Efficient Slice Allocation for Novel 5G Services

Min Kyung, Lee
Dept. Computer Science and Engineering
Kyung Hee University
Yongin, Republic of Korea
minkyung0110@khu.ac.kr

Choong Seon, Hong
Dept. Computer Science and Engineering
Kyung Hee University
Yongin, Republic of Korea
cshong@khu.ac.kr

Abstract— **The unprecedented growth of novel applications and tsunami of connected devices in the existing networks have forced the service operators to find novel solutions to fulfil the Quality of Service for end users. Network slicing has emerged as a promising solution to fulfil these stringent requirements. However, static network slicing do not scale well for heterogeneous requirements imposed by the services. In this paper, we design a novel approach based on multiple-choice knapsack problem that maximizes the utilization of these network slices. Simulation results reveal that our proposed solution can achieve low latency and high throughput compared to the static network slice approach.**

Keywords—5G Network; Network Slicing; Macro base station; Knapsack problem; Multiple-Choice Knapsack problem

I. Introduction

The proliferation of novel network services in existing networks is emerging at a very rapid pace to support novel industrial verticals such as smart factories, virtual reality and etc. Moreover, these services pose heterogeneous requirements based on their operations opposed to traditional wireless services such as voice and data. Thus, service providers are bound to fulfill these stringent requirements with their limited resources. However, there exists a big challenge pertaining to limited spectrum and computation resources available in the network such as macro base station, an edge server, a core network, etc. To solve this problem, the concept of Network Slicing is introduced in 5G networks[1].

Network Slicing, used on 5G services, is a physical network that is logically separated into end-to-end networks including network devices, access, and core networks. This allows the network slices to provide a dedicated network for various services with different characteristics. Moreover, each slice is insulated from each other, so that even if an error or a failure occurs in a specific slice, the communication of the other slice is not affected with any failure[2].

Network Slicing in the networks can be enabled via Network Functions Virtualization (NFV) and Software Defined Networking (SDN). However, before emergence of NFV and SDN, there was a concept of Network Virtualization (NV). It was an overlay or tunnel. However, instead of physically connecting two domains in the networks, two domains were connected by creating a tunnel through the existing network. NV is easier for an administrator who manages and controls it[3].

As more advances in NV, NFV is a technology applied to wired and wireless network equipment, technology to implement hardware-centered network function with software. It is also critical technology to comprise Network Slice with Software Defined Network which is a software-based controller that controls or manages traffic forwarding operation in the network for flexible use[3].

Moreover, the Network Slices can be categorized into two types: Vertical and Horizontal types. First, the Vertical type, usually used by Industrial area, means that one slice can include different network component. However, the Horizontal type, usually used on user or machine area, means that there are different types of slices to meet each of specific purposes. Additionally, the goal of network slicing is to make network flexible and enhance utilization of network resources. Moreover, it is the virtualization of network resources, which enable operators to assign the physical resources in a flexible way to use the available resources more efficiently. As depicted in Fig.1, the model is a standard figure of Network Slice in a case of vertical type. The system has Control Plane, User Plane, and Server for each services[2].

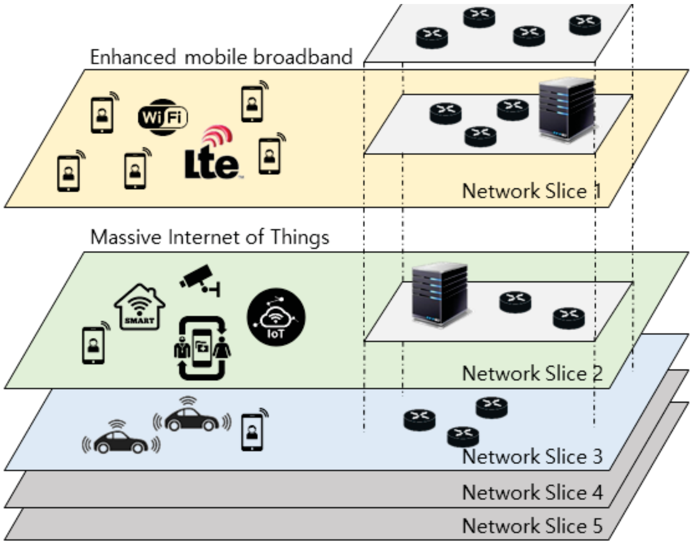

Fig. 1. Network Slice standard sample.

