Resource Defragmentation for Network Slicing

*Paolo Medagliani, *Sébastien Martin, *Jérémie Leguay,
*Shengming Cai, *Feng Zeng †Nicolas Huin
*Huawei Technologies Co. Ltd.
†IMT Atlantique, IRISA, Cesson-Sévigné

***Abstract*—Network automation in the fifth generation of mobile networks (5G) requires network slices to be efficiently computed and deployed. In order to guarantee physical isolation for virtual networks and no interference between different slices, it is possible to rely on *hard slicing*, whose principles can be implemented using Flex Ethernet (FlexE) technology. As slices are created and deleted over time, it is necessary from time to time to defragment resources and reoptimize bandwidth reservations for the remaining slices.**

In this demo, we present our algorithmic framework, based on Column Generation, and we showcase how to efficiently defragment the network in order to reduce the Maximum Link Utilization (MLU) of current reservations with a minimum number of configuration changes.

Index Terms*—*Resource Allocation, Network Slicing, Network Virtualization, Combinatorial Optimization.

I. Introduction

In 5G networking, we are witnessing the progressive consensus on the need for "slicing" the network, i.e., the possibility of using the same physical network infrastructure to serve different tenants with different SLA requirements. Two main requirements for the effective use of slicing into networks are isolation of users and deterministic performance guarantees. While the former feature is guaranteed by both *soft* [1] and *hard* [2] slicing, the latter can be only addressed by hard slicing, where resources are statically assigned to each user, avoiding interferences from other slices due to heavy load. The emerging standard for hard slice isolation is Flex Ethernet, a TDMA-like resource subdivision between different slices [3].

Within each slice, the customer defines, for each service, the source and the destination endpoints, and the requested bandwidth and SLA (e.g., end-to-end latency). Using this set of requirements, the network controller can compute the bandwidth resources to be assigned to the slice and enforce it into the network. During network O&M, slice creation requests from different customers may come in batches, causing a successive allocation of slices. In addition, some slices may leave the network. For this reason, it is necessary to periodically run a defragmentation on the used resources in order to optimize network utilization, freeing the resources previously assigned for future use.

In this demo, we show how we can defragment the currently used resources for hard slices on top of a physical network. The algorithmic framework, which is based on column generation [4], allows reducing network utilization by changing the current reservations for each slice.

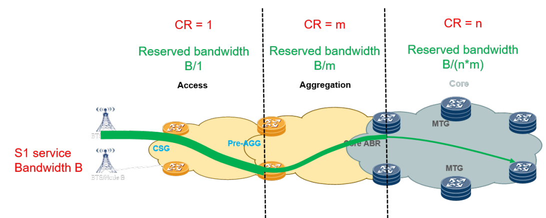

Fig. 1: Convergence ratio (CR) in IPRAN networks.

II. Slicing constraints

Technology constraint As Flex Ethernet (Flex) is based on a TDMA subdivision of the link capacity using a calendar shared by both the FlexE endpoints, the resources are assigned to the slices with a 5 Gbps granularity [3]. When some bandwidth is required on a link, enough slots must be activated to provide at least the given capacity. If some bandwidth remains available in the same slot, it can be used by other demands of the same slice without the need of carrying out another reservation. Huawei's implementation of FlexE offers a finer allocation of resources by dividing the first slot into five 1 Gbps slots. As each slice must use a different interface, the same granularity applies for each slice on each link.

Network constraint In many telco networks, often referred to as IPRAN networks, it is possible to apply statistical multiplexing between some flows, assuming that they can share the resources with other flows that are not active at the same time. For this reason, it is possible to scale down the reserved bandwidth by a multiplicative factor, as shown in Fig 1, referred to as *convergence ratio* (CR), which depends on the considered area of the network.

However, we must also guarantee that when a single service is active, enough bandwidth is reserved on each used link.

III. Slice Planning and Defragmentation

In realistic networks, requests for creation of multiple slices may come in batches, as shown in step 1 of Fig. 2. The controller computes (step 2) the best allocation for each slice, for instance, using algorithms described in [5] or other algorithms such as Constrained Shortest Path (CSP) or Shortest Path (SP), and deploys it into the network. As there could be some slices that are released from the network, the controller monitors the current network utilization (step 3). Whenever new batches of slices arrive (step 4), new reservations for slices are computed (step 5) according to the available resource utilization. As the slices are computed in an online fashion and as slices can leave the network, the current reservation can become suboptimal. It

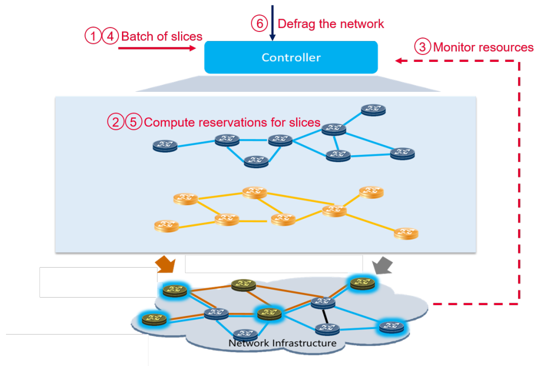

Fig. 2: Steps for slices creation and defragmentation.

is then necessary to run a defragmentation routine (step 6) in order to reduce network utilization and leave space for future slices. The main goals of the defragmentation are: (i) reduction of the Maximum Link Utilization (MLU) in the network and (ii) reduction of the cost of the reservations.

