Blockchain-anchored Disaggregated Optical Networks

Silvia Fichera, Andrea Sgambelluri, Francesco Paolucci,
Alessio Giorgetti, Nicola Sambo, Piero Castoldi, and Filippo Cugini

***Abstract*—Optical network disaggregation is attracting significant consensus to avoid vendor lock-in solutions. However, the presence of network controllers, nodes, and hardware/software components potentially provided by different entities and manufacturers may lead to remarkable responsibility issues in case of service level degradation.**

We propose the use of the blockchain technology to provide reliable and trusted accountability of events and interactions among disaggregated network elements. Three levels of interactions are specifically considered: *i)* among SDN controllers, *ii)* between each SDN controller and the underlying network nodes, and *iii)* within disaggregated network nodes.

The proposed solutions have been implemented and experimentally validated in a disaggregated network testbed. Results show the effectiveness of the method even in case of controversial service level degradation upon failure events. Results also show good scalability performance to retrieve/add/validate blocks recorded in the blockchain even in case of large optical network scenarios.

***Index Terms*—Blockchain, distributed ledger, DLT, software defined networking, SDN, Disaggregation, Multi-domain, White box, service level, SLA**

I. Introduction

DISAGGREGATED optical networks have gained significant attraction particularly in the context of metro/regional networks due to the potential CapEx savings enabled by white-boxes, which avoid vendor-locked solutions [1]–[4]. OpenROADM and OpenConfig are two relevant initiatives standardizing the disaggregation framework [5]–[8]. The former is a complete multi-source agreement covering both data and control aspects of optical metro networks; the latter focuses on the control of packet and optical devices. Both initiatives have reached a good level of maturity in the standardization of relevant YANG models and first compliant components are nowadays commercially available. However, network operators are still cautious in deploying disaggregation within their production networks. One of the main concerns is the responsibility management. Indeed, in traditional optical networks, the responsibility was fully attributed to the (single) vendor providing the data plane infrastructure. Whereas, in disaggregated optical networks, network elements provided by multiple vendors may coexist and interact each other. Thus, clear and trusted mechanism for the responsibility identification of actions, events (e.g., failures, service level degradation) is required.

Three different levels of interactions among network elements require to be taken into account in a disaggregated environment.

The first level refers to the interaction among network controllers. In multi-domain network environments, multiple Software Defined Networking (SDN) controllers provided by different entities may coexist [7]. Thus, timely and effective cooperation among controllers is required for multi-domain service provisioning. For example, inaccurate resource availability information shared by one controller during multi-domain provisioning or recovery may lead to setup delays and inefficient resource utilization.

The second level refers to the interaction between a network controller and the underlying network nodes, such as transponders and Reconfigurable Add/Drop Multiplexers (ROADMs), which may be provided by different entities. In this case, potential delays or inappropriate configurations from the controller or at network nodes may lead to service outage or transmission performance degradation. For this reason, trusted accountability of events and actions is needed to clearly identify potential causes of problems or delays.

The third level refers to the intra-node interactions, particularly between software components and the underlying hardware (e.g., NETCONF agents over bare metal hardware) or between hardware components (e.g., chassis and pluggables transceivers), which may be provided or maintained by different entities. Also in this case, delays or (mis)configuration at the software level or related to hardware malfunctioning need to be traced and reliably accounted to identify sources of service degradation.

In this work, we propose to leverage on the blockchain technology to support vendor neutrality and inter-operability in disaggregated optical networks, enabling trusted accountability of network events and actions with clear attribution of responsibilities.

The blockchain technology was introduced a decade ago to serve Bitcoin transactions without relying on a third party entity. Every transaction that occurs in the bitcoin economy is registered in a public, distributed ledger (i.e., distributed ledger technology - DLT), which is called the blockchain. Every transaction is checked against the blockchain to ensure that the same bitcoins have not been previously spent, thus eliminating the double-spending problem. So far, no solutions have been presented to guarantee trusted accountability of events and actions in the context of disaggregation and the

S. Fichera, A. Sgambelluri, N. Sambo, and P. Castoldi are with Scuola Superiore Sant'Anna, Pisa, Italy. E-mail: silvia.fichera@santannapisa.it
A. Giorgetti is with IEIIT, CNR, Pisa, Italy.
F. Paolucci and F. Cugini are with CNIT, Pisa, Italy
Manuscript received October 11, 2023.

blockchain technology has not being proposed so far to be applied to disaggregated optical networks.

In the last years, blockchain has been proposed for several use cases including storing and management of data. Within the communication technology world, blockchain has been proposed in Internet of Things (IoT) and SDN scenarios targeting, for example, sensors and fog nodes vulnerabilities mitigation [9], [10]. Most recently blockchain has been proposed also in the framework of Network Function Virtualization (NFV) orchestration [11], including multi-domain scenarios to avoid the need of a central validation authority [12]. Blockchain and DLT have been proposed also in optical networks scenarios mainly to guarantee Service Level Agreements (SLAs). In [13], a distributed ledger solution is integrated in a SDN-controlled optical network by adding the *consensus plane* dedicated to the blockchain ledger. For what concern SLA management, blockchain and distributed ledger have been applied with several flavours. The work in [14] proposed a mechanism based on smart contracts to automatically compensate SLA violations while the service is running. In [15], the authors proposed a framework to monitor the quality of service that leverages on blockchain to detect SLA violation and identify the responsible entity. Moreover, blockchain has been considered to regulate the inclusion of new network resources [16], enable spectrum trading between elastic virtual optical networks [17], or to guarantee secure access identification for 5G fronthaul [18]. In addition, recent works have addressed the main issue of scalability for distributed ledgers, by proposing lightweight and scalable ledger solutions and consensus protocols able to be deployed within devices and sensors with limited computational resources, such as in the Internet of Things scenario [19]–[22].

