Network Slice Mobility for 6G Networks by Exploiting User and Network Prediction

Hao Yu*, Zhao Ming*, Chenyang Wang†, and Tarik Taleb*
*Oulu University, Oulu, Finland.
†Tianjin University, Tianjin, China.

Abstract—**Beyond 5G applications, future 6G services would need to support very large data volumes for emerging industry verticals, such as holographic-type communications, as well as time-sensitive services, e.g., industrial control. Network slicing is the key technology to deliver such customizable services. Slices and their dedicated resources should be provisioned optimally where the services will be run with low network latencies and associated expenses. However, the user dynamics on resource demands within and between slices result in different resource re-allocation triggers, ultimately lead to distinct mobility patterns, e.g., scaling, migration, where sufficient resources must be transferred. Efficient slice mobility requires increasing flexibility in network operation and management to ensure the customized QoS while minimizing the corresponding mobility cost. In this paper, a prediction-based intelligent network analytic is proposed to facilitate the optimized network slice mobility scheme. We will investigate how to utilize the user and network prediction as the auxiliary information to make the slice mobility decision with the objective of maximizing the long-term profits while minimizing the latency and mobility cost. Finally, we evaluate the proposed prediction-based network slice mobility scheme in a simulated environment and compare its performance in terms of system costs, revenues, and profits with two benchmark solutions.**

I. Introduction

6G, as compared to current fifth generation (5G) wireless networks [1], goes beyond just meeting the KPI criteria for higher data rates, larger network capacity, and reduced latency. It will confront much more challenging situations which expect the following special characteristics within 6G networks. First, it will be an open communication ecosystem including radio access networks (RAN) and core networks for the various communication types and applications, e.g., holographic-type communication. Each service of a new vertical industry, e,g., automotive, smart medical and AR/VR, imposes unique requirements, which will motivate decoupling logical network functions from the physical infrastructure to form the self-contained, programmable and highly customizable networks. Second, to achieve the goal above, the resource virtualization using network slicing technique, which facilitate the advanced network virtualization, will demand higher network management level. Third, ubiquitous intelligence from end users to the network edges, boosts the intelligent network/performance analysis which will not only advance the network management level in terms of QoS guarantee and cost reduction, but also bring considerable challenge to coordinate multiple intelligent network nodes.

Network slices [2], represented by logically independent and self-contained networks running on software-based smaller modular network functionalities with varied granularity, are sufficiently adaptable and flexibly configurable to concurrently support a variety of business-driven 6G use cases over the same network infrastructure [3]. In this paper, we will refer to a network slice as an logical network consisting of a set of virtual network functions that operates on top of a common underlying network, is completely autonomous in terms of its management and control, and can be flexibly programmed to satisfy the service level agreement (SLA) and ensure the deterministic network performance [4] associated with a particular type of service. The computing and storage resources within a network slice are connected via virtual networks and connections. Multiple service requests issued by different mobile devices (MDs) are able to run simultaneously within one slice, e.g., video streaming services.

