A Network Slice Resource Allocation and Optimization Model for End-to-End Mobile Networks

Andrea Fendt[1], Simon Lohmüller[1], Lars Christoph Schmelz[2], Bernhard Bauer[1]
[1] University of Augsburg, Department of Computer Science, Augsburg, Germany
{andrea.fendt, simon.lohmueller, bernhard.bauer}@informatik.uni-augsburg.de
[2] Nokia Bell Labs, Network Management and Automation, Munich, Germany
christoph.schmelz@nokia.com

***Abstract*—With the fifth generation of mobile networks (5G) diverse new use cases arise, including enhanced Mobile Broadband (eMBB), industrial and sensor networks as well as highly safety and security critical communication services. These new use cases introduce very different requirements on the mobile communication networks, e.g., network reliability, latency and throughput. Massive network virtualization and end-to-end mobile network slicing are seen as key enablers to handle those differing requirements and providing mobile network services for the various 5G use cases by a shared physical network infrastructure. However, resource allocation and mobile network slice embedding are still unsolved problems. Therefore, in this paper a standardized and easy to understand Integer Linear Program for offline mobile network slice embedding, especially focusing on resource allocation and virtual node as well as link mapping, will be presented. Moreover, the question of how to efficiently determine a nearly optimal end-to-end mobile network slice embedding on a shared network infrastructure will be answered.**

I. Introduction

Mobile communication technologies are evolving towards constantly connected devices. Not only the number of connected devices and their mobile data traffic are expected to grow by 10 times between 2016 and 2022 [1], but also new use cases are expected to arise such as industrial and sensor networks introducing massive Machine Type Communication (mMTC). In addition to the conventional enhanced Mobile Broadband (eMBB) use cases, Ultra-Reliable and Low Latency Communication (URLLC), e.g., for car-to-x communication as well as large sensor and Internet of Things (IoT) networks will introduce very diverse requirements on the mobile networks of the fifth generation. Network slicing is seen as one of the key enablers to cope with those divergent requirements in existing physical networks [2].

Network slices are virtual networks sharing the same physical network infrastructure. Network virtualization allows to configure individual network slices for different use cases and to overcome the one-size-fits-all approach of current monolithic mobile networks. The mobile network slices provide isolated end-to-end connections, as specified in the associated Service Level Agreements (SLAs) between a mobile service provider and a tenant. Network slices can be permanent or on a short-term basis, but they can also be subject to dynamic instantiation, change and termination.[3]

However, network slicing presupposes that network functions and services can be flexibly allocated and configured for the network slices. This requires best effort on network virtualization and programmability as well as a strong hardware independence of software-based network functions from the physical network infrastructure [4] [5]. Since network slices have to be dynamically adapted to customer and user needs, fast decision making on network resource allocation and quick network slice instantiation is required. Therefore, this paper aims at automating this planning and decision process by modeling the network slice resource embedding on a shared physical network using Integer Linear Programming (ILP). The focus of the model is on resource assignment and on the allocation of network functions, user applications and services on multi-purpose cloud servers. In this way, a nearly optimal network slice embedding on a shared physical network can be calculated for an arbitrary mobile network and arbitrary network slice instances automatically. This embedding answers the question of whether or not the physical network provides enough resources to serve specified network slices in its current configuration as well as deciding which network slices shall be deployed regarding the business goals of the infrastructure provider.

II. Embedding Network Slices onto a Common Mobile Network Infrastructure

The problem of embedding mobile network slices into a common mobile network infrastructure can be modeled in form of a directed graph for each mobile network slice, connecting mobile User Equipments (UEs) with so called application nodes. Application nodes are mobile network services, like telecommunication services establishing a connection to another end-user device or providing a service like video streaming or a virtual reality application. Fig. 1 shows a simplified end-to-end mobile network architecture. The UEs establish a radio connection to one of the radio heads. Those radio heads either have their own base station or are connected to a base station via an optical or directional radio transport

link (remote radio heads). Each base station might own an edge cloud. The base stations are connected to the core network by the so called backhaul. The core network provides access to the central or main cloud. In contrast to the edge clouds, the main cloud has an enormous capacity for running applications and providing services.