In this paper, we propose, implement, and validate blockchain-based mechanisms to guarantee responsibility management with trusted accountability to all aforementioned levels of interactions among network elements of disaggregated optical networks. Specifically, blockchain-based solutions are implemented following the general proposed scenario depicted in Fig. 1 and validated to: (1) regulate and ratify Optical Signal to Noise Ratio (OSNR) requirements across domains with the objective of guaranteeing provisioning with adequate Quality of Transmission (QoT) to transparent optical connections crossing multiple domains (e.g., access and metro) and handled by multiple SDN controllers (Sec. II); (2) address two controversial failure use cases in disaggregated networks involving both network nodes and controllers: *a)* a joint data and control plane responsibility and *b)* a delayed failure localization and recovery (Sec. III); (3) provide a reliable mechanism to account for intra-node events involving software agents and underlying hardware (Sec. IV).

Although this paper demonstrates that the blockchain technology can efficiently guarantee trusted responsibility management in disaggregated optical networks, it is important to highlight that the implementation of blockchain is not free from limitations. In particular, it may lead to additional implementation costs and it might lead to intensive use of processing resources compared to to other non-cryptographic solutions. The following aspects need to be carefully considered before

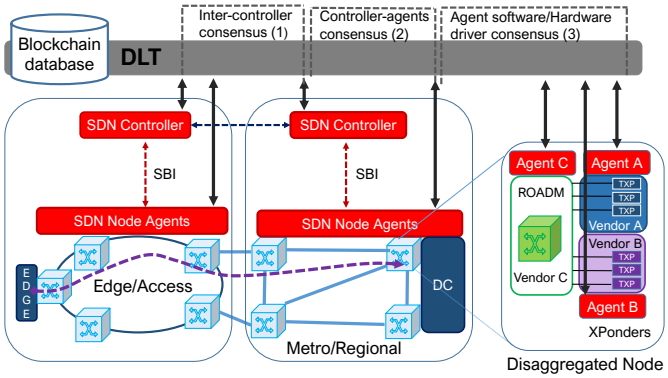

Fig. 1: Proposed blockchain-based plane in multi-controller disaggregated optical networks addressing controller-controller, controller-agent and intra-node agent-driver consensus and trusted SLA verification.

adopting the blockchain technology in deployed disaggregated optical networks:

1) The value of a common ledger of events covering the different components from different vendors (and possibly different operators).
   - The benefit is reliable QoT performance and validation of SLAs.
   - The cost is the requirement for common standards for recording events.
   - The risks are that the results may not be equally accessible to all parties and the results may be falsified after the event.
2) The value of distributing the common ledger.
   - The benefit is that this removes the risk of unequal access to the ledger.
   - The cost is that this distribution requires a trade-off between consistency across the distribution and the latency of access (e.g., CAP theorem).
3) The value of cryptographically assuring the distributed common ledger.
   - This removes the risk of falsification of event data after the event.
   - The cost is the introduction of further latency for the blockchain processing (see the experimental testing in Section V).

This work is an extended version of the contributions presented in [23], [24]. Besides the extended literature review and the definition of the overall architecture, this paper presents for the first time the third level of interaction applied to intra-node resources. That is, Section IV and V-D are new and not included in previous contributions.

II. Blockchain Consensus among SDN Controllers: QoT Agreement Use Cases

The proposed architecture is illustrated in Fig. 1 conceiving a DLT for three different consensus levels. The DLT choice is dictated by third-party visibility and applicability reasons, to allow each component to have equal access rights to a neutral

DLT resource space and to avoid unpractical deployments at network agents. All the consensus levels may co-exist in the same architectural framework. This section describes the first level of consensus, i.e., between SDN controllers, in the case of multi-domain multi-operator networks.

In multi-domain optical networks, lightpaths established transparently (i.e., without O-E-O regeneration at the domains edge) are subject to end-to-end QoT constraints. Each domain has to guarantee a certain agreed SLA that includes a set of requirements related to the offered optical signal quality (i.e., typically in terms of OSNR). In a SDN-based disaggregated optical network, such SLAs and QoT verification are typically in charge of the SDN controllers. The scenario depicted in Fig. 2 shows a multi-domain network where each domain offers the availability of a set of paths (e.g., for simplicity, primary and backup paths), each one described in terms of offered OSNR, that may be selected by the SDN controller. The paths include the last segment identifying the inter-domain link (i.e., the upstream domain is responsible for this segment).

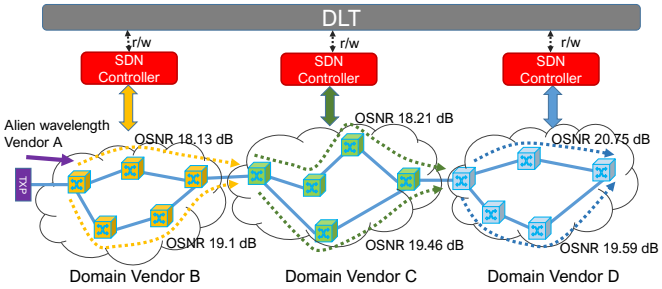

Fig. 2: Multi-domain optical network with blockchain-enabled SDN controllers exchanging per-domain OSNR values associated to a set of candidate paths (i.e., primary and backup path): alien wavelength and multi-vendor domains use cases.

