DLT-based End-to-end Inter-domain Transport Network Slice with SLA Management Using Cloud-based SDN Controllers

(Demo Session)

Lluis Gifre Renom*, Ricard Vilalta*, Sebastien Andreina†, Min Xie‡, Jane Frances Pajo‡, Konstantin Munichev†, Håkon Lønsethagen‡, Stanislav Lange§, Thomas Zinner§,Giorgia Marson†, Pol Alemany*, Marija Gajic§, Ramon Casellas*, Ricardo Martínez*, Raul Muñoz*

*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Spain
†NEC Laboratories Europe, Germany
‡Telenor Research, Norway
§ NTNU, Norway

***Abstract*—This paper presents an inter-domain transport network slice management with Service Level Agreements (SLA) using the ETSI TeraFlowSDN (TFS) controller. Different instances of the TFS controller are deployed for each involved domain. The communication between the TFS instances is supported by a Distributed Ledger Technology (DLT)–based database. The different TFS instances upload the abstracted view of their topologies and retrieve that from remote peers. When the end-to-end SLA of the transport network slice is violated, the slice is reconfigured avoiding the domain that originated the violation.**

***Keywords*—Network orchestration, DLT, Inter-domain.**

I. Introduction

The new generation of services and applications designed to operate on top of 5G and Beyond (B5G) networks are accompanied with an increase in complexity and size of the communication networks. The heterogeneity of technologies involved in the networks and the need to interconnect different network domains belonging to different operators pose many new challenges. One of the most prominent challenges concerns the security and privacy of the domain. Network operators want to keep the internal details of their domains private, while offering the high quality Transport Network Slices (TNSs) demanded by their customers with corresponding Service Level Agreements (SLAs) [1].

To deploy the inter-domain services in a reliable manner, as imposed by some SLA constraints, the element in charge of establishing the end-to-end connectivity needs to (*i*) know basic topological information of the domains, and (*ii*) provide appropriate traceability of inter-domain service requests. To protect the privacy of the network operators and minimize the interactions among them, abstracted network topologies have been proposed in [2]. Then, each domain can upload their per-domain abstracted topological information towards a centralized and permissioned repository accessible to the rest of the domains. The Blockchain technology has been proved to be a feasible solution for uploading and consolidating, in a distributed manner, the information coming from multiple sources and preventing any of these sources from taking ownership of the information [2]. Blockchain prevents data repudiation providing a source of trust for SLA validation.

To provide the demanded Quality of Service (QoS) assurance and traceability of the inter-domain services, some form of leadership between the involved domains needs to be defined. The domain that acts as a leader, should decide the end-to-end path and the list of domains traversed, communicate with the rest of the involved domains, monitor the end-to-end SLAs, and implement the appropriate mitigation mechanisms in case an SLA is violated. Authors in [3] used the Software-Defined Network (SDN) controller instance associated with the source domain of the inter-domain service as the leader.

This paper presents the main responsibilities of the Inter-Domain Component (IDC) and the Distributed Ledger Technology (DLT) components we are developing for the ETSI TeraFlowSDN (TFS) controller to create end-to-end transport network slicing services. Among others, these components manage the QoS-aware inter-domain connectivity services and enable the interactions between local and peer TFS instances, each responsible for a different network domain.

II. Demonstration

The testbed setup considered for this demonstration is illustrated in Figure 1. The data plane of each domain is implemented as a set of emulated packet routers. On top each domain's data plane, an instance of the TFS controller is in charge of controlling the respective data planes through NetConf/OpenConfig interfaces.

Each instance of the TFS controller features a DLT component in charge of interfacing with a vendor- and operator–agnostic permissioned Blockchain used to share the per-domain details with neighbor domains. The TFS controller incorporates an IDC component in charge of the interoperation with the remote peers to establish the end-to-end transport network slicing services.