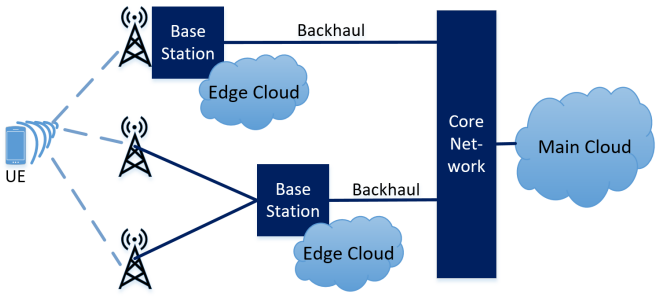

Fig. 1. Simplified End-to-End Mobile Network Architecture

In comparison with the model of the physical network infrastructure, the model of the network slice instances is pretty simple. It only specifies the UEs or UE groups connected to the required applications. Fig. 2 gives an example for a basic network slice with one UE and one application.

Fig. 2. Simplified End-to-End Network Slice

For reasons of simplicity, it is assumed that every application node can be deployed on an arbitrary cloud server as long as the server provides enough computational power and memory resources. This assumption will not hold in a practical context. However, this approach would be capable of handling additional constraints. Throughout this paper it is also assumed that the resources provided by the physical network as well as the resource demands of the network slices remain constant over time. This can be assumed as looking at a snapshot of the system or using average values for the resource provisioning and utilization. However, the signal quality of mobile communication is affected by a large number of environmental influences, like the weather or foliage. As a consequence, the available throughput resources vary over time in practices. Also the actual resource utilization of the users of the network slices is time dependent. For example, the data traffic volume at daytime is very different from the data traffic volume at night. Further research is required on modeling and solving this kind of dynamic resource embedding problem. Nevertheless, modeling and solving this simplified problem will help to understand and solve the core of the problem without being confused by too many technical details in the beginning. This will make future enhancements on the way to a practically feasible approach much easier. Based on this model, the main questions to be answered in this paper are:

1) Are there enough resources in the physical network to embed all network slices?
2) Which network slices shall be chosen for deployment when there are not enough resources?
3) Which cloud servers (nodes) shall the applications be deployed on?
4) Which communication links should be used if there are several alternative paths?

All four of these questions can simply be answered by only one Integer Linear Program. This Program will be used to efficiently determine a nearly optimal embedding of the mobile network slices in a substrate network.

III. Related Work

The challenges related to network slicing are intensively discussed at the moment. Recent related work in the field of network slice admission and resource allocation is, for instance, [6], [7] and [8].

Regarding network slice resource allocation, the paper of Zhang et al. [6] focuses on dynamic optimization of the available throughput for small cells and macro cells by modifying their power configurations. The goal of the paper from Wang et al. [7] is the dimensioning of network slices by determining the optimal resource pricing to simultaneously optimize social welfare and benefit of the network slice provides. The authors of [8] propose an algorithm for the admission of single UEs to a network slice to prevent network overloading at runtime. However, these approaches do not cover the problem of embedding network slices with fixed requirements (SLAs) in a shared network infrastructure. The algorithm proposed in this work is a method of determining whether or not a certain set of network slices can be deployed on a common physical network. If resource shortages are identified, the network slices with highest weight, i.e., the most beneficial or strategically most important ones, are chosen for deployment.