Two different use cases are considered in this scenario. The former focuses on transparent lightpath establishment crossing multiple single vendor domains. The latter considers alien wavelength injection in several vendor domains. In the single vendor scenario, end-to-end physical layer performance (i.e., OSNR) has to be guaranteed. Since, typically, the inverse of OSNR cumulates linearly, the OSNR contribution of each domain has to satisfy a certain OSNR level and, most important, have to be agreed and guaranteed among the involved domains (i.e., the involved SDN controllers). Thus, the path OSNR value needs to be ratified by a consensus mechanism ensuring a fully trusted control environment. In a multi vendor scenario, the alien wavelength transmission (i.e., a wavelength transmitted by vendor A transceiver and injected in one or more vendor B ROADMs) introduces a further set of requirements with respect to single vendor. Similarly with respect to single vendor, the host domains are subject to QoT guarantees, in terms of offered OSNR. In addition, the alien wavelength transmission has to guarantee several physical parameters to be compliant with the host domains. As an example, the launch power needs to be adjusted before being injected. Host ingress nodes (i.e., ROADM add-drop stages) are able to provide equalization ensuring optimal power levels, provided that the launch power does not exceed given ranges. Uncertainty on the actual values of such parameters may raise issues during lightpath lifetime monitoring, preventing proper responsibility attributions and correct SLA verification.

To address both use cases, the SDN controllers of the involved domains are enriched with a distributed QoT value consensus mechanism encompassing the blockchain technology. The OSNR values related to each domain are computed and provided by the responsible controller for both the primary and the backup paths. The computation can be executed resorting to either estimation tools [25] or monitoring/telemetry mechanisms [26]. Each new OSNR value is stored and added by the responsible SDN controller as a novel block of the distributed blockchain to be validated,referred to as DLT in Fig. 2.

The DLT is a distributed database storing data blocks agreed among all the sharing SDN controllers. The considered DLT blocks include: a timestamp reporting the block validation time, the hash of the previous block, the validated values (e.g., estimated/computed OSNR) and the block hash (i.e., computed by combining all the block fields). Each new block is validated before being pushed in the blockchain. The validation requires to retrieve the hash of the previous block within the chain and to compute the hash of the current block.. The double check process ensures that the data written in the blockchain is immutable, not re-writable or subject to corruption by means of third parties. However, the mechanism is subject to scalability issues, since the validation time is dependent on the blockchain size. The DLT is interfaced by means of REST Application Protocol Interface (API). To add elements to DLT a REST API with a POST command is performed. In order to identify the blocks added to the DLT the same REST API with the GET command is utilized.

Since in multi-domain (multi-vendor or even multi-provider) scenarios business and administrative policies require that data related to logging, statistics, SLA verification, pricing profiles and forensic aspects have the consensus of all the involved partners to guarantee data integrity, full trustworthiness and offline verification, the added value of a blockchained cluster of SDN controllers may favour inter-operator and inter-vendor optical services and further business opportunities. Moreover, blockchain permits to certify the QoT experienced in each domain. This feature is extremely relevant in such a transparent multi-domain scenario to, firstly, verify that the experienced QoT is actually guaranteed as certified, then, to assess responsibilities (of a domain, thus of an operator or a vendor) in case some problem is experienced end-to-end.

III. Blockchain Consensus among Controllers and Network Nodes

The second level of consensus relies in the interaction between the network SDN controller and the network agents (i.e., simple device agent or complex disaggregated node controllers) through the Southbound Interface (SBI). Disaggregation allows SBI interaction between controllers and agents/nodes of different vendors, relying on standard YANG models, such as OpenConfig and OpenROADM. However, although

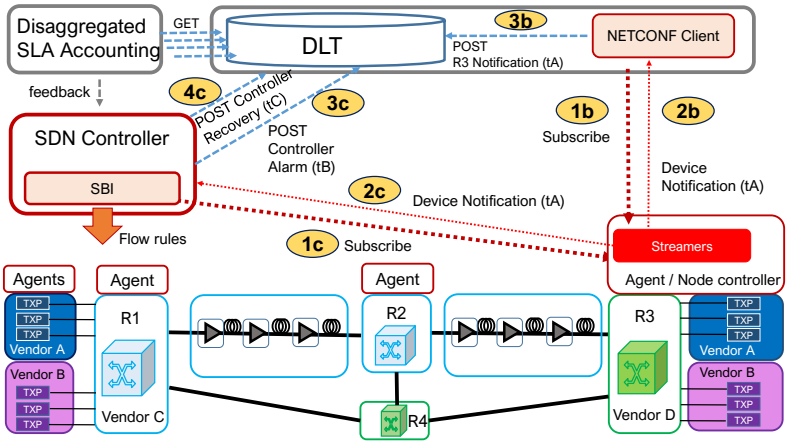

Fig. 3: Disaggregated SDN Scenario and proposed double layer blockchain notification assuring controller-agent SLA verification.

inter-operability at control plane level is guaranteed, full event notification awareness and SLA responsibility attribution may be inhibited by vendor-specific internal implementations and soft failures. For example, referring to the NETCONF protocol, flow entries enforced by the controller to the agent are confirmed typically by an ack message. The ack message specifies that the flow entry has been received and processed, however it does not guarantee that the flow entry has been activated effectively. Another critical example refers to event notifications subject to SLA evaluation. In the case of soft failures, notifications may be either not generated by the agent or not properly handled by the controller (e.g., delaying path recovery).

Fig. 3 illustrates the proposed solution: different vendors equipment are managed with software agents connected to the SDN controller. The network controller and each agent are allowed to post relevant events to the DLT. Specifically, after the establishment of a new lightpath, both the SDN controller and the DLT NETCONF client send a subscription to the agent R3 (step 1b and 1c, respectively). The involved agent, i.e., R3, creates a notification handler called streamer sending notifications to all subscribed entities once one of the specified alarms is detected. Thus, in case of alarm, the notification is sent to the controller (step 2c) and to the DLT NETCONF client *(step 2b)*. Both notifications contains the timestamp *tA* representing the time at which the alarm is detected by the agent. Once the notification is received, the DLT NETCONF client writes its content in the DLT (using a POST method, step 3b). Similarly, the controller uses a POST method to write in the DLT the time it receives the notification (step 3c). At this point, the controller triggers the recovery procedures. Once recovery is terminated, a final write event sent to the DLT to certify the recovery procedure with the related timestamp (step 4c).