As the defragmentation can change the reservations for a slice in the network, the services may experience performance degradation due to disruption. For this reason, the controller can impose that some VIP slices cannot be reconfigured. Vice versa, for the slices that change their reservation pattern, all the services inside the slice can be changed, i.e., there are no constraints on maximum service reconfiguration.

IV. Algorithm

To efficiently solve the defragmentation problem, we developed an algorithm in two phases, referred to as MCF-Def, based on nested Column Generation (GC), as shown in Fig. 3. The first objective function of the algorithm targets MLU minimization. The output of this phase, i.e., the minimum MLU for the current slice reservations, is used as input to the second phase, where the overall reservation cost is minimized. In each phase of the algorithm, we set a constraint on the maximum number of slices that can be reconfigured.

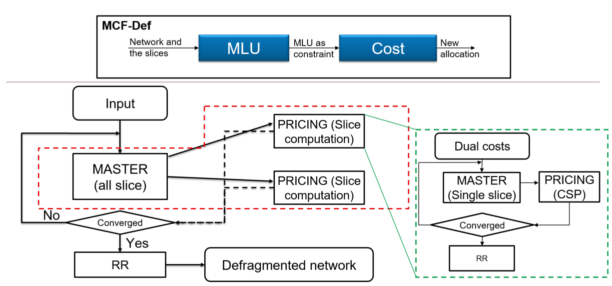

Fig. 3: The two-phases MCF-Def algorithm

The CG routine receives as input the network with the current utilization and the set of running slices. In the external CG routine (on the left), the master aims at minimizing either the MLU or the cost according to the phase of the algorithm currently executed. As in the traditional CG routine, the master and the pricing are solved iteratively to find the optimal, but relaxed solution. In each pricing phase, which can be executed in parallel, a new column, i.e., a new slice corresponding to a given set of requests, is computed according to the dual costs coming from the master. The computation of a single slice (on the right) can be solved using a CG routine where the relevant columns, i.e., the paths for a demand, are added to the master.

The output of the CG routine is the optimal solution for the linearized problem, thus it is necessary to round up to an integer solution via a Randomized Rounding (RR) routine.

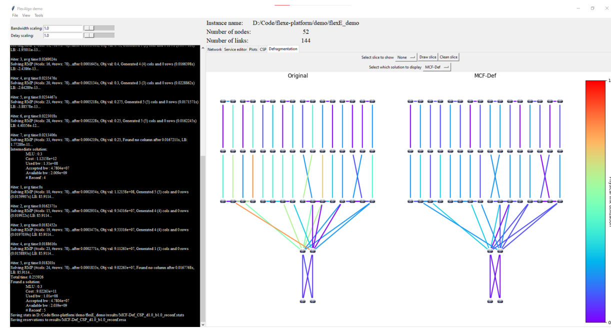

Fig. 4: Interface of the network planning tool.

V. Demo

In this demo, we will first show how the reservations for five slices, with 10 demands each, can be created in an online fashion (i.e., slice by slice) by using the network planning tool shown in Fig. 4. Then, we will show how we can defragment the network, improve the overall MLU, and reduce the overall cost by running the MCF-Def algorithm presented in Sec IV.

The network is composed of 52 nodes and 144 links, with capacities between 10 Gbps and 100 Gbps. No convergence ratio is considered. 5 hard slices with 10 demands each are created and deployed into the network.

The demonstrated steps are as follows. 1) Five slices are deployed into the network in an online fashion, one after the other, each one of them using CSP for slicing.
2) The reserved slices are shown. The bandwidth reserved on the physical links corresponds to the sum of capacities reserved for each slice on the same link.
3) The MCF-Def algorithm is executed over the previous reservations, considering that all the slices in the network can be reconfigured (5 allowed reconfigurations).
4) The MLU is reduced from 0.8 to 0.3 (62.5% gain), the overall reservation cost is reduced from $1.57e^{12}$ to $9.02e^{11}$ (42.5% gain), and the used bandwidth is reduced from $1.84e^{8}$ to $1.01e^{8}$ (45.1% gain).
The video of the demo is available at http://jeremie.leguay.free.fr/infocom2022.mp4

References

[1] 5G Service-Guaranteed Network Slicing White Paper. Huawei's whitepaper, February 2017.
[2] Network Slicing Architecture, January 2018.
[3] Flex Ethernet 2.0 Implementation Agreement, June 2018.
[4] G. Desaulniers, J. Desrosiers, and M.M. Solomon. *Column Generation*. GERAD 25th anniversary series. Springer US, 2006.
[5] Nicolas Huin, Jérémie Leguay, Sébastien Martin, Paolo Medagliani, and Shengming Cai. Routing and slot allocation in 5g hard slicing. In *INOC*, 2019.