Network Slicing with Multi-Topology Routing

Nicolas Huin, Sébastien Martin, Jérémie Leguay, Shengming Cai
Huawei Technologies Co. Ltd.
Email: {firstname.lastname}@huawei.com

***Abstract*—The deployment of 5G networks is paving the road to custom network services. It is now possible to envision the automatic decomposition of a physical network into several virtual networks to serve a wide range of user needs. This technology is also referred to as *network slicing*. To guarantee the strict isolation of virtual networks, it is possible to rely on underlay technologies such as Flex Ethernet (FlexE).**

In this demo, we present a slicing solution based on Multi-Topology-Routing (MTR). We will demonstrate how IGP weights can be designed for the embedding of a slice, described by a traffic matrix and end-to-end latency requirements, to minimize the cost of underlay bandwidth reservations.

***Index Terms*— Network Slicing, Multi-Topology IGP Routing, Flex Ethernet, Combinatorial Optimization.**

I. Introduction

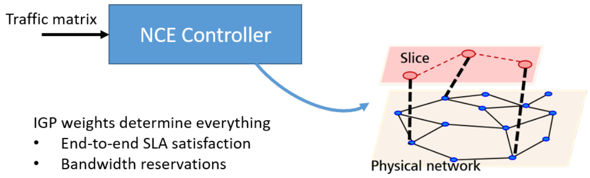
“Slicing” a network means creating virtual networks with different SLA requirements, operated by different tenants, on top of a common physical network [1]. Virtual links and virtual nodes can be easily established by a Software Defined Network (SDN) controller or a network orchestrator [2].

To guarantee traffic isolation between slices, different data plane technology can be used: from *soft* slicing [3] with traditional QoS and VPN technologies, to *hard* slicing [4] with technologies like Flex Ethernet [5] that leverages on a TDMA-like sharing of the capacity for strict isolation.

Extensions of Segment Routing (SR) technologies for hard slicing are under discussion at IETF and known as *Enhanced Virtual Private Networks (VPN+)* [6]. In this proposal, a set of dedicated underlay resources (e.g., FlexE sub-interfaces) is advertised to the network layer as labels. Via an SR label stack, the source node explicitly states the underlay resources that must transmit each packet. However, most operators still rely on standard IGP routing.

Our demo presents a turnkey solution for network slicing where operators cannot rely on SR and instead rely on a standard IGP routing extension called Multi-Topology Routing (MTR) [7]. To increase the routing flexibility in IGP protocols, MTR have been proposed for OSPF and IS-IS, for instance. In this case, the protocol maintains a separate Routing Information Base (RIB) and Forwarding Information Base (FIB) for each topology. Each slice must be deployed as a specific IGP topology with its own set of weights (i.e., its own IGP instance).

In this demo, we show how we can create a network slice on top of a physical network using MTR. We introduce the optimization problem that needs to be solved to decide link weights while accounting for traffic and end-to-end latency requirements, ECMP routing when possible and slotted reservations of underlay resources. We demonstrate how our algorithmic framework performs in practice for the embeding of a slice defined by a traffic matrix and QoS constraints.



IGP weights determine everything
- End-to-end SLA satisfaction
- Bandwidth reservations

Fig. 1. Slicing controller creating and deploying a slice.

II. MTR Slicing Problem

Given a set of services with end-to-end latency requirements and a physical network topology with link capacity and latency attributes, the goal is to decide link weights in order to minimize the bandwidth reservation cost and route services on paths that respect their QoS constraints. As depicted in Fig. 1, link weights of the IGP topology decide everything: the routing paths and the amount of bandwidth needed on physical links. ECMP is activated only when equal cost paths are found at a node. The basic problem of IGP metric design is called in the literature IP Network Design [8] and is a NP-Hard problem.