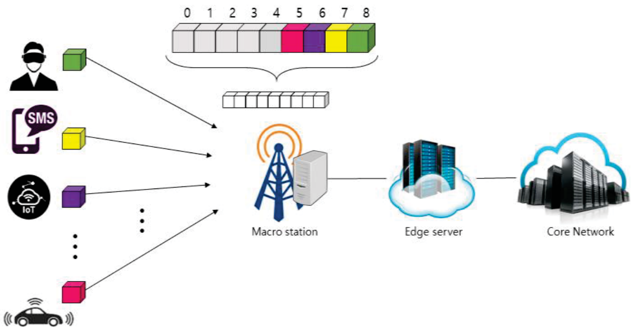

Fig. 2. Network Slice allocation figure

The system has one Control Plane to control all User Plane which has each of Server Gateway, Public data network (PDN) Gateway, and Edge Server. As enhanced Mobile Broadband, Massive IoT, and Mission-critical IoT appear, the slicing model can be utilized to make an efficient usage of resources in 5G which does not need to set more frequency bandwidth.

In this paper, we propose a way to maximize the number of network slice in limited network resources. When a macro cell already allocated its network slices to some services, there are still many services waiting a connection with its network. Thus, the macro cell divides its network slice more to be allocated by a number of services. While there are many papers which consider Network Slice Allocation system, they focused on the Core Network which is not easy to improve because of a cost. Therefore, we focus on Network Slice system of the base station at this point.

Our proposal is based on the idea that the amount of network resources provide a proper resource for service providers which have different requirements of network resources. However, there is a limitation that the more devices are, the more network resources should set. Then, 5G network reduces its benefits such as flexibility, more efficiency, high frequency, more capacity, etc. Thus, we pack the services' items with weights and values to be suitable to given network resources. Moreover, to solve the problem, we consider a vertical sector in the slice environment which is composed of a macro base station, an edge server, a backhaul, and a core network. In addition, we apply the multiple choice knapsack problem to make an optimal packing system on our scenario.

The remainder of this paper is organized as follow. In section II, we review a state of the art allocation of the Network Slice and introduce the Multiple-Choice Knapsack problem. Section III elaborates our system model about how to pack given services' resources to fixed weighted network resources. After explanation of the system model, we will describe the results of our proposal in Section IV. Finally, summary and future way of our work is given in Section V.

II. Related Works

While, from the current technology's point of view, Network Slicing is a theoretical step, many researches are looking forward to make a commercial product of the Network Slice for 5G systems.

An efficient approach for slice allocation based on services has been proposed in [7]. The proposed system consider a way to modify the Core Network. To adjust the Core Network, they set the Slice Selection Function (SSF) on the base station, which acts to sort out raw data from User Equipment Type before going to Mobility Management Entity (MME), Serving Gateway (SGW), and PDN Gateway (PGW) which are components of the Core Network. Then, through the SSF, given services are assigned a slice with the appropriate requirements. However, in such models, it only considers the Core Network, which are generally controlled by the network companies [7].

Similarly, the work in [6] proposed a Slice selection as an aspect of Internet of Things (IoT) services to reduce unexpected congestion of 5G network. It set up the 5G massive IoT Slice to control IoT traffics in IoT Gateways before uploading to IoT Broker. Also, the proposed model forms the IoT Broker and 5G Network Slice Management to communicate and orchestrate IoT traffic and network slicing operations.