Beyond the ongoing scientific work on all aspects of network slicing, the theory of Virtual Network Embedding (VNE) is pretty well researched. The NP-hard [9] problem can be solved with ILP as well as various heuristics proposed in literature. Those heuristics usually are more efficient but less accurate. The VNE problem is also getting increased attention in practice. The telecommunication industry, amongst others, is adopting network virtualization techniques. For example, Fischer et al. [10] give a survey on VNE for fixed networks. There are also some publications on wireless network embedding, see for instance [11] and [12]. However, there is still a lack of concepts on virtual end-to-end network embedding for mixed wired and wireless physical networks. Richard et. al [13] give a survey on recent work in mobile network slice embedding. But these publications mainly focus on runtime issues of the embedding, for example, resource partitioning and sharing as well as on network slice isolation techniques.

The model proposed in this paper will not cover network slice deployment and issues like network slice isolation as well as Physical Resource Block (PRB) assignment during runtime. However, this paper presents an approach that analyses the

available resources of mixed wired and wireless end-to-end mobile networks and provides a nearly optimal embedding of mobile network slices into this shared physical network. Usually, ILP solutions for VNE only perform reasonably well on small networks. However, network slice embedding is associated with large problem instance as well as numerous parameters and constraints. This approach follows the design guidelines for Integer Programmings (IPs) proposed by Despotovic et al. in [15]. The authors of [15] present a scalable algorithm for the VNE problem, the so called VNetMapper. Their problem formulation allows to solve a VNE with hundreds of nodes and thousands of links within a few seconds. Unfortunately, the VNetMapper is not optimized for end-to-end mobile networks. It only considers consumable resources, e.g., throughput and memory. Constraints for non-consumable resources like latency are not formalized by the approach. Beyond that, the VNetMapper does not allow to map several virtual nodes on the same physical node as well as several virtual links on the same edge of the substrate. This constraint does not seem to be feasible in practice since resource sharing is one of the most important intentions of network slicing in 5G. In contrast to that, the formal model presented in this paper is tailored to end-to-end mobile network slicing. It considers link latency constraints and allows "many-to-one" mappings. Furthermore, this approach takes advantage of the fact that the UE nodes are the same in the physical and the virtual networks.

IV. Model for Embedding Network Slices

In this section, a mathematical model for mobile network slice embedding will be proposed. The model will focus on the resource allocation for the network slices in a shared physical network. This formalization is intended to be solved automatically by a standard ILP solver like for example the GNU Linear Programming Kit (GLPK) library of the GNU Project [14]. The presented model complies with the design guidelines of Despotovic et al. [15] for efficient and scalable modeling of VNE problems.

A. Definitions

To describe the model for network slice embedding the following definitions from graph theory (see [16]) are needed: An undirected Graph G is an ordered pair $G = (\mathcal{V}, \mathcal{E})$ comprising a set of $n \in \mathbb{N}$ vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and a set of $m \in \mathbb{N}$ edges. An edge is associated with two vertices $e_{ij} := \{v_i, v_j\}$ for $i, j = 1, \ldots, n$, often abbreviated by $e_{ij} := v_i v_j$. v_i and v_j are called the ends of e_{ij} (or start node v_i and end node v_j). In undirected graphs edges do not have a direction, i.e., $e_{ij} = e_{ji}$.
A path of length $n \in \mathbb{N}$ is an undirected graph $P = (\mathcal{V}, \mathcal{E})$ such that $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ are pairwise different vertices and the set of edges is $\mathcal{E} = \{v_1v_2, v_2v_3, \ldots v_{n-1}v_n\}$. v_1 is called start-vertex of P and v_n is called end-vertex of P. P can also be written as $P = v_1v_2v_3 \ldots v_n$. $\mathcal{P}_{ij}$ denotes the set of all paths in an undirected Graph $G = (\mathcal{V}, \mathcal{E})$ with start-vertex $v_i \in \mathcal{V}$ and end-vertex $v_j \in \mathcal{V}$ with $i, j = 1, \ldots, n$, $n \in \mathbb{N}$ and $i \neq j$.
A network graph $\mathcal{N} = (\mathcal{U}, \mathcal{C}, \mathcal{E})$ is defined as an undirected Graph $G = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} := \mathcal{U} \cup \mathcal{C}$. $\mathcal{N}$ comprises a set of UEs $\mathcal{U} := \{u_1, \ldots, u_n\}$ with $n \in \mathbb{N}$, a set of cloud nodes $\mathcal{C} := \{c_1, \ldots, c_m\}$ with $m \in \mathbb{N}$ and a set of edges $\mathcal{E} \subseteq \{u_ic_j, c_kc_l\}$ for all $i = 1, \ldots, n$ as well as for all $j, k, l = 1, \ldots, m$ with $k \neq l$.