All the asynchronous notifications generated by the agents and the controller encompasses the following data fields: a) the type of the event, b) the identifier of the generating entity, c) the event timestamp performed by the generating entity. It is worthwhile to note that different network events (e.g., agent low input/output power, agent port up/down, controller TED update, controller recovery procedure termination) are sent by different actors and stored in the blockchain for subsequent failure localization purposes. Such distributed certified logging procedure allows to detect possible SLA violations and the double timestamp mechanism is able to include both data and control plane events, thus extending the SLA verification also to controller/orchestrator layers.

A. *ONOS extensions to handle blockchain*

A novel module has been designed inside the ONOS controller to enable subscription to agent notifications. The module subscribes to specific notifications of controlled devices by opening a NETCONF session. When a notification is received the controller extracts the actual values from the XML payload (i.e., element-name, status, and element-type) and generates the POST message to the blockchain by enclosing a timestamp and the identifier of the specific device.

Among the considered events, the most critical is the ROADM line port state change (i.e., port-down). Upon notification arrival, the provided ONOS extension removes the affected link from the network topology thus automatically triggering the recovery process for all disrupted connections. Referring to Fig. 4, the SLA Accounting module periodically downloads the entire chain to process it. Specifically, the module is able to statistically understand which devices are responsible for a given set of events, are more reliable and are compliant with the subscribed SLA. The statistical analysis results are sent back to ONOS to run subsequent SLA-aware computations for future requests (e.g., excluding ROADMs experiencing excessive fault events).

The implementation distinguishes among three kinds of failure event notifications. The first event is related to a port up/down events (the element-type field is set to *port*) and includes the port name and its current status; this notification affects the ONOS behaviour as described above. The second event is related to anomalous low power level detection at the port (i.e., degraded port status); the third event is related to OpenROADM internal component (element-type field is

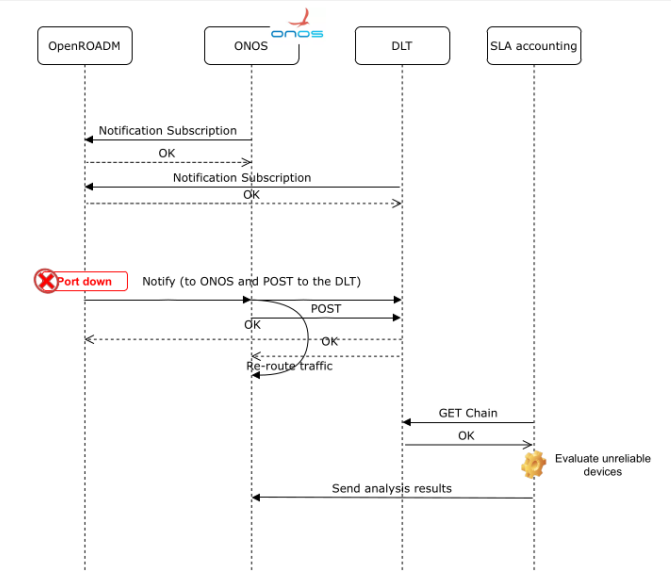

Fig. 4: Controller-Agent double notification: proposed full workflow upon a failure event.

set to *circuit–pack*). The last two notifications do not affect ONOS behavior directly since ONOS considers optical devices as black boxes. However, resorting to blockchain analysis, such notifications are of paramount importance to attribute responsibilities in case of malfunctioning affecting internal components.

IV. Blockchain within Disaggregated Network Nodes

In disaggregated scenarios, network nodes are typically implemented through a layered software structure over one or more hardware components as illustrated in Fig. 5. The upper software layer implements the network operating system of the node and includes on the agent providing the connectivity towards the SDN controller (e.g., using NETCONF and telemetry protocols). In addition, a local database is typically used to store configuration parameters and to manage locally generated data retrieved from the underlying hardware. The agents are substantially common for all the network elements of the same type. They can be based on open source solutions provided and maintained by open communities (e.g., Linux foundation, ONF). The lower software layer is the driver controlling the hardware resources. The driver is hardware-specific, and it is typically provided by the hardware manufacturer. The hardware components may be also provided by different manufacturers. For example, a transponder may be provided by a system integrator and it may include pluggable modules from different suppliers.

The interaction among software and hardware components provided by different entities may generate, in case of problems or unexpected latency, controversial attributions of responsibility. For example, delay may occur in the communication between NETCONF agent and hardware driver, or misconfigurations or inefficient alarm management between different components.

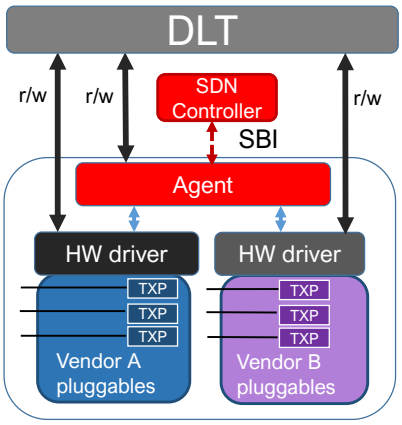

Fig. 5: Disaggregated node: software (agent) and hardware (proprietary driver) blockchain notifications.