All the TFS instances share an abstraction of their respective domain's topology. The IDC component is also responsible for composing the abstracted view of the local domain that is exposed to the remote peers. In that regard, we follow the approach proposed in [2] where the internal topologies of the

2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) | 978-1-6654-7334-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/NFV-SDN56302.2022.9974726

domains are abstracted into a single abstract node. The ports declared in the abstracted node correspond to the ports in the border devices of the domain that are connected with remote domains.

When an inter-domain transport network slicing service needs to be created, the source domain acts as the leader, decides the list of domains to be traversed, and inter-operates with these domains through the IDC components in each TFS instance. The communication between the IDC components is also supported by the DLT component. The latter stores and detects records in the Blockchain corresponding to inter-domain transport slices. If a new inter-domain slice request is received, the local IDC component stores through the DLT component the record for that inter-domain slice into the Blockchain. When the DLT component receives a remotely-originated inter-domain slice record, it redirects the record towards the local IDC component for further processing.

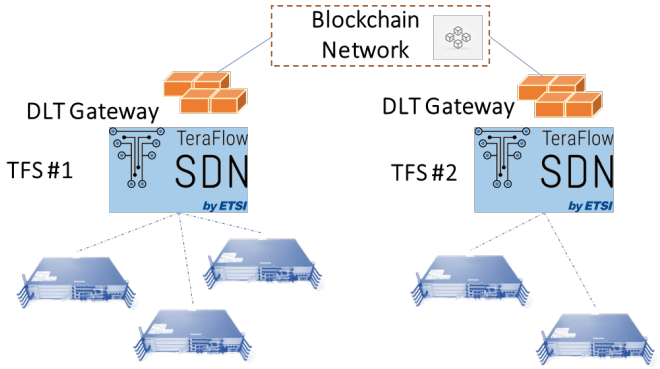

Fig. 1: Demonstration architecture.

III. Workflows

Three workflows are planned for this demonstration to showcase the inter-domain capacities of the TFS controller.

A. Share the local abstracted topology with remote domains

The workflow (see Fig. 2) starts with the IDC at each domain subscribing to receive notifications on changes done on the local Context component (1.1). The entities monitored in this workflow, denoted by the symbol *X*, are: *Context*, *Topology*, *Device*, *Link*, *Service*, and *Slice*. Besides, each DLT component subscribes to receive notifications when a change is done on the Blockchain (2.1).

When a change is stored on the local domain (#A) Context component, an event is broadcast (1.2) to all the subscribed TFS components. The IDC receives the event, re-computes the local abstracted topology (1.3), and for each change, it issues a *Record* request towards the DLT component (1.4 and 1.7), that creates, updates, or marks as deleted the record on the Blockchain (1.5 and 1.6), and synchronizes the change with the other peers of the Blockchain (1.8).

When the remote domain (#N) synchronizes the changes (2.2), if a record associated with the subscription has been changed, it issues a notification towards the DLT component containing the unique record identifier (2.3). The DLT component interrogates its associated Blockchain peer to retrieve the up-to-date record (2.4 and 2.5) and updates the appropriate record on its Context component (2.6 and 2.7), that stores the local copy of the domain #A's abstracted topology.