Furthermore, in [8], the authors proposed a resource allocation model for network slices in 5g network. In this paper, authors considered multiple network slices are made by the SDN and NFV system with a distributed algorithm to solve the Slice Provider's Problem to increase the net welfare and the profit of slice customers.

However, the above papers are focused on the Core network only and have ignored the access networks. Solutions presented for core networks cannot be applied to access networks due to different dynamic of access networks.

Moreover, as depicted on Fig.2, the static way of network slice does not consider requirements of each services to allocate the slice from low latency and high throughput points of view. Therefore, we address the problem of maximizing the slice allocation in the access network, i.e., a macro cell. To propose the model, we expect to improve the flexibility and latency.

The knapsack problem is a problem in combinatorial optimizations. A concept of the knapsack problem is an algorithm that efficiently packs a given a set of items with weight and value into a knapsack with a fixed weight that can hold items [4].

$$maximize\ z = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m_i} p_{ij}x_{ij} \qquad (1)$$

$$subject\ to,\quad \sum_{i=1}^{n}\sum_{j=1}^{m_i} w_{ij}x_{ij} \le c \qquad (2)$$

$$x_{ij} \in \{0,1\}, i = 1, \cdots, n\ , j \in m_i$$

In this paper, we applied the Multiple-Choice Knapsack Problem, defined as a 0/1 Knapsack Problem which is used when we do not divide a set of items to more small units with NP-hard and can be solved in pseudo-polynomial time by dynamic programming. It need to choose an item from each 'n' groups. The amount x_{ij} of each selected item is measured by a real number between 0 and 1.

