End-to-end Interdomain Transport Network Slice Management Using Cloud-based SDN Controllers

Ricard Vilalta[1], Lluis Gifre[1], Min Xie[2], Jane Frances Pajo[2], Håkon Lønsethagen[2], Stanislav Lange[3], Harald Øverby[3], Thomas Zinner[3], Raul Muñoz[1], Ramon Casellas[1], Ricardo Martínez[1]
[1]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Spain
[2]Telenor, Fornebu, Norway
[3]Norwegian University of Science and Technology, Norway

Author e-mail address: ricard.vilalta@cttc.es

Abstract: *This paper provides dedicated QoS-aware inter-domain connectivity services and enabling interaction between TeraFlowSDN instance and peer TeraFlowSDN instances that manage different network domains to create E2E transport network slicing service.*
Keywords: *Computer networks, Next generation networking, Virtual private networks, Optical fiber networks*

I. Introduction

Currently, most Internet traffic is served in a best-effort manner, and it is populated mostly by elastic applications that can adapt to changing conditions and do not pose strict overlay performance requirements [1]. However, the expected application heterogeneity and demands in the B5G era cannot be satisfied that way. Traditional overprovisioning, which is typically used to overcome resource constraints, results in low resource and energy efficiency, particularly outside peak hours. The Next Generation Internet (NGI) vision indicates an internetworking approach that supports new business model enablers at the network layer for sustainable business model innovation at the application level allowing and enabling application service and product diversity. This creates innovation opportunities for SMEs, which cannot otherwise build global private backbone networks, as can be done by large OTTs. Such SMEs and Online Application service Providers (OAPs) will rather be able to leverage an on-demand and open access Value Added Connectivity (VAC), from any endpoint to any endpoint, for their own service innovation and differentiation of their offerings [2]. Transport Network Slices are the expected services that will provide VAC[3].

To support service providers, the authors have presented a novel autonomous cloud-native optical software defined networking (SDN) controller in [4]. The TeraFlowSDN controller is based on a micro-service architecture. The objective is to foster innovation around multi-layer SDN controllers and evolve them to be suitable for beyond 5G networks. This paper presents the need to include more dynamicity to support operator requirements for new types of connectivity services. The proposed scenario demonstrates inter-domain connectivity services. In this scenario, novel techniques for domain inter-connection will be studied, as well as the load balancing of the connectivity service requests will be evaluated at cloud-scale.

This paper presents the main goals and responsibilities of the inter-domain component (IDC), which includes providing dedicated QoS-aware inter-domain connectivity services and enabling interaction between TeraFlowSDN instances and peer TeraFlowSDN instances which manage different network domains to create E2E Transport Network slicing services. Based on these goals, this paper demonstrates the design of the IDC along with its main interaction components and interfaces. Furthermore, we provide an overview of requirements from the IDC towards other TeraFlowSDN components as well as preliminary results.

II. Inter-domain provisioning of End-to-End Transport Network Slices

The inter-domain communication services can be classified into two types: inter-domain between administrative domains vs. inter-domain between technology domains (within one administrative domain), as shown in Fig.1(left). Depending on the type, different governance models are needed before defining the micro-services and corresponding requirements. In this paper, we focus on Type I, as shown in Fig.1(left).

To achieve its objectives, the proposed inter-domain component leverages existing data models [5] related to the context and slice components. The main operations include the authentication required prior to the information exchange between inter-domain components as well as interaction with the slice catalogue. Interactions with the slice catalogue include querying the catalogue (LookUpSlice), requesting the instantiation of an existing slice (OrderSliceFromCatalog), and the creation of a new slice as well as its inclusion into the catalogue (CreateSliceAndAddToCatalog).

Key functions of the inter-domain component are related to service lifecycle management, including the preparation and activation of a service as well as its modification during run time. Furthermore, monitoring of service KPIs across domains ensures that end-to-end requirements are met.