To achieve the necessary SLAs of services, network slices are provisioned with specific resources that are only used for that purpose. Such slice-specific resources should be coupled with services, as well as the corresponding mobile users, during the whole lifetime of the slices. Especially for some dynamic scenarios where the user demands rise or drop, it needs strict slice adjustment and migration strategies, for both the network resources and services running on them, which can be referred as network slice mobility (NSM) [5]. While a slice is migrated from one service region to another, all of these factors, i.e., service migration and network resource migration must be considered to guarantee service continuity. As a further step, the authors leveraged the reinforcement learning (RL) method in [6] to determine the trigger selection for different NSM options, i.e., slice mobility, slice splitting, slice merging and slice scaling, by analyzing the user demand pattern. Although, using RL method can bring intelligence to the network by solving some complicated scheduling problems, e.g., the trigger selection for NSM, there still exists lots of challenges that need to be addressed. For example, it generally takes a long time to train the RL agent before it works well. The trained agent has to adapt to the realistic production environment to work optimally if the agent is trained based on a simulation environment, since the network topology and the distribution of user demand in realistic networks environments may be different from the simulated environment and vary over the time. To cope with these issues, a intelligent NSM strategy that can leverage directly the realistic network information, including user, slice and network state information is exigent. Whether a mobility action, e.g., migrating a network slice,
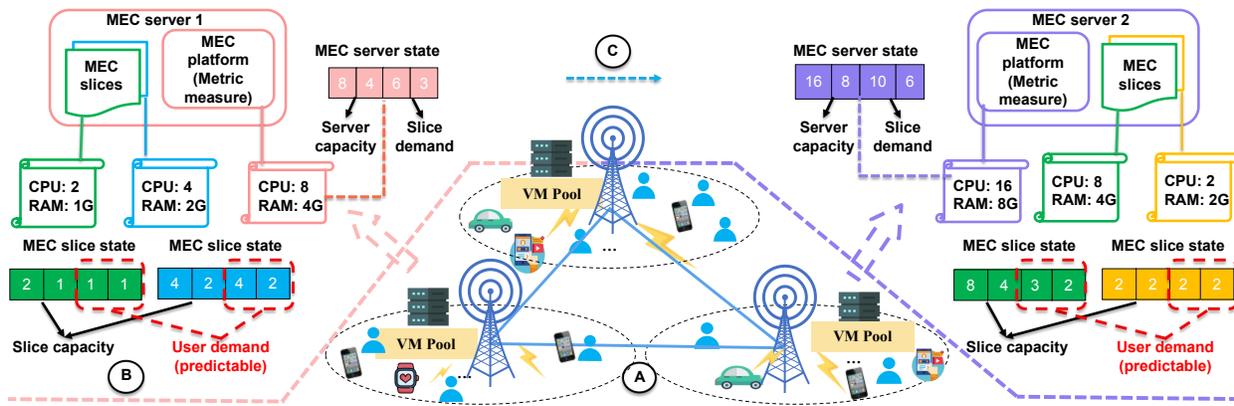
Fig. 1. Network representation for network slice mobility architecture

is optimal globally depends on if it has not only spatially, but also temporally global view of the network environments. Therefore, we introduce a user and network prediction-based NSM approach by taking advantage of the user and network prediction to facilitate intelligent trigger selections for the NSM. Existing prediction-based solutions in [7] [8] which introduced a predictive approach based on CPU load variation to detect over-utilized and under-utilized servers and schedule the Virtual Machines (VMs) migrations and determine optimal decisions. Although both studies considered using prediction to make decision on VM placement or migration, they neglected the internal user demand of a service and averted the actions such as scaling up/down various resource types, e.g., CPU, RAM to avoid performance degradation of applications with dynamic workloads. In this paper, we take into account the user, slice and network resource status (physical and virtual) comprehensively in a mobile edge computing (MEC) environment [9] and propose a user and network prediction-based NSM scheme by analyzing the historical user demand, resource utilization of physical nodes and network slices. The proposed prediction-based heuristic scheme utilizes network and data analytic functionality [10] and ubiquitous intelligence inside the networks to support the advancement of 6G network operation and management. Based on these observations, the contributions of this article are:

- We introduce different slice mobility patterns to correspond to the changing user demand and network status;
- We design a user and network prediction-based NSM strategy to optimally provision and manage network to satisfy the QoS while maximizing the system profits;
- Extensive simulation results demonstrate that the proposed solution outperforms two benchmark solutions.

The remainder of this paper is organized as follows. Section II presents the system model and formulates the NSM problem. Our proposed prediction-based solution is presented in Section III. The performance evaluation results are discussed in Section IV. Section V concludes this paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

Fig. 1 depicts an overview of the envisioned architecture which consists of two levels of networks, i.e., physical layer and slice layer. For the sake of simplicity, we assume that a network slice consisting all the necessary network functions, e.g., radio access network (RAN) functions, can be deployed in a single MEC server. This layering concept conforms to ETSI's MEC and network function virtualization (NFV) standards [11]. The infrastructure layer consists of a set of MEC nodes associating with cellular base stations (BSs). The wireless connections between MDs and networks should be established through the BSs, while each MEC node is comprised of CPU and RAM resources to support the computing and storage tasks of mobile services. On the other hand, the virtual network functions (VNFs) instantiated based on the NFV paradigm can form the network slices that server multiple mobile users, which are also featured by a certain amount of CPU and RAM capabilities. We define two types network representation, i.e., MEC server state, MEC slice state, to effectively express the network status. Different slice mobility patterns will be triggered according to the MEC server state and MEC slice state. Similar to the definition in [5], the triggers in this paper can be classified into two types as follows:

1) Slice Resource Trigger (SRT): the SRT trigger focuses on the performance of a single network slice, i.e., the internal resource consumption of the MEC applications. The primary objective of designing this trigger is to monitor the slices themselves, allowing for more flexibility and exploration of a wider variety of new activities, such as scale up/down operations. Moreover, these information may be expanded to include other characteristics, such as the number of requests/MEC applications. To this end, each MEC server transmits its local data to the orchestration layer for the slice demand data analytic.