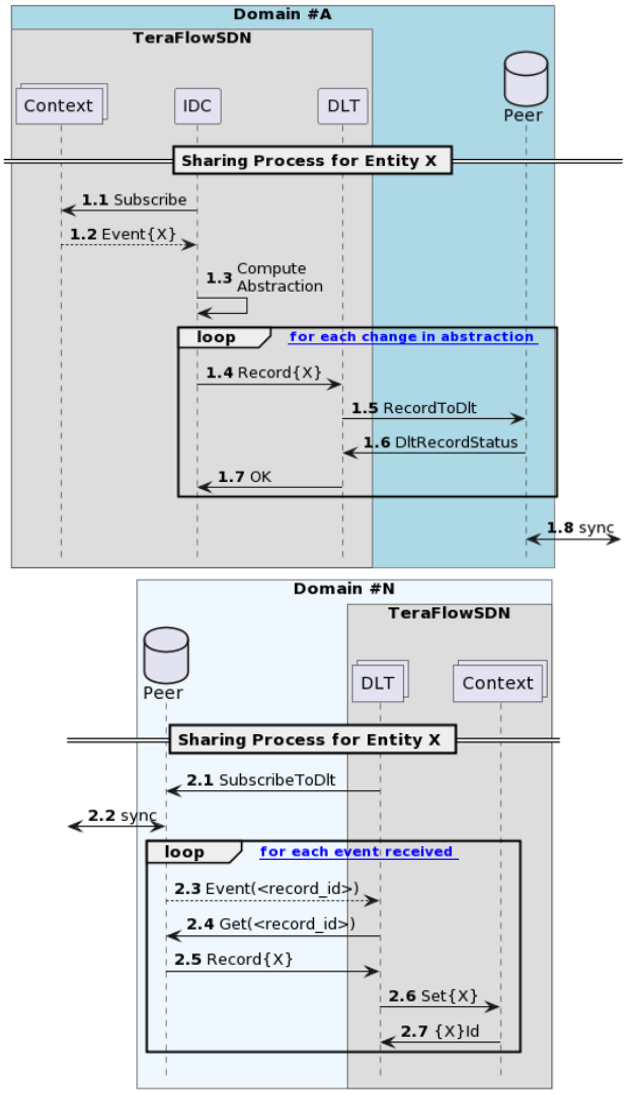

Fig. 2: Send/Receive (top/bottom) changes through the DLT.

B. Activate inter-domain slice with SLAs

The workflow (see Fig. 3) starts when the Slice component receives a request to create an inter-domain slice, for instance, from a customer or an external system. The Slice component issues an inter-domain slice creation request to the local IDC (1.1) that delegates on the Path Computation (PathComp) component to compute the end-to-end inter-domain path (1.2). The PathComp component uses the inter-domain data on the Context component to compose the global abstract topology, and retrieve the list of traversed domains and abstract nodes per domain (1.3).

Then, for each traversed domain, the IDC inspects if it is the local or a remote domain. For the local domain (1.4 and 1.5), it configures the services appropriately through the Slice and Service components that create the required SLA policies on the Policy component (not depicted for the sake of simplifying the workflow) according to the requested slice. For the remote domains, it stores a Slice record carrying the

required SLAs on the Blockchain through the DLT component (1.6), and waits for the reception of an event from the remote domain informing that the Slice record was modified. When this event is received, it retrieves that record and inspects that the Slice was properly activated (not shown in the figure), and returns the control to the IDC. Note that, for the sake of efficiency, requests to the multiple domains are issued in parallel.

On the remote domain (#N), when a Slice record belonging to the domain (#N) is created, and the event is detected (2.1), the DLT component delegates on the IDC the creation of the inter-domain sub-slice (2.2). The IDC issues a local slice creation request to the Slice component (2.3 and 2.4) similar to that described for messages 1.4 and 1.5. When the slice is activated, the IDC replies to the DLT component with the up-to-date Slice record (2.5); the DLT component stores it on the Blockchain (2.6).

