure for Research on Space-Division Multiplexed Transmission).

REFERENCES

1. E. Riccardi, P. Gunning, O. G. de Dios, M. Quagliotti, V. Lopez, and A. Lord, “An operator view on the introduction of white boxes into optical networks,” J. Light. Technol. **36**, 3062–3072 (2018).

2. L. Velasco, A. Sgambelluri, R. Casellas, L. Gifre, J. Izquierdo-Zaragoza, F. Fresi, F. Paolucci, R. Martinez, and E. Riccardi, "Building autonomic optical whitebox-based networks," J. Light. Technol. **36**, 3097–3104 (2018).
3. N. Sambo, K. Christodoulopoulos, N. Argyris, P. Giardina, C. Delezoide, A. Sgambelluri, A. Kretsis, G. Kanakis, F. Fresi, G. Bernini, H. Avramopoulos, E. Varvarigos, and P. Castoldi, "Experimental demonstration of a fully disaggregated and automated white box comprised of different types of transponders and monitors," J. Light. Technol. **37**, 824–830 (2019).
4. D. King and A. Farrel, "A PCE-based architecture for application-based network operations," IETF RFC 7491.
5. M. Filer, M. Cantono, A. Ferrari, G. Grammel, G. Galimberti, and V. Curri, "Multi-vendor experimental validation of an open source QoT estimator for optical networks," J. Light. Technol. **36**, 3073–3082 (2018).
6. "http://www.openroadm.org," .
7. "http://www.openconfig.net," .
8. "https://telecominfraproject.com/," .
9. "https://www.ietf.org/," .
10. R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NETCONF)," IETF RFC 6241 (2011).
11. M. Bjorklund, "YANG - a data modeling language for the network configuration protocol (NETCONF)," IETF RFC 6020.
12. C. Delezoide, K. Christodoulopoulos, A. Kretsis, N. Argyris, G. Kanakis, A. Sgambelluri, N. Sambo, P. Giardina, G. Bernini, D. Roccato, A. Percelsi, R. Morro, H. Avramopoulos, E. Varvarigos, P. Castoldi, P. Layec, and S. Bigo, "Marginless operation of optical networks," J. Light. Technol. **37**, 1698–1705 (2019).
13. N. Sambo, P. Castoldi, G. Fioccola, F. Cugini, H. Song, and T. Zhou, "YANG model for finite state machine," draft-sambo-netmod-yang-fsm-05 (2019).
14. N. Sambo, K. Christodoulopoulos, N. Argyris, P. Giardina, C. Delezoide, D. Roccato, A. Percelsi, R. Morro, A. Sgambelluri, A. Kretsis, G. Kanakis, G. Bernini, E. A. Varvarigos, and P. Castoldi, "Field trial: Demonstrating automatic reconfiguration of optical networks based on finite state machine," J. Light. Technol. pp. 1–1 (2019).
15. R. Ramaswami and K. Sivarajan, "Optical networks: a practical perspective," Morgan Kaufmann (2002).
16. E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) architecture," RFC 3945 (2004). IETF.
17. L. Liu, T. Tsuritani, and I. Morita, "From GMPLS to PCE/GMPLS to OpenFlow: How much benefit can we get from the technical evolution of control plane in optical networks?" in *2012 14th International Conference on Transparent Optical Networks (ICTON),* (2012), pp. 1–4.
18. A. Giorgetti, S. Fazel, F. Paolucci, F. Cugini, and P. Castoldi, "Path protection with hierarchical PCE in GMPLS-based multi-domain WSONs," IEEE Commun. Lett. **17**, 1268–1271 (2013).
19. R. Munoz, R. Casellas, R. Vilalta, and R. Martinez, "Dynamic and adaptive control plane solutions for flexi-grid optical networks based on stateful PCE," J. Light. Technol. **32**, 2703–2715 (2014).
20. R. Martinez, R. Casellas, R. Vilalta, and R. Munoz, "GMPLS/PCE-controlled multi-flow optical transponders in elastic optical networks [invited]," IEEE/OSA J. Opt. Commun. Netw. **7**, B71–B80 (2015).
21. K. Rauschenbach, R. Hain, A. Jackson, J. Jacob, W. Leland, J. Lowry, W. Milliken, P. Pal, R. Ramanathan, C. Santivanez, I. Baldine, Shu Huang, and D. Wood, "Dynamic provisioning system for bandwidth-scalable core optical network," in *MILCOM 2009 - 2009 IEEE Military Communications Conference,* (2009), pp. 1–7.
22. N. Sambo, A. Sgambelluri, F. Cugini, A. D'Errico, and P. Castoldi, "Enabling locally automated reconfigurations in disaggregated networks," in *Proc. of OECC/PS,* (2019).
23. "https://www.lumentum.com/en/products/dci-transport-roadm-whitebox-graybox," .
24. "Draft revised G.694.1 version 1.3," Unpublished ITU-T Study Group 15, Question 6.
25. "Openroadm v2 device whitepaper," in *http://openroadm.org/download.html,,* (2018).
26. M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "Control and management of transponders with NETCONF and YANG," IEEE/OSA JOCN **9**, B43–B52 (2017).
27. N. Sambo, F. Cugini, G. Bottari, G. Bruno, P. Iovanna, and P. Castoldi, "Lightpath provisioning in wavelength switched optical networks with flexible grid," in *2011 37th European Conference and Exhibition on Optical Communication,* (2011), pp. 1–3.