We propose a compact formulation for the MTR slicing problem. Its constraints can be grouped into four different parts. First, we consider the routing and the slot allocation and the minimization of slotted bandwidth reservations. Given a set of binary variable y_{es} active if the slot configuration s on a given link e is used and a set of variables x_{ek} representing the amount of bandwidth routed on link e for demand k, the first part of the formulation is as follows:

$$\min \sum_{e\in E} C_e \sum_{s\in S^e} \xi_{es} y_{es} \tag{1a}$$

$$s.t. \sum_{s\in S^e} y_{es} \leq 1 \qquad \forall e \in E \tag{1b}$$

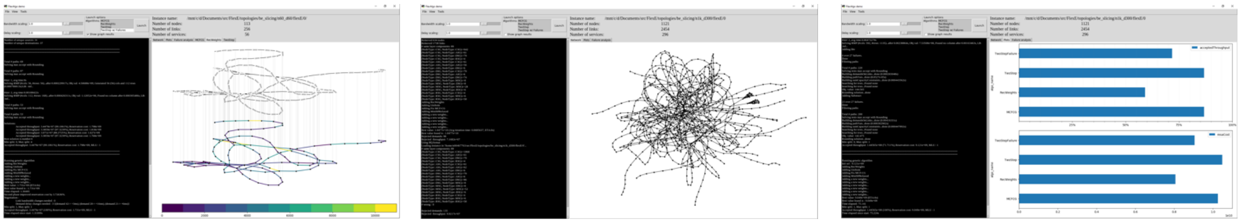
$$\sum_{e\in\omega^+(v)} x_{ek} - \sum_{e\in\omega^-(v)} x_{ek} = \begin{cases} D_k & \text{if } v = s_k \\ -D_k & \text{if } v = t_k \\ 0 & \text{otherwise} \end{cases} \qquad \forall v, \forall k \tag{1c}$$

$$\sum_{k\in K} x_{ek} \leq \sum_{s\in S^e} \xi_{es} y_{es} \qquad \forall e \in E \tag{1d}$$

The goal is to minimize the cost of the network. Constraints (1c) are the flow conservation constraints, constraints (1d) are the capacity constraints. Finally, we limit the number of slot configuration to 1 per link with constraints (1b).

ECMP constraints. Then, we consider the even split of traffic on nodes and we need a new set of variables p_{vk} that represent the amount of flow going out of node v for the

Annex to ISBN 978-3-903176-39-3© 2021 IFIP



Instance name: /mnt/c/d/Documents/src/FlexE/topologies/be_slicing/n60_d60/flexE/0/
Number of nodes: 113
Number of links: 256
Number of services: 56

Instance name: /mnt/c/d/Documents/src/FlexE/topologies/be_slicing/n1k_d300/flexE/0
Number of nodes: 1121
Number of links: 2454
Number of services: 296

Instance name: /mnt/c/d/Documents/src/FlexE/topologies/be_slicing/n1k_d300/flexE/0
Number of nodes: 1121
Number of links: 2454
Number of services: 296

Fig. 2. Interface of the MTR slicing demonstrator.

demand k after split.

To be able to link variables p and x, we need to keep track of which links are used by which demand; we introduce the set of boolean variables z_{ek}, equal to 1 if demand k is routed through link e and link them to variables x via the following two constraints:

$$x_{ek} \leq D_k z_{ek} \qquad \forall e \in E, \forall k \in K \tag{1e}$$

$$z_{ek} \leq D_k x_{ek} \qquad \forall e \in E, \forall k \in K \tag{1f}$$

Then, we can link all three set of variables (x, p, and z) using the two following constraints:

$$x_{ek} \leq p_{s_e k} \qquad \forall e \in E, \forall k \in K \tag{1g}$$

$$x_{ek} \geq p_{s_e k} - D_k(1 - z_{ek}) \qquad \forall e \in E, \forall k \in K \tag{1h}$$

Weight constraints. Then, we consider the IGP weights by introducing two new set of variables: π_v^k that represents the distance from s_k to v and w_e the IGP weight of link e. The new constraints are as follows:

$$w_e \geq \pi_{t_e}^k - \pi_{s_e}^k \qquad \forall e \in E, \forall k \in K \tag{1i}$$

$$w_e \geq \pi_{t_e}^k - \pi_{s_e}^k + 1 - z_{ek} \qquad \forall e \in E, \forall k \in K \tag{1j}$$

$$w_e \leq \pi_{t_e}^k - \pi_{s_e}^k + M(1 - z_{ek}) \qquad \forall e \in E, \forall k \in K \tag{1k}$$

Constraints (1i) correspond to the dual feasibility constraints of the shortest path problem. Constraints (1j) and (1k) ensure that if a demand k uses a link e then it must belong to the shortest path to t_k by setting the weight accordingly.

