Blockchain-Based Connectivity Provisioning in Multiple Transport SDN Domains

Pol Alemany, Ricard Vilalta, Raul Muñoz, Ramon Casellas, Ricardo Martínez
Optical Networks and Systems Dept.
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Catelldefels (Barcelona), Spain
pol.alemany@cttc.cat

***Abstract*—Hierarchical Software Define Networking (SDN) architectures is used to manage the co-existence of multiple domains by having an element on top. A collaborative relationship between domains, might solve this issue. Blockchain may become the key element for this change to happen. This paper presents a Blockchain-based architecture to provide SDN actions to configure connectivity services in transport domains. The results presented show that the use of Blockchain is a promising candidate for inter-domain SDN control.**

***Index Terms*—Software-Defined Networks, Blockchain, T-API**

I. Introduction

Software-Defined Networking (SDN) manages a set of network resources by using software applications, to differentiate the data plane from the control plane and many other advantages. Some solutions propose a centralized management using a hierarchical architecture [1], with a single element on top that has the End-to-End (E2E) knowledge of the network infrastructure. While this centralisation improves control actions, it also makes the whole system have an important weakness as the top element becomes the center of any action. So, if the element on top becomes unavailable, a central point of failure situation may appear. But the most important issue is the fact that this type of centralised architectures cannot be deployed in multi-operator architectures. To avoid this issues, a solution is the use of distributed control systems where the different transport SDN domains work together on the E2E network control instead of depending on a single top element.

Distributed Ledger Technologies (DLT), with Blockchain (BL) [2] being its most known example, allows a set of nodes to become a distributed database by sharing information in a public and transparent way and accepting the data to be stored using a consensus mechanism. Furthermore, BL is not only a distributed data base, it also allows the execution of small programs (i.e., smart contracts) allowing the peers to work together without the need of a central element commanding any action. BL is being used in multiple research SDN paths, such as security aspects on the flows management [3] or the recovery of SDN nodes after a failure [4].

This paper presents a BL-based SDN architecture to allow different transport domains to collaborate among them and avoid the dependence on an E2E SDN controller on the top. Compared to the work presented in [6] [7], this paper is based on the idea to have a collaborative SDN architecture, and with respect to [8], the current work presents how a set of SDN domains interact among them to compute and deploy E2E transport connections. This paper presents a set of experimental results to prove the workflows and the BL influence in the E2E transport connections services creation.

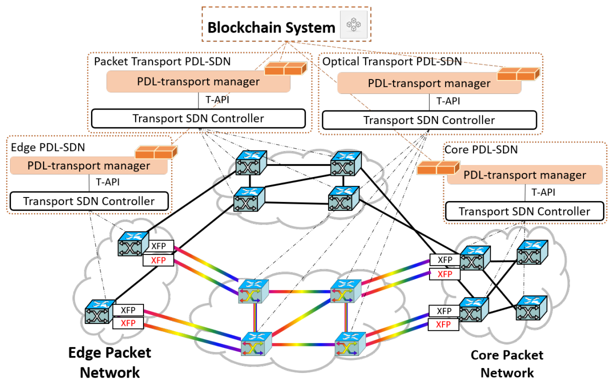


Fig. 1. SDN Transport BL-based Architecture on the ADRENALINE testbed.

II. A Blockchain Transport SDN Architecture

In order to let SDN controllers join the BL system as easy as possible, the designed architecture makes use of a new module that allows an SDN Transport controller to become a BL peer (i.e., PDL-Transport Manager). Fig.1 shows multiple SDN domains as peers of a BL system. Each peer (called PDL-SDN) is composed by the SDN Controller and the PDL-Transport Manager module. This new module takes assigned events from the BL requests (called transactions) generated by other peers, and maps them into ONF Transport API (T-API) [5] requests for the underlying Transport SDN Controller. The use of T-API was selected as it allows the deployment of per-domain Connectivity Services (CS) to configure an E2E transport connection (from now on E2E CS). A T-API CS request allows the configuration of transport connections between a pair of client ports known as Service Interface Points (SIPs) of an SDN transport domain.