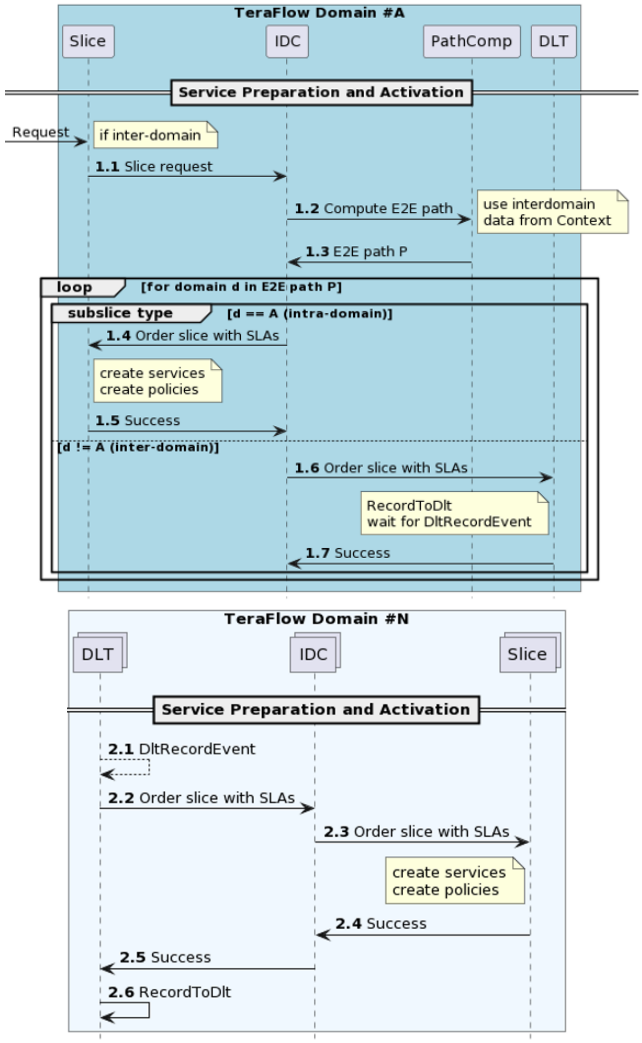

Fig. 3: Activate Inter-domain Slices with SLAs at the source domain (top) and the other crossed domains (bottom).

C. Handle SLA violations on inter-domain slices

The workflow starts when a connectivity service involving SLAs is created. The Service component interacts with the Policy component to create policy rules to monitor the performance of the connectivity service. The Policy component subscribes to receive alarm notifications from the Monitoring component. The Device component is in charge of collecting status reports and alarms from the devices and storing them on the Monitoring component. The latter broadcasts the alarms to the subscribers.

When an alarm is received by the Policy component, it identifies the possible issues, e.g., an SLA violation on a connectivity service, and executes the action configured for that policy. In this workflow, we assume the issue is a latency violation. In that case, the Policy component updates the corresponding Service with the status of *SLA_VIOLATED*; this status change is then propagated to the Slice component.

For intra-domain slices, the issue is addressed locally by deleting and re-creating the slice and the associated connectivity services. For inter-domain slices, passing through the IDC and DLT components, the inter-domain Slice record with the status of *SLA_VIOLATED* is propagated until the domain responsible for the affected service. On that domain, the slice is deleted and re-created excluding the appropriate resources. If the issue persists, the source domain marks as excluded the entire remote domain for this end-to-end slice and the inter-domain slice is re-established.

IV. Relevance

The topic of secured inter-domain end-to-end transport network slicing is of major importance for B5G networks. This demonstration will showcase a novel strategy to enable network operator interactions while preventing to share internal network details. We expect this demonstration will be very interesting for the community and will motivate discussions in the transport network community regarding network abstraction, secure operator interactions, and end-to-end transport slice management involving SLAs.

V. Conclusion

In this demonstration, we showcase the feasibility of the proposed inter-domain solution, where a number of TFS controller instances collaborate to provide End-to-End Transport Network Slices. To this end, we will detail the design of the IDC, that features service preparation and activation, service modification, and synchronization of service monitoring data between domains through the Blockchain.

Acknowledgment

The research has received funding from the EC H2020 TeraFlow Project (101015857), Spanish MINECO 6GMICROSDN-NOS (TSI-063000-2021-20), and RELAMPAGO grant PID2021-127916OB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

[1] R. Vilalta *et al.*, "Teraflow: Secured autonomic traffic management for a tera of sdn flows," in *2021 Joint European Conf. on Netw. and Commun. & 6G Summit (EuCNC/6G Summit)*. IEEE, 2021, pp. 377–382.

[2] P. Alemany *et al.*, "Evaluation of the abstraction of optical topology models in blockchain-based data center interconnection," *J. of Optical Communications and Networking*, vol. 14, no. 4, pp. 211–221, 2022.

[3] R. Vilalta *et al.*, "End-to-end interdomain transport network slice management using cloud-based sdn controllers," in *OECC/PSC*, 2022.