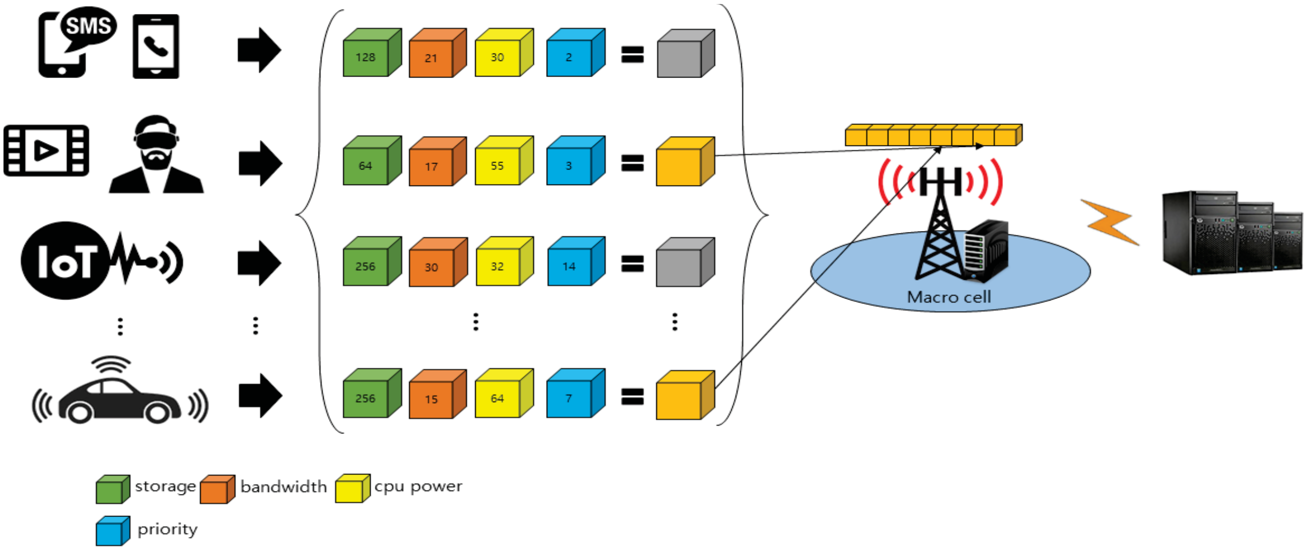

Fig 3. Proposed system model in Network Slicing

III. Proposed Algorithm

Fig. 2 depicts the traditional approach for slice allocation. In this approach, the services are fulfilled without considering their priory, i.e., each services is scheduled based on its order of arrival. Thus, some delay-constrained services may need to wait to get processed such a camera sensor in an autonomous car, or a gas sensor in a smart home, etc. Moreover, if the Macro cell can support a number of services, but other services keep sending signals to use the cell's network slice, then in this case, the cell would take few services. Because, the cell already took services in the order of arrival requests. Therefore, we propose a new approach in which delay-tolerant services can also be handled by the network.

As shown in Fig.3, in the system model, a base station in a macro cell has an important role to divide many slices to take heterogeneous services. At this time, we consider that a base station can reuse its network slice more efficient way to get many services. Thus, we use one of the combinatorial optimization, Multiple Choice Knapsack Problem. Because when a base station consider many requirements of each services, it may lead to more low latency and high throughput.

In contrast to Fig.2 model, we consider multiple requirements for each service such as storage capacity, CPU power, and bandwidth. By considering different requirements, we have an option to decide efficiently in respect to which service to process at the Macro cell based on current resource availability at the macro cell. Thus, among remaining network slices, more priority and efficient services may be processed at the macro cell. To decide in such way, we apply the Multiple-Choice Knapsack Problem to calculate multiple requirements of each service. Also, by using Multiple-Choice Knapsack Problem, the cell may allocate its remaining network slice more efficiently than before, because the cell is aware of its remaining slices and also aware of the quality of service requirement for various services such as Internet of Things, autonomous car, virtual reality, etc. Also, Fig.4 show a specific system model how to operate the proposed model in the macro cell.

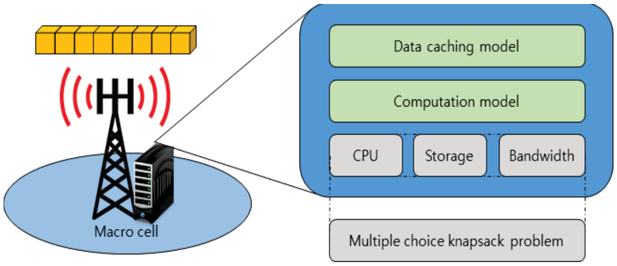

Fig. 4. System module in macro cell

According to the equation (1) and (2), we can set an algorithm of our system model. First, we can divide three classes of service features: Storage, CPU power, and Bandwidth. Each of features have the same number of services.

$$maximize\ z = \sum_{i=1}^{n}\sum_{j=1}^{m_i} p_{ij}x_{ij} \qquad (1)$$

$$subject\ to, \sum_{i=1}^{n}\sum_{j=1}^{m_i} w_{ij}x_{ij} \ \le \ c \qquad (2)$$

$$x_{ij} \in \{0,1\}, i = 1, \cdots, n\,, j \ \in \ m_i$$

Then, we calculate a feature's weight and decide whether the service can be processed into the macro cell or not. If a service can use Network Slice, then x_{ij} will get number one to multiplex with the feature's weight. However, if a feature's weight is quite huge, so total capacity of macro cell's base station does not accept its requirement, then a binary function will be zero to exclude the service from the macro cell.

TABLE I. DESCRIPTION OF THE FUNCTION'S VARIABLE

variable	description
n	number of classes in the services
m_i	number of services
$\mathcal{P}_{ij}$	profits of each services with classes
x_{ij}	binary variable whether services are in or not
w_{ij}	weight varaible of services
c	total capacity of macro cell's base station

However, the most important function from the proposal is to maximize profits of each services with a feature's weights. Because, we should put services which have more profits with proper feature weights to the macro cell. When a service have a large profit but its weights are too large to decrease efficiency and reduce the allocation number of network slice, we should not choose the service to pack into the macro cell. This means that more profits do not always have a top priority among all services.