The use of BL brings the following advantages: a) any request for networking resources is public, transparent and immutable once done, which makes it highly difficult to tamper it, b) the avoidance of a hierarchical E2E architecture and so, there is no central point of failure that may block E2E

978-3-903176-33-1 © 2021 IFIP

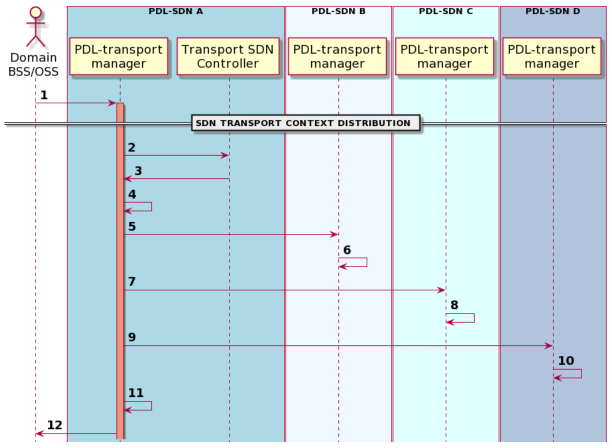


Fig. 2. Transport SDN context and topology distribution.

actions, c) when a peer joins the BL network, its information is dynamically added and the other peers update their vision of the whole infrastructure, d) as there is no hierarchy and all the peers are equally important, if a peer becomes unavailable, the others can still work together and, finally, e) the way the architecture is designed, only one Transport SDN domain is able to configure each domain CS as its creation request is linked to a unique BL address identifier. On the other hand, the current architecture has some drawbacks: a) a domain not being available may be included in a path computation because the domain computing the path does not have updated infrastructure information, b) BL is designed to avoid unfinished transactions, to do so: makes use of an associated cost per transaction, which means that any transaction must be generated with precision and the security to be accomplished, c) due to the use of costs, the information in a transaction must be as precise as possible and avoid possible redundancies. Despite their importance, these drawbacks can be solved by checking each domain availability once the path is computed or by applying improving the requests design in order to select the essential information.

A. SDN Context and Topology distribution

Before any CS can be requested, when a Transport SDN Controller domain joins the BL network, it must distribute its SDN T-API context to the other BL peers. The minimum information in a T-API context is a set of SIPs which are used by an optical SDN controller to request CSs. Furthermore, a T-API context may also define the topology of the network domain with the real or an abstract vision. In both cases the topology is defined by nodes and links. Nodes include a set of node ports called Node Edge Points (NEPs), and the links are defined by a pair of NEPs. So, a SIP is associated to a NEP at the edge of a network domain.

As presented in Fig.2, the process begins when the Domain Operations and Business Support Systems (OSS/BSS) specifies (1) to the PDL-transport manager the context to share. The PDL-transport manager gets (2) the context information and with the response (3) from the Transport SDN Controller, it selects (4) the necessary parameters for the BL. Then, it starts a transaction to distribute the information (5,7,9) and each do-

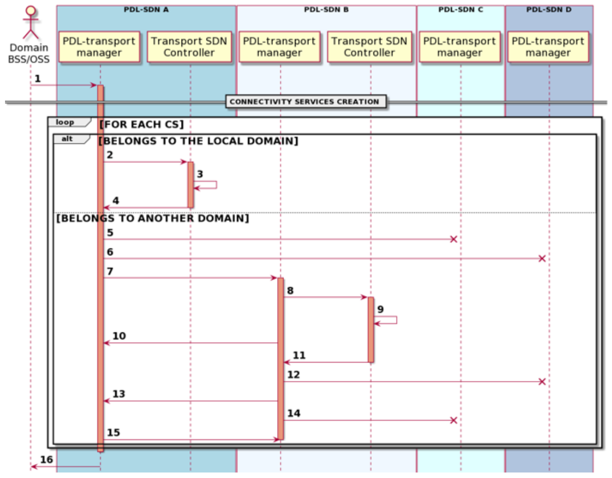


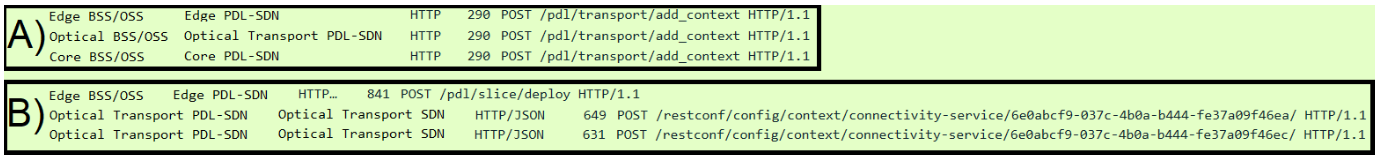
Fig. 3. CS deployment

main updates their vision of the whole infrastructure (6,8,10). Finally, the original requester waits for the transactions to be accepted (11) and the PDL-transport manager to inform about the correct context distribution (12).

To be aware of the E2E topology, each SDN domain has a graph generated with the shared information. Each SDN domain is mapped as a node and the edge NEPs are related a local SIP and a remote SIP from another transport SDN domain. For each E2E CS requested, a domain CS list is computed using domain selection.