2) Slice Migration Trigger (SMT): this trigger is concerned with the aggregate infrastructure-level resources associated to the underlaying nodes, i.e., MEC servers, hosting virtualization
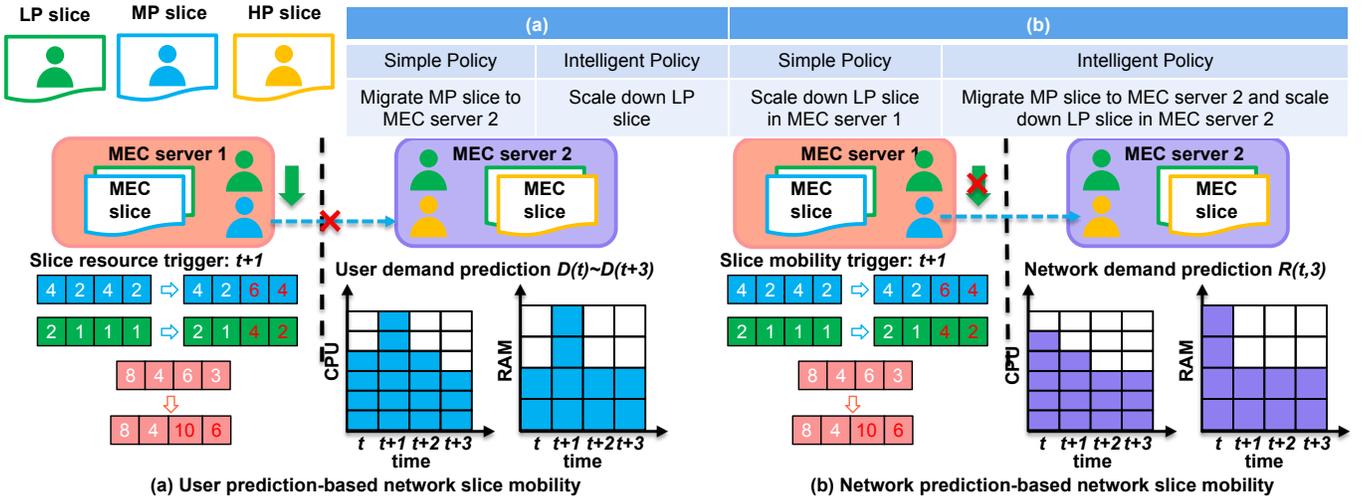
Fig. 2. The motivational example of prediction-based NSM scheme

instances. The SMT trigger may result in slice migration based on the CPU and RAM utilization of the MEC server, if the current residual resources in a MEC server cannot support the scaling up of some slices. All the slice demands will be collected to compose the MEC server state, the proportion of each MEC server's used resources after receiving information from all the MEC servers constitute the network environment. The system will decide whether and where to migrate slices based on these network environment information.

B. Problem Formulation

We consider a distributed system comprised of a set of N MEC servers, each MEC server $n \in \mathcal{N}$ runs a number of MEC slices, i.e., a set of VMs or containers, $s \in \mathcal{S}$ as in [12]. The mapping between the MEC servers and the MEC slices are defined as $x_{s,n}(t) = 1$ if slice s is deployed in MEC server n, otherwise 0, which means the location of slice s will vary over the time.

For each MEC slice running on the MEC server, we define $C_s(t)$ and $R_s(t)$ as the allocated CPU and RAM resources to the MEC slice s at time slot t. We use $M_s(t) = \{C_s(t), R_s(t), c_s(t), r_s(t)\}$ to represent the MEC slice state of v at time t, where $c_s(t), r_s(t)$ denote the accumulated user demand within slice s at time t. The variation of each item in $M_s(t)$ denotes a NSM pattern, i.e., migration from a given source MEC server to a target MEC server, or scale up/down different resource types such as CPU and RAM.