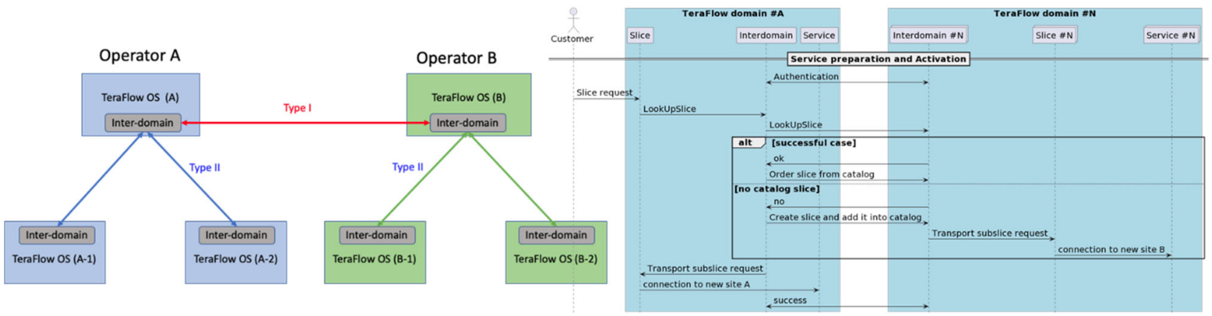

Fig. 1. Left, Typology of inter-domain communication services. Right, E2E slice provisioning through multiple control-domains sequence diagram.

Fig.1(right) displays the sequence diagram regarding service preparation and activation. The workflow is initiated by a customer which can be any entity consuming TeraFlow services such as the OSS or other management domains including end-to-end service management. The customer's request is handled by the slice component which forwards the end-to-end transport slice request to the inter-domain component. The inter-domain component, in turn, decomposes the end-to-end transport slice into per-domain sub-slices and requests their creation in the respective TeraFlow domains through the corresponding inter-domain components. This inter-domain communication is performed in a secure manner by mutual authentication prior to the exchange of sub-slice requests.

In case an appropriate sub-slice can be provided, the remote inter-domain component informs the requesting inter-domain component, and the latter can order the slice. Otherwise, the creation of a corresponding slice is initiated alongside its insertion into the catalogue and establishment of connectivity, triggering interaction with slice and service components, respectively.

Finally, the same procedure of sub-slice creation and connection establishment is performed at the local TeraFlow domain (domain #A in the figure).

III. Experimental evaluation

TeraFlowSDN is a cloud-native SDN controller, where each component runs as a micro-service. The design of the IDC is based on three use cases: service preparation and activation, service modification, and synchronization of service monitoring data between domains. We have identified and amended interactions that involve the IDC and provide an overview of the resulting IDC architecture in Fig.2.

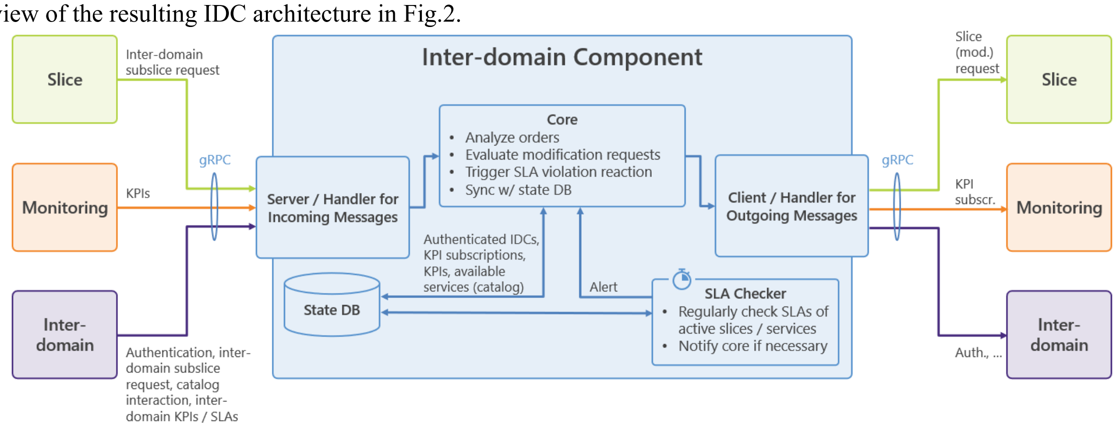

Fig. 2. Inter-domain component proposed architecture and its relationship with other TeraFlowSDN components.