Algorithm process of maximum allocation of network slices

```
NClass ← number of feature classes that services have in common
MItem ← total number of services
profit_service← a profit of each services
x_num_service ← a service number loaded in to Macro cell
weight_service← a weight of each services
C_apacity ← total weight capacity in Macro base station
IF MItem has each NClass
IF each of NClass has each of different profit_service of MItem
calculate profit_service and weight_service in NClass's MItem
IF profit_service and weight_service are calculated
IF the result is smaller than C_apacity
FOR calculating weight_service with x_num_service
IF the results is more than C_apacity
STOP transferring the Knapsack
ELSE IF
keep going transferring until under C_apacity
```

To be more specific, the macro cell was already used by many services. However, the case is that the macro cell should take other services on its network slice. In this situation, we can consider to set more macro cells near an area of services. However, to set the more cell, is not an economical solution for the network providers. Thus, we propose an efficient way to maximize the reusability of network slices with minimum cost in limited network. From the proposed model, we expect that efficient slice allocation may reduce more latency on the Macro cell.

IV. PERFORMANCE EVALUATION

To evaluate the system model, we use dataset from Udacity Open Sourcing[6] and Data Science Central[7]. From Udacity Open Sourcing dataset, we can apply 223GB driving data. Also, from Data Science Central, we can test the evaluation with IoT Sensor dataset. In addition, we use MATLAB 2018-a version to make a configuration of network slices in 5g network, especially Macro cell.

We investigated two types of results: Throughput (Bytes) and Latency between the proposed way to allocate Network Slice and the static way to allocate Network Slice. First, from the throughput result point of view, we can decide that our proposed model consider heterogeneous services to pack into the Network Slice which was already using. Second, in a perspective view of Latency result, it shows about high performance rate except considering a performance of the service which was already in the Slice. Low latency means that the model can be applied on the 5G network area with various services: massive-Internet of Things, mission-critical Internet of Things, and enhanced Mobile Broad Band.

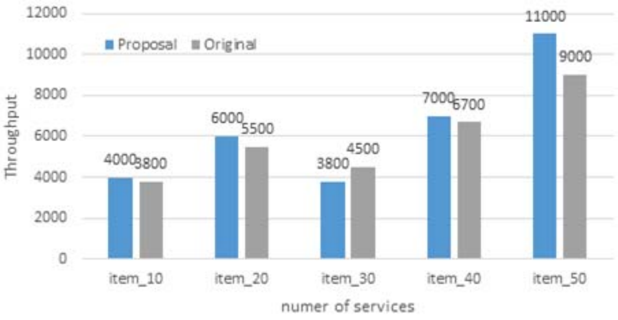

Fig. 5. Throughput of the number of services

Figure 5 shows the Throughput between the proposed model and original model. In this case, our proposed model has a higher throughput than the original. This is because our proposed model can handle heterogeneous requirements set by the service. The proposed algorithm can utilize the network slices efficiently and accommodate more services compared to the traditional approach. However, the gap of throughput is not large, because we consider only Macro cell.