When a NETCONF *edit-config* message is received by the node, the content of the command is stored within the local database for subsequent enforcement through the driver. Thus, the blockchain technology is here used to enable trusted responsibility management within the disaggregated node. Each event, such as reception of the edit-config message and each communication to the driver is stored in the blockchain. In particular, specifically designed REST clients have been implemented to provide the blockchain with timestamp information of each event at the different involved components.

This paper considers two types of optical nodes based on two different YANG models: OpenROADM-based (i.e., ROADMs to perform the optical cross-connection) and OpenConfig-based (i.e., the xPonder nodes, where the transponder/Muxponder cards are installed). Then, according to the disaggregated approach each physical component (i.e., filters and transponders) is controlled by a driver, enhanced with the mentioned blockchain NETCONF client. Considering the case of an OpenROADM-based node, we implemented a procedure to report in the blockchain register the main events happening in the node, as shown in Fig. 5. More specifically, two messages are generated by the NETCONF agent (Fig. 6) for the activation of an optical cross-connection: the first message is generated when the NETCONF agent receives the cross-connection edit-config from the NETCONF controller, the second message is generated when the driver is triggered. Then, the filter driver generates two messages: the first message is sent when the configuration command is received, while the second is generated when the filter configuration is completed (Fig. 7), working as acknowledgement.

In the case of an OpenConfig-based node, the same procedure has been implemented, in order to register the main events to the blockchain register. One message is generated by the NETCONF agent for each change in the NETCONF tree. Fig. 8 reports two examples of message generated by the OpenConfig NETCONF agent after receiving an edit-config from the NETCONF controller: the first message is generated when the wavelength is configured at the line-port of the card, while the second message is sent when the line-port of the

```
data{
  "Event": {
    "time": 2020:11:05T15:14:42.957,
    "message": "OR NETCONFAgent: Add connection command received"
  }
}
data{
  "Event": {
    "time": 2020:11:05T15:14:43.158,
    "message": "OR NETCONFAgent to the driver: adding connection from
                port 11 to port 10, frequency 192.3THz and band 50.0"
  }
}
```


Fig. 6: Messages generated by the OpenROADM agent.


```
data{
  "Event": {
    "time": 2020:11:05T15:14:43.344,
    "message": "DRIVER-WS: Received SET command"
  }
}
data{
  "Event": {
    "time": 2020:11:05T15:14:43.836,
    "message": "DRIVER-WS: configuration completed"
  }
}
```


Fig. 7: Messages generated by the filter driver.

card is enabled (i.e., admin-state set to enabled). Then, for each parameter to be configured the xPonder driver generates a couple of messages: the first one is sent once the SET command is received, the second one is generated when the configuration is completed (Fig. 9), in order to confirm the completion of the task.

V. Experimental validation

The proposed blockchain-enabled workflows have been evaluated in terms of SLA accountability and scalability in a multi-vendor network testbed depicted in Fig. 10, including control and data planes.

The control plane network is based on Gigabit Ethernet interfaces and reproduces a multi-domain control plane network including three different SDN ONOS controllers (C1, C2 and C3), version 2.2 with Optical Information Model and NETCONF SBI. Controller C1 (southbound IP address 10.30.2.95)


```
data{
  "Event": {
    "time": 2020:11:05T15:14:42.428,
    "message": "OC NETCONFAgent: Received the frequency configuration:
                tp=channel-11811, freq=192300000"
  }
}
```


Fig. 8: Message generated by the OpenConfig agent.


```
data
  "Event": {
    "time": 2020:11:05T15:14:42.828,
    "message": "DRIVER-SPO: Received SET frequency command"
  }
}
data{
  "Event": {
    "time": 2020:11:05T15:14:43.232,
    "message": "DRIVER-SPO: Frequency configured."
  }
}
```


Fig. 9: Messages generated by the transponder driver.

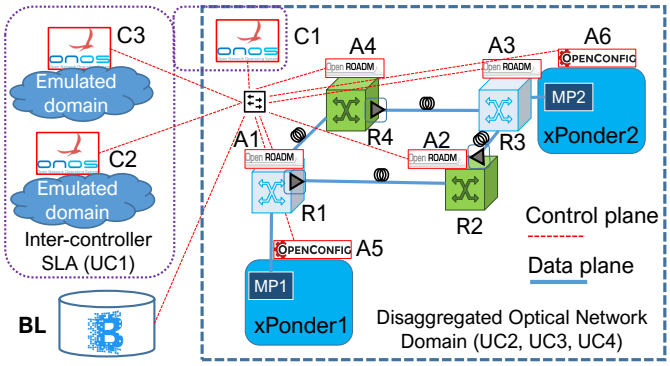

Fig. 10: Experimental data/control plane multi-domain optical disaggregated network testbed and use case location.

handles a real disaggregated optical network, while controllers C2 and C3 handle emulated domains. Moreover, a Blockchain Ledger (BL) is implemented in Javascript exposing a REST server able to process URL from the controllers and BL-collocated NETCONF client (IP address 193.205.83.72). In particular, the BL software implements an asynchronous runtime module specifically designed to support scalability and a crypto submodule specifically employed for hash computation. BL and C1, depicted in the figure as independent functional elements, run on the same physical Linux-based machine. Moreover, multiple YANG data models are supported by the controllers, in particular OpenROADM version 5.1.0 is considered for NETCONF notifications generated by ROADMs and OpenConfig version 1.0.0. The SLA Accounting Module is implemented as an internal ONOS module.

The data plane of the real domain is a multi-vendor disaggregated optical network composed by four ROADMs R1-R4 (R1, R3 of Vendor A; R2 and R4 of Vendor B), connected in a ring topology and two xPonders, each one hosting one muxponder (MP1 and MP2) of the same vendor (Vendor C). ROADMs are handled by A1-A4 OpenROADM agents, while xPonders are handled by OpenConfig Agents A5-A6. This way, the disaggregated network is multi-vendor and multi-YANG model.