B. SDN Connectivity Services deployment

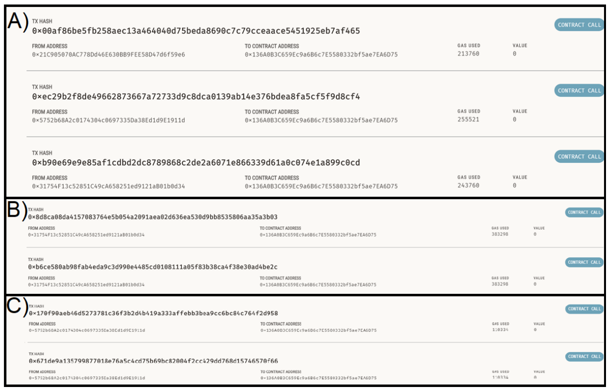
Fig.3 presents the procedure to create a BL-based E2E CS across transport SDN domains. The process starts when an E2E CS is requested by a domain OSS/BSS to its PDL-transport manager (1). A set of domain CSs is computed based on the graph generated when the transport SDN domains join the BL network. Currently, to select the domains involved in the E2E CS, a shortest path algorithm is used. So, starting from one extreme of the found path and checking one by one the involved SDN domains, the PDL-transport manager uses the shared context information to select the right pair of SIPs (i.e., source and destination) and to request each one of the domain CSs by distributing their information in the BL. If the domain CS must be done in the local domain, the request is forwarded (2) to the local Transport SDN Controller which configures the domain CS (3) and informs back (4). If, on the other hand, the domain CS must be done in another SDN domain, the PDL-transport manager generates a transaction and distributes it (5,6,7) specifying the peer address in charge. The specified peer is the only one able to take it and forward it to its local Transport SDN Controller (8) and notify its acceptance (10). Meanwhile, the domain CS is configured (9) and its information sent back (11) to its domain PDL-transport manager. Then, the updated domain CS information is distributed through a new transaction (12,13,14) with the peer address that originally requested the domain CS. Once the transaction is distributed and accepted (15) and if all the domain CSs are ready, the PDL-transport manager informs the OSS/BSS about the complete configuration of the E2E CS.



```
A) Edge BSS/OSS      Edge PDL-SDN                HTTP    290  POST /pdl/transport/add_context HTTP/1.1
   Optical BSS/OSS   Optical Transport PDL-SDN   HTTP    290  POST /pdl/transport/add_context HTTP/1.1
   Core BSS/OSS      Core PDL-SDN                HTTP    290  POST /pdl/transport/add_context HTTP/1.1
```

```
B) Edge BSS/OSS    Edge PDL-SDN    HTTP...    841  POST /pdl/slice/deploy HTTP/1.1
   Optical Transport PDL-SDN   Optical Transport SDN   HTTP/JSON    649  POST /restconf/config/context/connectivity-service/6e0abcf9-037c-4b0a-b444-fe37a09f46ea/ HTTP/1.1
   Optical Transport PDL-SDN   Optical Transport SDN   HTTP/JSON    631  POST /restconf/config/context/connectivity-service/6e0abcf9-037c-4b0a-b444-fe37a09f46ec/ HTTP/1.1
```

Fig. 4. HTTP requests to order the distribution of the context and the CSs deployment.



```
A) TX HASH
   0x00af86be5fb258aec13a464040d75beda8690c7c79cceaace5451925eb7af465            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x21C905070AC778Dd46E630BB9FEE58D47d6f59e6  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    213760     0

   TX HASH
   0xec29b2f8de49662873667a72733d9c8dca0139ab14e376bdea8fa5cf5f9d8cf4            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x5752b68A2c0174304c0697335Da38Ed1d9E1911d  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    255521     0

   TX HASH
   0xb90e69e9e85af1cdbd2dc8789868c2de2a6071e866339d61a0c074e1a899c0cd            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x31754F13c52851C49cA658251ed9121aB01b0d34  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    243760     0

B) TX HASH
   0x8d8ca08da4157083764e5b054a2091aea02d636ea530d9bb8535806aa35a3b03            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x31754F13c52851C49cA658251ed9121aB01b0d34  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    383298     0

   TX HASH
   0xb6ce580ab98fab4eda9c3d990e4485cd0108111a05f83b38ca4f38e30ad4be2c            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x31754F13c52851C49cA658251ed9121aB01b0d34  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    383298     0

C) TX HASH
   0x170f90aeb46d5273781c36f3b2d4b419a333affebb3bea9cc6bc84c764f2d958            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x5752b68A2c0174304c0697335Da38Ed1d9E1911d  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    110334     0

   TX HASH
   0x671de9a135799877018e76a5c4cd75b69bc82004f2cc429dd768d15746570f66            CONTRACT CALL
   FROM ADDRESS                                TO CONTRACT ADDRESS                           GAS USED   VALUE
   0x5752b68A2c0174304c0697335Da38Ed1d9E1911d  0x136A0B3C659Ec9a6B6c7E5580332bf5ae7EA6D75    110334     0
```

Fig. 5. BL transactions Log.