In order to achieve a separation of concerns within the IDC, we propose a modular design with the following sub-components:

- Handlers for receiving and forwarding incoming and outgoing gRPC-based messages act as the main interaction points with other TeraFlowSDN components, as well as another IDC component from another TeraFlowSDN controlled-domain.
- A state database (DB) to keep track of authenticated remote IDC, active subscriptions to service KPIs alongside actual KPI values, as well as a service catalogue and inventory. In order to maintain a clear separation of concerns, the IDC does not perform active measurements or discovery, but the state DB merely represents a local cache that provides fast access to data related to KPI values, catalogue, and inventory. In particular, the latest and most reliable version of the data resides

in the respective TeraFlow components, such as monitoring or slice management. For keeping the state DB up-to-date, two options are considered: (a) the IDC can behave like a proxy and forward requests towards the responsible component on demand, storing the most recent data; or (b) the IDC can keep a local copy and synchronize it regularly with the authoritative data source.

- The Core module constitutes the main decision-making entity within the IDC. It is in charge of making decisions based on received messages related to service creation or modification requests, reacting to detected SLA violations, and keeping the state DB up to date.
- The SLA Checker module regularly queries monitoring information from the state DB to make sure that E2E SLAs are met, and alerts the Core module to trigger corrective actions in case of violations. It is worth noting that the IDC does not have to share detailed KPI and infrastructure information between domains; instead, it can transmit high-level, aggregated, and potentially anonymized information on SLA conformance status. Preliminary results regarding different options for sharing aggregated topology information are discussed in more depth in [5].

Fig.3. shows the authentication sequence between two IDC from different TeraFlowSDN controllers. The permissioned TeraFlowSDN peer information is stored in configuration file.

Time	Source	Destination	Protocol	Length	Info
10.009	d1-interdomain	d2-interdomain	GRPC	492	Magic, SETTINGS[0], SETTINGS[0], HEADERS[1]: POST /interdomain.InterdomainService/Authenticate, WINDOW_UPDATE[1], DA
10.010	d2-interdomain	d1-interdomain	GRPC	261	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.AuthenticationResult, HEADERS[1], WINDOW_UPD
10.127	d2-interdomain	d1-interdomain	GRPC	423	SETTINGS[0], HEADERS[1]: POST /interdomain.InterdomainService/Authenticate, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOB
10.128	d1-interdomain	d2-interdomain	GRPC	261	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.AuthenticationResult, HEADERS[1], WINDOW_UPD

Fig. 3. Wireshark capture of Authenticate sequence.

Once TeraFowSDN controllers are authenticated, an E2E Transport Network Slice request can be triggered (in Fig.4. from OpenSourceMANO). As no slices have been created, it can be observed as they are requested from inter-domain TeraFlowSDN controller 1 to controller 2.