Delay constraints. Finally, we need to constrain the delay of each demand. We introduce a new set of variables δ_v^k that represents the worst delay from s_k to v, and the following constraints:

$$\delta_{t_e}^k \geq \delta_{s_e}^k + \lambda_e z_{ek} - \Lambda_k(1 - z_{ek}) \qquad \forall e \in E, k \in K \tag{1l}$$

$$\delta_{t_k}^k \leq \Lambda_k \qquad \forall k \in K \tag{1m}$$

Constraints (1l) propagate the delay on a link only if it is used by a demand. Constraints (1m) limit the worst delay to the destination of the demands.

III. DEMONSTRATION

In this demo, we showcase a two-steps algorithm for the embedding of MTR slices that is based on math-heuristics and local search methods to design IGP weights. In the first step it maximizes traffic acceptance, while in the second step it minimizes the cost of the embedding for accepted services. We compare this algorithm, called *TwoStep*, to two baseline approaches. The first benchmark solution, called *RecWeights*, uses standard IGP metrics from OSPF's RFC where link weights are $10^8/c_e$ (where c_e are link capacities). The second benchmark solution, called *MCF*, considers that segment routing can be used to steer traffic with maximum routing flexibility. In this case a Multi-Commodity Flow (MCF) problem is solved using also an efficient math-heuristic.

Using an interactive graphical interface, we show on small and large IPRAN topologies with thousands of nodes that MTR slices can be created within a few seconds. We compare for the 3 routing solutions (i.e., MTR, RecWeight and MCF) the accepted traffic and the cost of bandwidth reservations. We show that even if routing is restricted to shortest paths over a designed topology, MTR slicing is a competitive solution compared to MCF (i.e., SR-based slicing).

The demonstrated steps are the following:

1) We will first show how a small and a large slice can be created by a network controller. As presented in Fig. 2, a GUI is used to load input data, modify SLA constraints and network capacity, visualize routing and IGP metrics.

2) Then, we will show an additional feature where IGP metrics are designed so as to protect all services against all possible 1-link failures. For this case we also compare to SR and IGP benchmark solutions.

The video of the demo is available at https://drive.google.com/file/d/1tmmTRMACoWQYaxjrKDG0_9nW9XDkxnQu/view?usp=sharing

REFERENCES

[1] 5G Service-Guaranteed Network Slicing White Paper. Huawei's whitepaper, February 2017.
[2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, Jan 2015.
[3] A. Destounis, G. Paschos, S. Paris, J. Leguay, L. Gkatzikis, S. Vassilaras, M. Leconte, and P. Medagliani. Slice-based column generation for network slicing. In *IEEE INFOCOM demo*, 2018.
[4] Nicolas Huin, Jérémie Leguay, Sébastien Martin, Paolo Medagliani, and Shengmin Cai. Routing and Slot Allocation in 5G Hard Slicing. In *INOC*, pages 72–77, 2019.
[5] Flex ethernet 2.0 implementation agreement, June 2018.
[6] J. Dong, S. Bryant, Z. Li, T. Miyasaka, and Y. Lee. A Framework for Enhanced Virtual Private Networks (VPN+) Services. In *IETF Draft*, 2020.
[7] P Psenak, S Mirtorabi, A Roy, L Nguyen, and P Pillay-Esnault. Multi-topology (MT) routing in OSPF. Technical report, RFC 4915, June, 2007.
[8] Kaj Holmberg and Di Yuan. Optimization of internet protocol network design and routing. *Wiley Networks*, 2004.