In this paper, we define a cost model $C(t)$ which consists of two parts: migration cost and resource cost. To evaluate the performance impact that a NSM pattern will induce at time t, the migration cost of a mobility pattern can be evaluated from the mobility type and slice size. Basically, the operation of migration or scaling up/down needs some operation time to finish the whole procedure. We consider the time for the migration operation depends on the slice size, and assume the time for scaling up/down to be static in this paper. The more

resources a slice is associated, the higher migration cost it will induce. Regarding the cost for system resources, the higher utilization of resources, the higher resources cost that network slice provider (NSP) should pay for the infrastructure provider (IP). Moreover, we also reverse the sum of the service latencies T_s of the total users within the slice s, to denote the impact of allocated resources on QoS. The lower latencies, the higher system cost. The total system cost is finally calculated by

$$C(t) = \sum_{s \in S} MC_s + \frac{1}{T_s}(\alpha C_s(t) + \beta R_s(t)), \qquad (1)$$

where MC_s denotes the migration cost of s and $C_s(t)$ and $R_s(t)$ denote the CPU and RAM resource capacity of MEC slice s. α and β are defined to balance the impact of two types of resources on system cost.

Then, considering that different network slice types correspond to the applications with different QoS guarantees, thus each network slice is featured with a unique priority level, i.e., p_s. For simplicity, smaller value means the lower priority. On the one hand, instantiating a network slice will consume a certain amount of resources, which will induce resource costs. On the other hand, network slices with different priorities will produce differential revenue for NSP. Naturally, the network slice which can provide stricter QoS guarantee will make higher revenue, thus it should be prioritized regarding network resource provisioning, i.e., CPU, RAM. The revenue by providing network slices to the slice tenants can be calculated by

$$R(t) = \sum_{s \in S} \frac{1}{T_s} p_s(\alpha c_s(t) + \beta r_s(t)), \qquad (2)$$

where $c_s(t)$ and $r_s(t)$ represent the actual utilized CPU and RAM resources of MEC slice s. α and β are also used as the coefficient to adjust the weights of different resource types in evaluating how they affect the revenue [13]. Note that, due to

the resource constraints, MEC slice cannot always satisfy the user requirements in terms of the CPU or RAM, some slices cannot scale up successfully in response to the increasing user demands.

Therefore, the NSM problem can be formulated as maximizing the system profits over the time concerning the slice mobility as

$$\max \sum_{t \in \mathcal{T}} R(t) - C(t). \tag{3}$$

For the online scheduling purpose where a mobility decision will be generated immediately upon the SRT or SMT trigger, we propose a prediction-based NSM scheme which is near optimal but with less computation complexity.

III. PREDICTION-BASED NETWORK SLICE MOBILITY SCHEME

In this section, we propose a network and user demand prediction-based NSM scheme, which takes the advantage of the network prediction information to eliminate the limitation of greedy strategy. We assume the edge networks to be a time-slotted system and we can obtain the accurate prediction information of networks and users prior to the current time slot t, for example, user resource demands of T time slots ahead of current time slot will be given (predicted) by using the traffic prediction method, e.g., multiple linear regression (MLR) [14]. Therefore, no prediction error will be considered, which actually may have an effect on the system.

As shown in Fig. 2, regarding to the user demand $D_i(t) = (c_i(t), r_i(t))$, the prediction information of user demand $D_i(t+1)$, $D_i(t+2)$, ..., $D_i(t+T)$ is given. Let $U(s)$ denote the set of users which are served by network slice s, the resource demand of network slice s at time slot t should be $D_s(t) = \sum_{i \in U(s)} D_i(t)$. In the proposed scheme, the prediction information of slice resource demand will indicate the future trend of mobility pattern, e.g., scaling up/down or migration. The resource demand trend of multiple network slices located in one MEC server will be taken into account as a key factor to determine the migration target. Thus, we define a variable $R_n(t, T)$ to reflect the traffic demand of MEC server n in the next T time slots as follows:

$$R_n(t, T) = \sum_{s \in S(n)} A_s(t, T), \tag{4}$$

where if $A_s(t, T) = D_s(t+T) - D_s(t) > 0$ means the traffic demand is increasing in the next T time slots and vice versa.