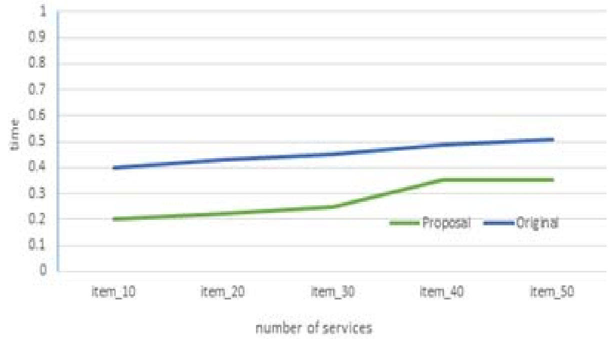

Fig.6. Latency on the number of services

Furthermore, as shown in Fig. 6. Our system model has lower latency in the same circumstance, because, we considered CPU power, Storage, and Bandwidth to pack the services into the Macro cell. Thus, when each services want to access the Macro cell, the cell may decide with the three requirements of the services and its available amount of network slices. In addition, in this case, the cell checks a profit and weights of each services. Then, by using the Multiple Choice Knapsack Problem, the cell calculate the combination of a service's profit and weight. Thus, if the cell has available network slices and also the service offers profits, then, the cell accept the service into its network slice. However, original one has high latency, because, in this case, the macro cell and services both do not consider their network requirements and available slices. Also, the cell does not cater for profit of each services.

V. Conclusion and Future Work

In this paper, we address the problem of efficient utilization of network slices. We considered using the solution of Multiple Choice Knapsack Problem to pack more services into the network slice. We considered the parameters of CPU power, Storage, and Bandwidth of each services in our model. We achieved high performance in terms of lower latency and higher throughput compared to the traditional static approach. In future, we intend to extend our proposal for femto cell and other small cells. Moreover, we intend to apply a deep learning algorithm to further improve the network slicing utilization. By using a deep learning, we can learn the repeated behavior on a process of slice allocation. With the repeated behavior, the model can recognize which services should be served. Thus, the model can allocate network slices by itself and assign slices to services with more flexibility.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government(MSIT) (No.2015-0-00557, Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices). And also, This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2017R1A2A2A05000995).*Dr. CS Hong is the corresponding author.

References

[1] H. Zhang, N. Liu, X. Chu, et al. "Network Slicing Based 5G and Future Mobile Netwokrs: Mobility, Resource Management ,and Challenges", IEEE Communications Magazine, vol. 55, Issue. 8, 2017, pp. 138-145

[2] Xenofon foukas, Georgios Patounas, Ahmed Elmokashfi, Mahesh K. Marina, "Network Slicing in 5G: Survey and Challenges", IEEE Communication Magazine, DOI: 10.1109/MCOM.2017.1600951, pp 94-100, 12, May, 2017

[3] Choong Seon Hong, S.M. Ahsan Kazmi, Seungil Moon, and Nguyen Van Mui, "SDN Based Wireless Heterogeneous Network Management", the 2nd International Conference Advanced Engineering - Theory and Applications 2015(AETA 2015), Dec 9-11, Ho Chi Minh city, Vietnam

[4] T. Ibaraki, T. Hasegawa, K. Teranaka et al. "The multiple-choice knapsack problem", Journal of the Operations Research, Society of Japan, vol.21, No.1 Issue. 1, March, 1978, pp.59-93.

[5] David Pisinger, "A minimal algorithm for the Multiple-Choice Knapsack Problem," European Journal of Operational Research, vol. 83, Issue. 2, pp. 394-410, 1995.

[6] V. Sciancalepore, F. Cirillo, X. Costa-Perez, "Slice as a Service (SlaaS) Optimal IoT Slice Resources Orchestration," GLOBECOM 2017.

[7] M. Sama, S. Beker, W. Kiess, et al, "Service-based Slice Selection Function for 5G," IEEE Global Communication Conference, GLOBECOM 2016

[8] Gang Wang, Gang Feng, Wei Tan, Shuang Qin, Ruihan Wen, and San Shan Sun, "Resource Allocation for Network Slices in 5G with Network Resource Pricing", IEEE Global Communication Conference, GLOBECOM 2017.

[9] Oliver Cameron, "Open Sourcing 223GB of Driving Data", UDACITY, Available on: https://medium.com/udacity/open-sourcing-223gb-of-mountain-view-driving-data-f6b5593fbfa5

[10] Vincent Granville, "Great IoT, Sensor and other Data Sets Repositories", Data Science Central, 2015, Avaialbe on: https://www.datasciencecentral.com/profiles/blogs/great-sensor-datasets-to-prepare-your-next-career-move-in-iot-int