B. Parameters of the Model

Let's assume that $n \in \mathbb{N}$ network slices, or virtual networks shall be embedded into one physical network. The physical network will also be called the substrate in the following. The substrate network is denoted as a network graph $\mathcal{N} = (\mathcal{U}, \mathcal{C}, \mathcal{E})$ with $u_v \in \mathcal{U}$, the UEs in the physical network, $c_w \in \mathcal{C}$, the cloud nodes in the substrate, applications can be deployed on, and $e_j \in \mathcal{E}$, the communication links (edges) of the physical network. The k-th virtual network, or in this case the k-th network slice, is defined as an undirected graph $\mathcal{N}_k = (\mathcal{U}_k, \mathcal{A}_k, \mathcal{L}_k)$ for $k = 1, .., n$ with $\mathcal{U}_k \subseteq \mathcal{U}$, a subset of all UEs in the mobile network. (Note that the UEs u_v refer to the same instances in the substrate and the virtual network, i.e., the mapping of the UEs of the network slices onto the UEs in the substrate is already fixed.) $a^k_m \in \mathcal{A}_k$ refers to the m-th application node of the k-th network slice. We assume that each slice uses its own application nodes. That means, although the slices might share the same application types they are modeled as different application instances. Finally, $l^k_i \in \mathcal{L}_k$ denotes the i-th virtual communication link of the k-th network slice. A mapping of the applications nodes a^k_m on the cloud nodes c_w as well as a mapping of the virtual links l^k_i on suitable paths of edges e_j has to be determined such that the available resources of the physical network are not exceeded by the network slices. A path $P_r \in \mathcal{P}_{vw}$ is defined as a path with start-vertex $u_v \in \mathcal{U}$ and end-vertex $c_w \in \mathcal{C}$. A link $l^k_i \in \mathcal{L}_k$ can be mapped on a path $P_r \in \mathcal{P}_{vw}$ if $l^k_i = \{u_v, a^k_m\}$ and the application a^k_m has been mapped on the cloud node c_w. In this paper three types of resources, the throughput T of the communication links as well as the computational power P and the storage capacity M (memory) of the cloud server nodes, will be considered. However, this could easily be extended to further resources. For the mobile network slice embedding problem the amount of available resources in the mobile network has to be compared with the amount of demanded resources by the network slices. Therefore, the following parameters are defined: the throughput T^s_j of the physical network denotes the maximum throughput of the mobile edge $e_j \in \mathcal{E}$ of a physical network link, i.e., the maximum throughput that is expected to be transmitted via radio or by wire. For mobile radio communication a fixed bandwidth and the (average) Channel Quality Index (CQI) of the UEs have to be specified in advance. For ease of understanding, Uplink (UL) and Downlink (DL) resources are not differentiated throughout this paper. The power P^s_w denotes the computational power of the w-th cloud node $c_w \in \mathcal{C}$ of the substrate and the storage M^s_w is defined as the memory capacity of the cloud node $c_w \in \mathcal{C}$ in the physical network. For