On the one hand, the prediction information of traffic demand within an existing slice will indicate 1) when this slice needs to be scaled up in the future and 2) whether the network resources should be reserved in advance according to its priority level. Besides, the network load prediction could also facilitate the optimized slice mobility scheme which will take advantage of a complementary mobility strategy, e.g., migrating a network slice with increasing resource demands to a MEC server with decreasing network loads. Thus, the network resource competition could be mitigated. Moreover,

the user and network prediction will also promote the priority-based slice mobility scheme. With the prediction information, the network slice with higher priority can always be prioritized by provisioning or reserving adequate network resources to ensure the QoS. As shown in Fig. 2, two motivational examples are presented to illustrate the prediction-based slice mobility scheme. Three types of network slices with high priority (HP), medium priority (MP) and low priority (LP) exist in the networks. In Fig. 2(a), the MP slice needs to be scaled up at time $t+1$ due the increasing resource demands of users, i.e., $(8, 4)$, and the resource demand of LP slice is also increasing in the following time slots. Considering the physical resource capacity constraints, there are not enough network resources, e.g., CPUs, for this scaling up operation. Thus, an intuitive policy might be migrating this MP slice to MEC server 2, which still has adequate network resources. However, if we consider the future resource demand of this MP slice, we will find there is a downward tendency of resource demand in the next time slot, e.g., from $t + 2$. As a result, an intelligent policy should be scaling down the LP slice in MEC server 1 at time $t+1$ since slice migration operation will also induce some cost. In the other case, if the resource demand of an MP slice increases continuously in $t + 1$, $t + 2$..., there will be resource competition in the next time slots with LP slices for a relatively long time period, which will induce much degradation cost for this LP slice. At the same time, the total resource demands of MEC server 2 present a decreasing tendency from $t + 1$ as shown in Fig. 2(b) if we have the prediction information of all the slices in MEC server 2. Under this circumstance, an intelligent policy should be migrating the MP slice towards MEC server 2 to accommodate the increasing user demands, eventually minimizing the total system cost.

We aim to design an intelligent NSM strategy that takes advantage of the complementary user demand profiles provided by the user and network prediction information to determine the NSM policies. The proposed scheme is detailed in the Alg. 1. Given the geographically distributed MEC servers and mobile users, the network slices are firstly initialized based on the user distribution and resource demands. For example, a high priority slice will be deployed within the MEC server which covers most of mobile users requiring high priority services (Line 1). Afterwards, we iteratively determine the appropriate slice mobility as the resource requirements of mobile users change. (Line 2-15). Specifically, we first sort the network slices deployed in the networks in terms of the priority levels and operate on the slices to highest priority to ensure the resource provisioning of critical network slices (Line 3). When the resource demands $D_s(t)$ of slice s increase at time slot t compared with $t - 1$, we check if there are enough resources to scale up the slice in the current MEC server of s. If not, we consider other MEC server that satisfy the resource requirements of s (Line 9-13). In Line 9, the rank values of the neighbor MEC server including the current MEC server of slice s, i.e., $R_n(t, T)$, are calculated to provide an evaluation of the load tendency of MEC server n in the next T time slots. After sorting the MEC server in $N(s)$,

Algorithm 1: Network Prediction-based Network Slice Mobility Scheme

---

Input: Network Topology $\mathcal{G}$, Mobile Users $\mathcal{U}$
Output: Network Slice Configuration

1 **Initialize:** Deploy network slices based on user distribution
2 **for** $t \in \mathcal{T}$ **do**
3      $\mathcal{S} \leftarrow$ Sort the slices in terms of priority in a descending order.
4   **for** $s \in \mathcal{S}$ **do**
5   **if** $D_s(t) > D_s(t-1)$ **then**
6   **if** $L_{I(n)}(t) > D_s(t) - D_s(t-1)$ **then**
7   Scale up slice s.
8   **else**
9   Calculate the rank $R_n(t,T)$ of neighbor MEC server $n \in N(s)$.
10                 $\hat{N}(s) \leftarrow$ the set of neighbors in terms of $R_n(t,T)$ in ascending order.
11   **for** $n' \in \hat{N}(s)$ **do**
12   **if** $Check(s,n') > 0$ **then**
13   Migrate the slice s with $Migrate(s,n')$.
14   **else**
15   Continue
16   **else**
17   Scale down the slice s or no action.