III. Experimental Validation

The proposed solution has been implemented as presented in Fig.1 using the ADRENALINE [9] testbed. With four different SDN domains; an edge, a transport and a core packet-based domains and an optical-based transport domain. All of them with an SDN controller managing the incoming domain CSs requests. Finally, over each domain SDN controller, there is the PDL-Transport Manager described in section II. To validate the proposed solution, an experimental evaluation was done by sharing the context of three different domains (edge, optical and core) and requesting a bidirectional CS (i.e., two unidirectional CSs) between the edge and core domains. Finally, the Blockchain system is implemented using an Ethereum emulator called Ganache.

Fig.4 and Fig.5 show the HTTP requests and the transactions generated among the BL peers in the set up and deployment procedures previously described in section II. On the one hand, regarding the workflow to distribute the context of each SDN domain, Fig.4-A shows the three HTTP requests to share the context of each SDN domain. Once each HTTP request reaches the corresponding PDL-transport manager, a BL transaction is generated and distributed as demonstrated in Fig.5-A. On the other hand, regarding the deployment of CSs, Fig.4-B shows the HTTP request that triggers the whole process (i.e., step 1 in Fig.3) and the HTTP requests to create the different CSs (as the step 8 in Fig.3) between the PDL-transport manager and its associated Transport SDN Controller. The distribution of the CSs requests across the BL network is presented with the two pairs of transactions log in Fig.5-B/C. Each transaction log belongs to the CSs creation transaction distribution (steps 5/6/7 in Fig.3) and the CSs update transaction distribution (steps 12/13/14 in Fig.3).

From the multiple tests done, a set of measures were obtained. Table I presents the mean and standard deviation values for each E2E CS deployment, their total deployment time and

TABLE I
CS DEPLOYMENT TIME VS. RELATED BL TRANSACTIONS TIME.

	Time (s)			
	Networking Deployment			**Blockchain Transactions**
	CS 1	**CS 2**	**Total**	
Mean Value	2.01	2.07	4.08	2.34
Std. Dev.	0.11	0.22	0.19	0.49

the total time associated to the different BL transactions (i.e., CS creation and update). A complete E2E CS configuration requires around 2 s, giving a total mean time value of 4.08 s to create the two unidirectional domain CSs (CS1 and CS2 in Table I). The BL transaction mean time value is of 2.34 s, which adds an increment of a 50 %. Despite this time increment is significant, compared to possible SDN situations such as a reconfiguration of optical amplifiers that may take minutes, it becomes less significant. Based on this, the trade-off to implement this new architecture might be an increment of seconds per each CS creation to keep the BL advantages.

IV. Conclusions

This paper has presented a BL-based architecture to manage transport SDN E2E CSs across transport networks. Moreover, the workflows to share SDN contexts and to create per-domain CSs were described. Finally, a set of experimental results were discussed presenting the influence of the BL actions on the whole process to create SDN CSs.

Acknowledgment

Spanish AURORAS (RTI2018-099178-B-I00) and Thematic Network Go2Edge (RED2018-102585-T), and EC H2020 TeraFlow (101015857) projects.

References

[1] R. Vilalta et al., *Hierarchical SDN orchestration for multi-technology multi-domain networks with hierarchical ABNO,* ECOC, 2015.

[2] S.S. Sachin et al., *Blockchain for Distributed Systems Security,* Wiley-IEEE Computer Society, 2019.

[3] S. Boukria et al., *BCFR: Blockchain-based Controller Against False Flow Rule Injection in SDN,* 2019 IEEE ISCC, 2019.

[4] S. Misra et al., *Blockchain-Based Controller Recovery in SDN,* IEEE INFOCOM, 2020.

[5] V. Lopez et al., *Transport API: A Solution for SDN in Carriers Networks,* ECOC, 2016.

[6] P. Alemany, et al., *Peer-to-Peer Blockchain-based NFV Service Platform for End-to-End Network Slice Orchestration Across Multiple NFVI Domains,* IEEE 5GWF, 2020.

[7] P. Alemany, et al., *Managing Network Slicing Resources Using Blockchain in a Multi-Domain Software Defined Optical Network Scenario,* ECOC, 2020.

[8] P. Alemany, et al., *End-to-End Network Slice Stitching using Blockchain-based Peer-to-Peer Network Slice Managers and Transport SDN Controllers,* accepted in OFC, 2021.

[9] R. Muñoz, et al., *The ADRENALINE Testbed: An SDN/NFV Packet/Optical Transport Network and Edge/Core Cloud Platform for End-to-End 5G and IoT Services* , EUCNC, 2017.