the resource utilization of the mobile network slice $\mathcal{N}_k$, T_i^k is defined as the required throughput of the i-th communication link $l_i^k \in \mathcal{L}_k$. The required computational power an application node $a_m^k \in \mathcal{A}_k$, the k-th network slice requires, is denoted as P_m^k and its required memory is defined as M_m^k accordingly. The communication latency is another very important constraint of the communication links of the network slices that has to be satisfied by the communication path in the physical network mapped to it. The latency L_j^s of the communication links $e_j \in \mathcal{E}$ is dependent on the throughput. However, it can be assumed that it remains constant as long as the bandwidth is not exceeded. In this paper it will be assumed that the given latencies are upper estimations of the latency of the physical communication links under full load. In addition to that, the allowed maximum latency for each communication link $l_i^k \in \mathcal{L}_k$ of the network slice k is denoted as L_i^k. The latency caused by the applications is not considered here since it is assumed to be constant, i.e., independent of the hardware the application is deployed on.

C. Variables of the Model

To formulate the objective function and the constraints of this optimization problem in form of an Integer Linear Program, the following mapping variables are needed:

$$y_k := \begin{cases} 1 & \text{if } \mathcal{N}_k \text{ is embedded into } \mathcal{N}_s \\ 0 & \text{otherwise} \end{cases}$$

$$a2c_{mw}^k := \begin{cases} 1 & \text{if } a_m^k \text{ is mapped on } c_w \\ 0 & \text{otherwise} \end{cases}$$

$$l2p_{ir}^k := \begin{cases} 1 & \text{if } l_i^k \text{ is mapped on } P_r \in \mathcal{P}_{vw} \\ 0 & \text{otherwise} \end{cases}$$

$$p2e_{rj} := \begin{cases} 1 & \text{if } e_j \text{ is used in } P_r \\ 0 & \text{otherwise} \end{cases}$$

$$l2e_{ij}^k := \sum_r (l2p_{ir}^k \cdot p2e_{rj})$$

Obviously, the virtual link l_i^k is mapped on the physical edge e_j if and only if l_i^k is mapped onto a path P_r that contains e_j. Note that all defined variables are binary. This contributes to a short runtime of the ILP solver.

D. Objective Function

It is crucial to choose an objective function that is as simple as possible, since the runtime of solving an Integer Linear Program is heavily dependent on the complexity of its objective function [15]. Hence, it is better to perform a preprocessing to determine weights ω_k representing, for example, how beneficial the deployment of each network slice $\mathcal{N}_k$ would be for the infrastructure provider. The objective function of the mobile network slice embedding problem could simply be maximizing the weighted sum of all embedded network slices as proposed in [15]:

$$\max \sum_k (\omega_k \cdot y_k)$$

E. Optimization Constraints

The above objective function shall be maximized under the following constraints.

- Map-once and graph constraints

$$\sum_w a2c_{mw}^k = y_k, \forall k, m \tag{1}$$

$$\sum_{P_r \in \mathcal{P}_{vw}} l2p_{ir}^k = a2c_{mw}^k, \forall k, i \text{ with } l_i^k = \{u_v, a_m^k\} \tag{2}$$