Four use cases (UCs) have been considered and evaluated, controversial in terms of SLA accounting. UC1 refers to inter-domain controller SLA, described in Sec. II. The other UCs refer to both controller-agent (Sec. III) and disaggregated agent-agent interactions (Sec. IV) inside a single controller domain (C1), including a slow link failure recovery (UC2), a link failure delayed notification (UC3) and a disaggregated agent-agent interaction (UC4). UC1 and UC4 are used to evaluate the scalability in terms of BC performance, while UC2 and UC3 are used to evaluate the effectiveness of the proposed notification approach to detect the right component/vendor responsibilities during complex failure events affecting SLA.

A. Use case 1: inter-controller SLA

The first use case experiment considers a multi-domain scenario with three different controllers (C1, C2 and C3 of Fig. 10). Each controller send a bundle of information related to the signal power to the blockchain. The recorded

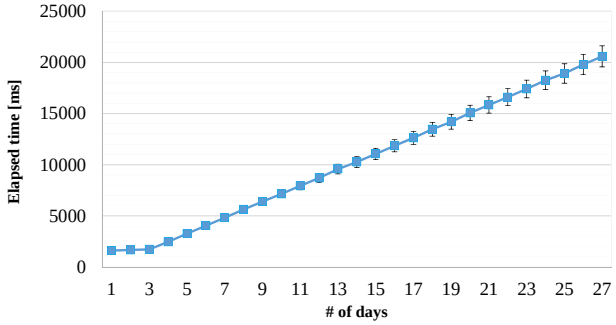

Fig. 11: UC1: controller-to-controller blockchain new block validation times.

information can be used to assess SLA violation responsibilities. Bundles are sent every 5 minutes. In Fig. 11 we show the time needed to add in the blockchain the last group of bundles (composed of the contribution provided by the three controllers) of each day in 27 days. This means that in day 1 the last group of bundles of the days takes 1.5s to be added in the chain. This time grows linearly and reach 20.5s at day 27. This is acceptable because the blockchain collects historical data (not real-time) that will be processed in case of a possible dispute between operators. To reduce this time, operators can also agree on how many days they need to collect into a single blockchain before archive it and start a new one.

In order to meet SLA responsibility requirements, blockchain scalability may be bounded easily. Indeed, unlike money-based business transactions, after a certain amount of time (i.e., years), a blockchain may be terminated. It is worthwhile to note that the blockchain achieved with the proposed approach refers to notifications successfully concluded without any issue. For example, after a major network upgrade, typically occurring in a 10 years average window due to key technological advances, device obsolescence and traffic growth, QoT blockchain may be saved for offline processing and replaced with new instances, thus guaranteeing the scalability of next-generation optical networks while maintaining high trustworthiness standards among different vendors/operators.

B. Use case 2: slow link recovery

In UC2, the full disaggregated single domain is considered, controlled by C1. A lightpath has been configured involving MP1 and MP2 along the route R1-R2-R3. A failure at ROADM R1 internal network component is induced. The full sequence of notifications is shown in the Wireshark capture of Fig. 12 (messages N1-N6). The failure affects link R1-R2 and triggers two port-down notifications N1 and N2 generated by A1 and A2, respectively, and A1 internal failure notification N3 (see the full XML notification expanded in Fig. 13). All these notifications sent by agents are processed by the NETCONF client (IP 193.205.83.72) co-located at the DLT. However, when notifications are received, the controller delays notifications processing and recovery due to internal software issues (e.g., CPU overloading). Thus, only after 5s, C1 deletes R1-R2 link in the TED, updates BL (N4-N6 notifications) and

	No.	Time	Source	Destination	Protocol	Length	Info
N1	3745	183.391930960	193.205.83.72	10.30.2.95	HTTP	454	POST
N2	3757	183.404331098	193.205.83.72	10.30.2.95	HTTP	467	POST
N3	3801	183.473054159	193.205.83.72	10.30.2.95	HTTP	454	POST
N4	3893	188.352350186	10.30.2.95	10.30.2.95	HTTP	252	POST
N5	3911	188.359431093	10.30.2.95	10.30.2.95	HTTP	265	POST
N6	3929	188.466807886	10.30.2.95	10.30.2.95	HTTP	265	POST
	No.	Time	Source	Destination	Protocol	Length	Info
N7	6548	143.015487244	10.30.2.95	10.30.2.95	HTTP	256	POST
N8	6585	143.085253324	193.205.83.72	10.30.2.95	HTTP	458	POST
N9	6854	148.157872829	10.30.2.95	10.30.2.95	HTTP	252	POST
N10	6861	148.157993223	10.30.2.95	10.30.2.95	HTTP	252	POST
N11	6894	148.166361190	193.205.83.72	10.30.2.95	HTTP	454	POST
N12	6902	148.167853841	10.30.2.95	10.30.2.95	HTTP	265	POST
N13	6903	148.167855392	10.30.2.95	10.30.2.95	HTTP	265	POST

Fig. 12: Wireshark captures at the Controller: POST messages notifications generated towards the Blockchain upon failure events (N1-N6 related to UC2, N7-N14 related to UC3).

```
('twin-wss', '2019-10-22T12:07:06.068557+00:00', 'circuit-pack', 'DOWN')
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2019-10-22T12:07:06.068557+00:00</eventTime>
  <element-change xmlns="http://org/openroadm/device">
    <element-type>circuit-pack</element-type>
    <element-name>twin-wss</element-name>
    <status>DOWN</status>
  </element-change>
</notification>
```

Fig. 13: Agent notification: circuit pack twin-wss failure (UC2).

triggers recovery to exclude link R1-R2. The SLA module, by inspecting and processing the blockchain entries and the related timestamps, assigns responsibility to node R1 due to N3 internal failure (no responsibility of R2) and to C1 due to excessive recovery response. In this case, SLA responsibilities are shared between the controller and one ROADM. This use case demonstrates the ability of the proposed notification mechanism to both identify the affected nodes and multiple SLA responsibilities during a failure event, that may affect different hardware/software components at the same time. Fig. 14 shows the BL database entry related to N3 notification (type circuit-pack affecting the internal Wavelength Selective Switch - WSS module). The ONOS notification bypass time is 0.1s, while POST messages to BL are pushed in less than 1 ms.