Algorithm 2: $Check(s,n)$

---

if $n \neq I(s)$ **then**
     $MC(s,n) \leftarrow |U(s)| * f_s$.
else
     $MC(s,n) \leftarrow 0$.
$MB(s,n) \leftarrow p_s * (D_s(t) - D_s(t-1))$
if $L_n(t) < (D_s(t) - D_s(t-1))$ **then**
     $MP(s,n) \leftarrow$ Penalty for release the network resources $(D_s(t) - D_s(t-1))$ from the network slice s' with lower priority than s multiplied by a factor $P_{s'}$.
return $MB(s,n) - MC(s,n) - MP(s,n)$.

---

we check if MEC server n' is a valid migration option. As shown in Alg. 2, three indicators are defined to check if the MEC server n is a feasible destination for slice s to migrate. $MC(s,n)$ denotes the migration cost since slice migration will induce the service interruption and degrade the QoS for a while. Hence, the migration cost is defined as the number of users $|U(s)|$ served by slice s multiplied by a factor f_s which indicates the slice size. Migration benefit $MB(s,n)$ is defined to indicate how much revenue the slice s can obtain by scaling up operation in the target MEC server. If the residual

network resources are not enough for slice s to scale up, the corresponding resources will released from an existing slice s' with lower priority than s. And a certain of penalty will be paid for degrading this slice which is $P_{s'} * (D_s(t) - D_s(t-1))$, i.e., $MP(s,n)$. To evaluate the necessity to migrate slice s, Alg. 2 will return the value of $MB(s,n) - MC(s,n) - MP(s,n)$. If this value is positive, the MEC server n is a valid option for migration (Line 13), otherwise, the algorithm will iterate to the next candidate MEC server.

IV. PERFORMANCE EVALUATION

In this section, we demonstrate the performance evaluation of our proposed scheme. We first present the simulation settings used in the evaluation. Then, we compare our proposed scheme with the existing benchmark schemes and evaluate their performance in different cases.

A. Simulation Setup

We initialize a scenario consists of several BSs, each of which serves 20 mobile users and deploys an ES with 8 CPU cores and 32 GB RAM. The topology of the BSs is generated by the *internet graph generator* of **Networkx** tool [15]. We set the latency from each BS to its served mobile users as $10\ ms$, and the latency between two connected BSs as $50\ ms$. We set the CPU cores that each user request each time slot randomly from 2 to 4, and RAM from 2 to 12 GB. The priorities of requests are set randomly from 1 to 5. Moreover, to observe the simulation performances in a long term, we collect the simulation results of 50 time slots and iterate the process of determining the strategies of NSM for 100 times to obtain more stable performances. We evaluate the performance of the proposed scheme in terms of system cost, revenue, and profit. To demonstrate the effect of predicting the user demand in NSM, we compare our proposed scheme with two schemes as: **1) Reset scheme:** the slices in t scale up/down based on user demand, and the slices cannot migrate to other MEC servers even the resources are exhausted; **2) Greedy scheme:** the slices scale up/down firstly, if the resources are exhausted, the slices greedily migrate to the nearest MEC servers of its users. Different from these two schemes, all the slices can determine its nearest MEC server of its users in our prediction-based scheme, and we try to greedily migrate the slices to its nearest MEC server according to the priorities of slices.

B. Performance Results

Based on these settings, we evaluate the performances in different numbers of MEC server, and different combinations of α, β which control the proportion of CPU and RAM resources in calculating the performance metrics.

We investigate the system cost, revenue, and profit under different combinations of α, β versus different N in Fig. 3. Note that the simulation results are averaged over 100 iterations. From Fig. 3(a), we can observe that the system cost of different schemes and combinations gradually increase with N, since the increasing MEC server serve much more users, who request more resources. Moreover, in Fig. 3(a), each scheme with lower α results in higher cost, which reflects
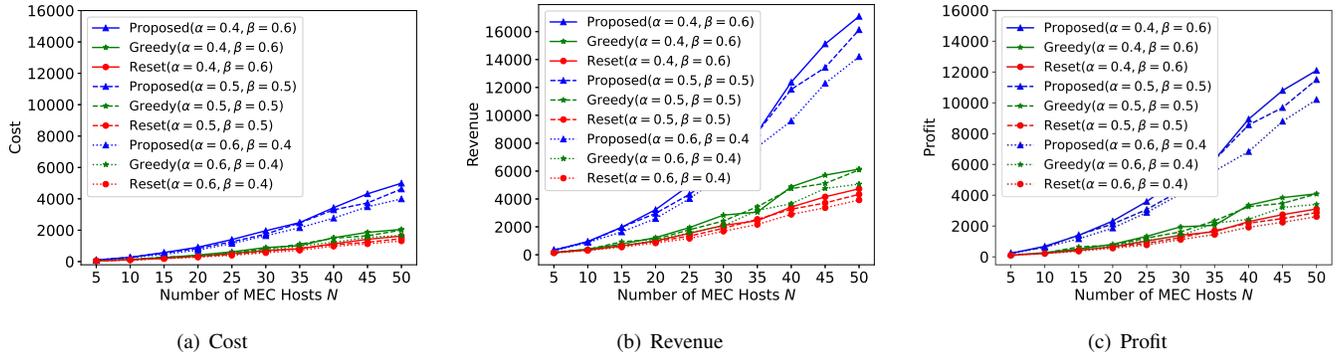
Fig. 3. The cost, revenue, and profit of the considered system versus N.