- Capacity constraints

$$\sum_k \sum_i l2e_{ij}^k \cdot T_i^k \leq T_j^s, \forall j \tag{3}$$

$$\sum_k \sum_m a2c_{mw}^k \cdot P_m^k \leq P_w^s, \forall w \tag{4}$$

$$\sum_k \sum_m a2c_{mw}^k \cdot M_m^k \leq M_w^s, \forall w \tag{5}$$

- Latency constraints

$$\sum_j l2e_{ij}^k \cdot L_j^s \leq L_i^k, \forall k, i \tag{6}$$

Eq. 1 defines the map-once constraints for the application nodes, similar to [15]. There is one constraint for each application node in each network slice. Obviously, the application to cloud node mapping variable $a2c_{mw}^k$ must sum up to 1 for all application nodes of the slice if and only if the slice has been embedded into the network. Whether or not a network slice is embedded into the substrate is determined by the variable y_k. Eq. 2 specifies the map once constraints for the virtual link to physical link mapping, as already seen for the node mapping in Eq. 1. The second equation states that if the application node $a_m^k \in \mathcal{A}_k$ is mapped on the cloud node $c_w \in \mathcal{C}$, then l_i^k has to be mapped on exactly one path P_{vw} with end node c_w that starts in the same $u_v \in \mathcal{U}_k$ as l_i^k if the k-th network slice is embedded in the substrate. Note that, due to Eq. 1, the k-th network slice is embedded in the substrate if and only if $a2c_{mw}^k = 1$ in Eq. 2. The capacity constraints in Eq. 3 require that the total throughput capacity of each link in the substrate network must not be exceeded by the sum of throughput requirements of all virtual links of the different slices mapped to it. This results in one constraint per physical link. The same goes for the computation power and memory capacity of the cloud nodes in the physical network which must not be exceeded by the mapped application nodes in total (see Eq. 4 and 5). The capacity constraints are similar to the formalization proposed in [15]. Finally, Eq. 6 defines the latency constraints. The allowed latency of each virtual link in any of the network slices must be greater or equal to the sum of the actual latencies of all physical edges contained in the path mapped to this virtual link. Consequently, there is one constraint for each virtual link in all network slices.

V. Solving the Mobile Network Slice Embedding Problem and Model Validation

This section shows how the model introduced in section IV can be set up and solved for a simple example. The following example is intended to give a better intuition and understanding of the model.

Given a simple substrate network and two basic mobile network slices, as depicted in Fig. 3, 4 and 5. The substrate $\mathcal{N} = (\mathcal{U}, \mathcal{C}, \mathcal{E})$ defines two UEs $\mathcal{U} = \{u_0, u_1\}$, each of them can also be interpreted as a group of UEs located in the same cell. $\mathcal{N}_s$ also defines three cloud server nodes $\mathcal{C} = \{c_0, c_1, c_2\}$ connected by five edges $\mathcal{E} = \{e_0, e_1, e_2, e_3, e_4\}$. The physical

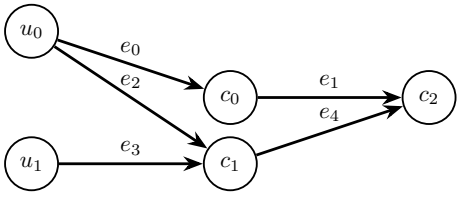

Fig. 3. Example of Simple Substrate Network

network in the example shown in Fig. 3 has six paths, that have to be considered by the mobile network embedding model. These paths are $P_0 = e_0$, $P_1 = e_0e_1$, $P_2 = e_2$, $P_3 = e_2e_4$, $P_4 = e_3$ and $P_5 = e_3e_4$. Two network slices

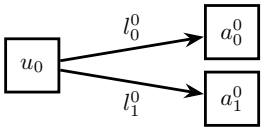

Fig. 4. Example Network Slice 0

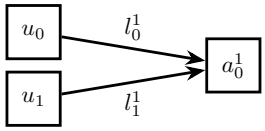

Fig. 5. Example Network Slice 1

$\mathcal{N}_0 = (\mathcal{U}_0, \mathcal{A}_0, \mathcal{L}_0)$ and $\mathcal{N}_1 = (\mathcal{U}_1, \mathcal{A}_1, \mathcal{L}_1)$, as shown in Fig. 4 and Fig. 5, shall be embedded in $\mathcal{N}$. Only u_0 belongs to the first slice, whereas the second slice is dedicated to both UEs. The network slice $\mathcal{N}_0$ defines two application nodes $\mathcal{A}_0 = \{a_0^0, a_1^0\}$ connected by the virtual links $\mathcal{L}_0 = \{l_0^0, l_1^0\}$. The two links $\mathcal{L}_1 = \{l_0^1, l_1^1\}$ of $\mathcal{N}_1$ are both pointing to the only application node of the second slice $a_0^1 \in \mathcal{A}_1$. If the weights of the network slices are $\omega_0 = \omega_1 = 1$, the objective function is: $\max(y_0 + y_1)$. Tab. I specifies the throughput and latency for all links and Tab. II defines the CPU power and the memory capacity for all nodes. Fig. 6 shows the calculated embedding of the two mobile network slices. Both network slices can be embedded into the substrate network with a_0^1 mapped on c_1 and a_0^0 as well as a_1^0 mapped on c_2. The link l_0^0 is embedded on the path $P_1 = e_0e_1$, while l_1^0 uses $P_3 = e_2e_4$, l_0^1 is mapped on $P_2 = e_2$ and l_1^1 must use $P_4 = e_3$. Since both network slices have been embedded in the substrate, this is an optimal solution regarding the objective function.