C. Use case 3: Link failure with delayed notification

In UC3, the same lightpath is installed between MP1 and MP2. A R1-R2 link failure event is first notified by A3 by


```
"timestamp": 1571746026126,
"lastHash": "a4d80c2724450d5d78674d0c45337d5d593fd36aaf22c1cabec839c209947421",
"hash": "7999d17190fe6ef3414165cae137983a208a2d1766fd9b0f91f7d6094302db15",
"data": {
  "Event": {
    "status": "DOWN",
    "time": "2019-10-22T12:07:06.068557+00:00",
    "type": "circuit-pack",
    "DeviceId": "netconf:10.100.100.6/2022",
    "element": "twin-wss"
```


Fig. 14: Blockchain ledger entry related to UC2: internal ROADM failure event (type circuit-pack at the WSS module).

detecting low optical power at R2-R3 component (N7, N8 notifications in Fig. 12). However, controller C1 is not able to trigger recovery since no port/link down events have been detected. After 5s C1 receives A2 port down notification on link R1-R2 and A2 internal failure (N9-N12 notifications). Thus, C1 immediately starts recovery (N13) with no controller issues. Thanks to the N8 event registration, the SLA module is able to identify R2 full responsibility due to internal failure and delayed notifications inducing anomalous failure localization and delayed recovery, while no SLA responsibilities are issued to R1, R3 and C1. This use case highlights the importance of a proper SLA responsibility attribution. In this specific use case the controller might be identified as a candidate responsible for the delayed recovery, whereas the double notification mechanism allows each workflow segment timestamp to be stored in the blockchain.

D. Use case 4: disaggregated optical network

The last use case is related to the scenario described in Sec. IV. In the considered use case the two OpenConfig-based agents A5 and A6 and one OpenROADM agent R3 are involved. A single lightpath between MP1 and MP2 crossing R3 is considered and mapped in the ONOS controller as a single intent. The enforcement of a single intent generates 16 different registration events stored in the blockchain. To evaluate the disaggregated scenario scalability, we have measured the time needed to store the 16-events bulk as a function of the number of intents already installed in the network (i.e., as a function of the blockchain size). Fig. 15 shows that the data related to a single intent takes 1500 ms to be recorded in the Blockchain. This value is practically constant until 700 intents, after which the time needed to record and validates the block related to the intent increases linearly, due to the blockchain size and to the increased number of hash operations required to validate the new block.

Such plot suggests that, for a significant number of fully disaggregated intents the blockchain computation time is below 3s, which is an acceptable value even for online failure detection purposes. However, again, blockchain is utilized in this use case mainly to address multi-vendor SLA accountability, so it may be inspected offline. In addition, even if the optical network is big and lightpaths lengths (and hops) may be higher, blockchain computation times in the order of hundreds of seconds (e.g., few minutes) may be considered acceptable, given that optical connections are quasi-static and are typically established with long duration times (i.e., days, months). This demonstrates the feasibility of the full blockchain-certified notification mechanism also in the disaggregated scenario. Further scalability optimization is achievable by implementing lightweight DLT computation and consensus mechanisms specifically designed for devices with limited computational capabilities (e.g., as in IoT) [19]–[22].

VI. Conclusion

We proposed the use of blockchain to ratify QoT performance and validate Service Level Agreements in different SDN-controlled optical networks scenarios: in multi-controller networks supporting alien wavelengths, where several business actors are involved in the process of service provisioning, and in disaggregated optical networks, where multi-vendor interoperability requires full visibility of software and hardware component responsibility. The proposed architecture, conceiving a close relationship between DLT and SDN controller, represents a tradeoff between a fully centralized blockchain (unpractical for third party trustworthiness in multi-domain scenarios) and fully distributed blockchain (unfeasible for scalability reasons when realized at each agent and hardware driver in the disaggregated scenario). ONOS Controller extensions were proposed to enable augmented Blockchain-enabled SLA awareness framework in disaggregated optical networks. Experimental demonstration results obtained resorting to OpenROADM and OpenConfig models successfully showed that SLA accountability due to controversial component failure events is improved and evaluated the blockchain applicability in terms of chain size and new block validation times. In conclusion, the use of blockchain is an answer to open, multi-operator and multi-vendor optical network architectures which has trade-offs between cost/performance and security. The paper has quantified the cost/performance and the improved SLA accountability. It is left to network operators to evaluate the trade-off with their trustworthiness, accountability and security needs.

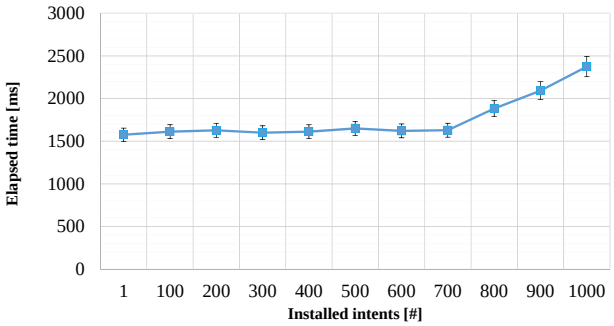

Fig. 15: UC4: time needed to record all the events related to the n-th intent.