REF	OSM	d1-compute	HTTP/JSON	226	POST /restconf/data/ietf-l2vpn-svc:l2vpn-svc/vpn-services HTTP/1.1 , JavaScript Object Notation (application/json)
0.119	d1-compute	d1-slice	GRPC	435	SETTINGS[0], HEADERS[1]: POST /slice.SliceService/CreateSlice, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.Slice
0.125	d1-slice	d1-compute	GRPC	299	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.SliceId, HEADERS[1], WINDOW_UPDATE[0]
0.126	d1-compute	OSM	HTTP/JSON	71	HTTP/1.0 201 CREATED , JavaScript Object Notation (application/json)
0.132	OSM	d1-compute	HTTP/JSON	439	POST /restconf/data/ietf-l2vpn-svc:l2vpn-svc/sites/site=1/site-network-accesses/ HTTP/1.1 , JavaScript Object Notation (
0.276	d1-compute	d1-slice	GRPC	455	SETTINGS[0], HEADERS[1]: POST /slice.SliceService/UpdateSlice, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.Slice
0.281	d1-slice	d1-compute	GRPC	299	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.SliceId, HEADERS[1], WINDOW_UPDATE[0]
0.282	d1-compute	OSM	HTTP	176	HTTP/1.0 204 NO CONTENT
0.289	OSM	d1-compute	HTTP/JSON	439	POST /restconf/data/ietf-l2vpn-svc:l2vpn-svc/sites/site=2/site-network-accesses/ HTTP/1.1 , JavaScript Object Notation (
0.427	d1-compute	d1-slice	GRPC	475	SETTINGS[0], HEADERS[1]: POST /slice.SliceService/UpdateSlice, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.Slice
0.440	d1-slice	d1-interdomain	GRPC	559	Magic, SETTINGS[0], SETTINGS[0], HEADERS[1]: POST /interdomain.InterdomainService/RequestSlice, WINDOW_UPDATE[1], DATA[1
0.442	d1-interdomain	d2-interdomain	GRPC	280	HEADERS[3], WINDOW_UPDATE[3], DATA[3] (GRPC), WINDOW_UPDATE[0]
0.445	d2-interdomain	d1-interdomain	GRPC	131	HEADERS[3]: 200 OK, WINDOW_UPDATE[3], DATA[3], HEADERS[3] (GRPC), WINDOW_UPDATE[0]
0.446	d1-interdomain	d2-interdomain	GRPC	295	HEADERS[5], WINDOW_UPDATE[5], DATA[5] (GRPC), WINDOW_UPDATE[0]
0.448	d2-interdomain	d2-slice	GRPC	487	SETTINGS[0], HEADERS[1]: POST /slice.SliceService/CreateSlice, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.Slice
0.459	d2-slice	d2-service	GRPC	456	SETTINGS[0], HEADERS[1]: POST /service.ServiceService/CreateService, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context
0.469	d2-service	d2-slice	GRPC	311	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.ServiceId, HEADERS[1], WINDOW_UPDATE[0]
1.262	d2-slice	d2-interdomain	GRPC	311	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.SliceId, HEADERS[1], WINDOW_UPDATE[0]
1.282	d2-interdomain	d1-interdomain	GRPC	305	HEADERS[5]: 200 OK, WINDOW_UPDATE[5], DATA[5], HEADERS[5] (GRPC), WINDOW_UPDATE[0]
1.283	d1-interdomain	d1-slice	GRPC	484	SETTINGS[0], HEADERS[1]: POST /slice.SliceService/CreateSlice, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.Slice
1.292	d1-slice	d1-service	GRPC	452	SETTINGS[0], HEADERS[1]: POST /service.ServiceService/CreateService, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context
1.303	d1-service	d1-slice	GRPC	308	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.ServiceId, HEADERS[1], WINDOW_UPDATE[0]
2.119	d1-slice	d1-interdomain	GRPC	308	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.SliceId, HEADERS[1], WINDOW_UPDATE[0]
2.133	d1-interdomain	d1-slice	GRPC	299	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.SliceId, HEADERS[1], WINDOW_UPDATE[0]
2.181	d1-slice	d1-compute	GRPC	299	HEADERS[1]: 200 OK, WINDOW_UPDATE[1], DATA[1] (GRPC) (PROTOBUF) context.SliceId, HEADERS[1], WINDOW_UPDATE[0]
2.183	d1-compute	OSM	HTTP	176	HTTP/1.0 204 NO CONTENT

Fig. 4. Inter-domain End-to-End Transport Network Slice deployment

IV. CONCLUSIONS

In this paper, we have demonstrated the feasibility of the proposed inter-domain solution, where two TeraFlowSDN instances have collaborated in order to provide an End-to-End Transport Network Slice. To this end, we have detailed the internal design of inter-domain component, that provides multiple use-case support, such as service preparation and activation, service modification, and synchronization of service monitoring data between domains.

ACKNOWLEDGMENT

This work is partially funded by the EC through the EC/5GPPP TeraFlow PROJECT (101015857) and the "Grant RTI2018-099178-B-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by ERDF A way of making Europe".

REFERENCES

[1] H. Lønsethagen, et al. “Service Level Awareness and open multi-service internetworking,” Internet Requests for Comments, NetWorld, whitepaper, 2016.

[2] G. Biczok, M. Dramitinos, L. Toka, P. E. Heegaard, and H. Lonsethagen, “Manufactured by software: SDN-enabled multi-operator composite services with the 5G exchange,” IEEE Communications Magazine, 2017.

[3] A. Farrel, et al., Framework for IETF Network Slices, IETF draft-ietf-teas-ietf-network-slices-08, 2022.

[4] R. Vilalta et al., "TeraFlow: Secured Autonomic Traffic Management for a Tera of SDN flows," 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2021, pp. 377-382, doi: 10.1109/EuCNC/6GSummit51104.2021.9482469.

[5] P. Alemany, R. Vilalta, R. Muñoz, R. Casellas, and R. Martínez, "Evaluation of the abstraction of optical topology models in blockchain-based data center interconnection," J. Opt. Commun. Netw. 14, 211-221 (2022).