The example explained above also serves as one of the test cases used to ensure the correctness of the proposed model as well as its implementation.

TABLE I
Network Link Parameters

Link	T	L
e_0	10	5
e_1	8	1
e_2	10	20
e_3	2	1
e_4	10	1
l_0^0	3	18
l_1^0	7	30
l_0^1	1	30
l_1^1	2	5

TABLE II
Network Node Parameters

Node	P	M
c_0	20	100
c_1	40	300
c_2	2000	10000
a_0^0	10	90
a_1^0	300	15
a_0^1	40	150

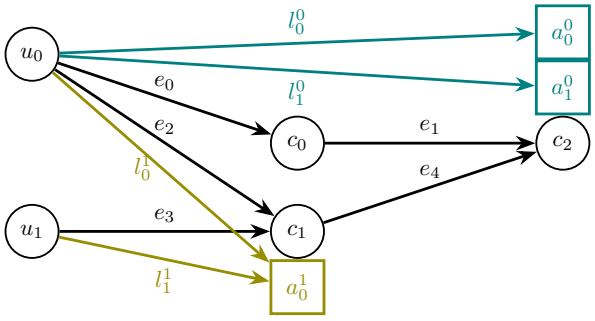

Fig. 6. Example of Simple Network Slice Embedding

TABLE III
Model Evaluation

ID	Substrate			n	Average of Slices			time
	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{E}\rvert$		$\lvert\mathcal{U}_k\rvert$	$\lvert\mathcal{A}_k\rvert$	$\lvert\mathcal{L}_k\rvert$	
1	200	61	399	3	12	2	53	21s
2	200	61	381	5	24	2	54	59s
3	300	61	545	3	17	3	72	76s
4	150	111	387	3	17	4	96	41s
5	150	131	407	5	6	4	22	25s
6	140	151	437	5	2	4	26	37s
7	150	161	457	5	4	4	23	28s
8	180	191	559	5	2	4	18	43s

A dedicated simulator, written in Java programming language, has been implemented for setting up and solving the mobile network slice embedding problem with ILP. Our implementation uses the SCPSolver [17], a Java interface for ILP which is based on the GLPK [14].

Tab. III shows an evaluation of the actual runtime of the ILP solver for randomly generated substrates and network slices. The computations have been done on a MacBook Pro 2015 with a 3,1 GHz Intel Core i7 and a 16 GB, 1867 MHz DDR3. The solver was running on one of the two equally sized cores only. Eight physical network model instances have been generated randomly. They consist of between 140 and 300

UEs (or groups of UEs located in the same cell), between 61 and 191 cloud server nodes and contain between 381 and 559 physical communication links. The number of the UEs/UE groups and the nodes as well as links of the network slices are given in average over the embedded network slices. Embedding between 3 and 5 network slices with up to 24 UEs/UE groups per network slice (in average) and between 2 and 4 application nodes per slice (in average), connected by between 18 and 96 virtual links per slice (in average), takes only between 21 and 76 seconds. These results show that the proposed approach is feasible and efficient on large problem instances.