Acknowledgment

This project has received funding from the ECSEL Joint Undertaking (JU) BRAINE Project under grant agreement No 876967. The JU receives support from the European Union's H2020 research and innovation programme and from Italy Ministry of Education, University and Research (MIUR).

References

[1] E. Riccardi, P. Gunning, Óscar González de Dios, M. Quagliotti, V. López, and A. Lord, "An operator view on the introduction of white boxes into optical networks," *J. Lightwave Technol.*, vol. 36, no. 15, pp. 3062–3072, Aug 2018.

[2] A. Giorgetti, A. Sgambelluri, R. Casellas, R. Morro, A. Campanella, and P. Castoldi, "Control of open and disaggregated transport networks using the open network operating system (onos) [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, no. 2, pp. A171–A181, 2020.

[3] J. Kundrát, O. Havliš, J. Jedlinský, and J. Vojtěch, "Opening up roadms: Let us build a disaggregated open optical line system," *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4041–4051, 2019.

[4] V. Lopez, W. Ishida, A. Mayoral, T. Tanaka, O. Gonzalez de Dios, and J. P. Fernandez-Palacios, "Enabling fully programmable transponder white boxes [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, no. 2, pp. A214–A223, 2020.

[5] M. Birk, O. Renais, G. Lambert, C. Betoule, G. Thouenon, A. Triki, D. Bhardwaj, S. Vachhani, N. Padi, and S. Tse, "The openroadm initiative [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, no. 6, pp. C58–C67, 2020.

[6] F. Paolucci, R. Emmerich, A. Eira, N. Costa, J. Pedro, P. W. Berenguer, C. Schubert, J. K. Fischer, F. Fresi, A. Sgambelluri, and F. Cugini, "Disaggregated edge-enabled c+l-band filterless metro networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, no. 3, pp. 2–12, 2020.

[7] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Abstraction and control of multi-domain disaggregated optical networks with openroadm device models," *Journal of Lightwave Technology*, vol. 38, no. 9, pp. 2606–2615, 2020.

[8] A. Sgambelluri, J.-L. Izquierdo-Zaragoza, A. Giorgetti, L. Gifre, L. Velasco, F. Paolucci, N. Sambo, F. Fresi, P. Castoldi, A. C. Piat *et al.*, "Fully disaggregated roadm white box with netconf/yang control, telemetry, and machine learning-based monitoring," in *Optical Fiber Communication Conference*. Optical Society of America, 2018, pp. Tu3D–12.

[9] P. K. Sharma, M. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for iot," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[10] C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing sdn security for iot-related deployments through blockchain," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2017, pp. 303–308.

[11] P. Alemany, R. Vilalta, R. Munoz, R. Casellas, and R. Martinez, "Peer-to-Peer Blockchain-based NFV Service Platform for End-to-End Network Slice Orchestration Across Multiple NFVI Domains," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 151–156.

[12] P. Alemany, R. Vilalta, R. Munoz, R. Martinez, and R. Casellas, "Managing Network Slicing Resources Using Blockchain in a Multi-Domain Software Defined Optical Network Scenario," in *2020 European Conference on Optical Communications (ECOC)*, 2020, pp. We1K–3.

[13] H. Yang, Y. Liang, Q. Yao, S. Guo, A. Yu, and J. Zhang, "Blockchain-based secure distributed control for software defined optical networking," *China Communications*, vol. 16, no. 6, pp. 42–54, 2019.

[14] E. J. Scheid, B. B. Rodrigues, L. Z. Granville, and B. Stiller, "Enabling dynamic sla compensation using blockchain-based smart contracts," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 53–61.

[15] R. Ranchal and O. Choudhury, "Slam: A framework for sla management in multicloud ecosystem using blockchain," in *2020 IEEE Cloud Summit*, 2020, pp. 33–38.

[16] S. Kou, H. Yang, H. Zheng, W. Bai, J. Zhang, and Y. Wu, "Blockchain mechanism based on enhancing consensus for trusted optical networks," in *2017 Asia Communications and Photonics Conference (ACP)*, Nov 2017, pp. 1–3.

[17] S. Ding, G. Shen, K. X. Pan, S. K. Bose, Q. Zhang, and B. Mukherjee, "Blockchain-assisted spectrum trading between elastic virtual optical networks," *IEEE Network*, pp. 1–7, 2020.

[18] H. Yang, Y. Wu, J. Zhang, H. Zheng, Y. Ji, and Y. Lee, "Blockonet: Blockchain-based trusted cloud radio over optical fiber network for 5g fronthaul," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, March 2018, pp. 1–3.

[19] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[20] S. Biswas, K. Sharif, F. Li, B. Nour, and Y. Wang, "A scalable blockchain framework for secure transactions in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4650–4659, 2019.

[21] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A lightweight consensus algorithm for scalable IoT business blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2020.

[22] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: Towards a scalable public blockchain in fog computing of IoT services computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252–262, 2020.

[23] S. Fichera, N. Sambo, F. Paolucci, F. Cugini, and P. Castoldi, "Leveraging blockchain to ratify QoT performance in multi-domain optical networks," in *45th European Conference on Optical Communication (ECOC 2019)*, 2019.

[24] S. Fichera, A. Sgambelluri, A. Giorgetti, F. Cugini, and F. Paolucci, "Blockchain-anchored failure responsibility management in disaggregated optical networks," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, 2020.

[25] 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," *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3073–3082, Aug 2018.

[26] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, "Network telemetry streaming services in sdn-based disaggregated optical networks," *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3142–3149, Aug 2018.