VI. Conclusion and Future Work

In this paper the mobile end-to-end network slice embedding problem is formalized in form of an Integer Linear Program. The program is suitable for solving large problem instances under various resource constraints and other restricting parameters, like latency requirements. The validation of the model with a dedicated Java tool shows its feasibility. In addition to that, the proposed formalization is easy to understand and easy to use with a standard ILP solver. Due to using only binary variables throughout the model and a simple objective function, the algorithm is efficient and scales for large problem instances. In addition to that, the model is tailored to mobile end-to-end network slice embeddings making its solution even more efficient without requiring any "not-many-to-one" constraints. When advancing into a more practical implementation of the model and algorithm, the number of restricting parameters as well as the number of embedding variables will rise significantly. However, the proposed formalization can easily be extended to handle those additional constraints. For instance, the latency of the execution of an application on a specific server hardware is an important parameter that has not been modeled so far.

Future work will include the consideration of more advanced objective functions, for instance, taking differing costs of cloud servers and communication links into account or minimizing latency of the communication. However, these changes have to be evaluated carefully, since they might cause a tremendous increase in runtime, especially for solving large problem instances. In addition to that, network slices usually are not static, they will be deployed, run and terminated dynamically. Thus, their resource utilization will change from time to time and what is more, network slices can also be defined to be only active during specific time windows, e.g., a certain time of the day. This gives the resource allocation and planning performed with the proposed ILP algorithm an additional time dimension. Beyond that, future work will include enhancing the model with further details that need to be considered when embedding network slices into a common physical network, for instance, network function chaining.

References

[1] Ericsson, "Future mobile data usage and traffic growth", https://www.ericsson.com/en/mobility-report/future-mobile-data-usage-and-traffic-growth, accessed Feb. 21th, 2018.

[2] Nokia, "Dynamic end-to-end network slicing for 5G", Espoo, Finland, 2016.

[3] 3GPP, "Study on management and orchestration of network slicing for next generation network", TR 28.801 V15.0.0T, 2017.

[4] 5G NORMA, https://5gnorma.5g-ppp.eu, accessed: Nov. 24th, 2017.

[5] 5G NORMA, "5G NORMA network architecture - intermediate report", Deliverable D3.2, 2017.

[6] H. Zhang et al., "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges" in IEEE Communications Magazine, 2017.

[7] G. Wang et al., "Resource Allocation for Network Slices in 5G with Network Resource Pricing" in IEEE Global Communications Conference, 2017.

[8] M. Jiang, M. Condoluci and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems" in European Wireless 2016, Oulu, 2016, pp. 1-6.

[9] D. Andersen, "Theoretical approaches to node assignment", Computer Science Department, 2002, Unpublished Manuscript.

[10] A. Fischer et al., "Virtual Network Embedding: A Survey" in IEEE Communication Surveys & Tutorials", Vol. 15, No. 4, 2013.

[11] R. Riggio et al., "Scheduling Wireless Virtual Networks Functions" in IEEE Transactions on network and service management, Vol. 13, No. 2, 2016.

[12] I. Tsompanidis, A. Zahran and C. Sreenan, "A Utility-based Resource and Network Assignment Framework for Heterogeneous Mobile Networks" in 2015 IEEE Global Communications Conference, San Diego, 2015.

[13] M. Richart et al., "Resource Slicing in Virtual Wireless Networks: A Survey" in IEEE Transactions on Network and Service Management, VOL. 13, NO. 3, 2016.

[14] Free Software Foundation, "GLPK (GNU Linear Programming Kit)", https://www.gnu.org/software/glpk/, accessed Feb. 21th, 2018.

[15] Zoran Despotovic et al., "VNetMapper: A Fast and Scalable Approach to Virtual Networks Embedding", 2014 23rd International Conference on Computer Communication and Networks (ICCCN), Shanghai, 2014, pp. 1-6.

[16] Reinhard Diestel: "Graphentheorie", 3rd edition, Springer, Heidelberg, 2006.

[17] Hannes Planatscher and Michael Schober, "SCPSolver - an easy to use Java Linear Programming Interface", http://scpsolver.org, accessed Feb. 25th, 2018.