that the requested RAM resources cost more than the requested CPU cores. The proposed scheme with all (α, β) combinations has much higher cost compared to the reset scheme and greedy scheme, which means that more resources are utilized.

In Fig. 3(b), we compare the revenue of different schemes with different parameter combinations. We can observe that the each scheme with lower α haves higher revenue, and the proposed scheme with all (α, β) combinations has much higher revenue compared to the reset scheme and greedy scheme, since the proposed scheme's strategy considers to migrate all the slices to their nearest MEC server and thus achieves much lower system latency. To further evaluate the performances of different schemes, we investigate profit of these schemes, as shown in Fig. 3(c). We can observe that the proposed scheme outperforms other schemes in all parameter combinations. As a result, the proposed scheme averagely increases the system profit by up to 30 times and 19 times compared to reset scheme and greedy scheme, respectively.

V. Conclusion

In this paper, we have investigated the user and network prediction-based NSM problem in 6G MEC environments. We have firstly formulated the mathematical model of this problem with the objective of maximizing the system profits and then proposed a novel network analytic-enabled slice mobility algorithm by exploiting user demand and network load prediction information to make the mobility decision. The proposed scheme will leverage the complementary user demand profiles or the complementary network load in different MEC servers to avoid the resource competition from a long term perspective. To this end, the network slices with high priorities can always be provisioned with enough resources to ensure the QoS associated services. The performance evaluation results have shown that the proposed prediction-based scheme obtains higher overall profits regarding mobility and network resources compared with two benchmark strategies.

Acknowledgment

References

[1] T. Taleb, B. Mada, M.-I. Corici, A. Nakao, and H. Flinck, "Permit: Network slicing for personalized 5g mobile telecommunications," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 88–93, May 2017.

[2] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 2429–2453, Mar. 2018.

[3] M. Shokrnezhad and T. Taleb, "Near-optimal cloud-network integrated resource allocation for latency-sensitive b5g," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4498–4503.

[4] H. Yu, T. Taleb, J. Zhang, and H. Wang, "Deterministic latency bounded network slice deployment in ip-over-wdm based metro-aggregation networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 2, pp. 596–607, 2022.

[5] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Netw.*, vol. 34, no. 1, pp. 84–93, Jan. 2020.

[6] R. A. Addad, D. L. C. Dutra, T. Taleb, and H. Flinck, "Toward using reinforcement learning for trigger selection in network slice mobility," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2241–2253, May 2021.

[7] F. Farahnakian, P. Liljeberg, and J. Plosila, "Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers," in *Proc. IEEE Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Sept. 2013, pp. 357–364.

[8] R. Shaw, E. Howley, and E. Barrett, "A predictive anti-correlated virtual machine placement algorithm for green cloud computing," in *Proc. IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, Dec. 2018, pp. 267–276.

[9] Y. Chen, Y. Sun, B. Yang, and T. Taleb, "Joint caching and computing service placement for edge-enabled iot based on deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19501–19514, 2022.

[10] T. Taleb, R. L. Aguiar, I. Grida Ben Yahia, B. Chatras, G. Christensen, U. Chunduri, A. Clemm, X. Costa, L. Dong, J. Elmirghani *et al.*, "White paper on 6g networking," 2020.

[11] G. ETSI, "Mobile edge computing (mec); deployment of mobile edge computing in an nfv environment," *ETSI ISG*, Feb. 2018.

[12] X. Wang, R. Li, C. Wang, X. Li, T. Taleb, and V. C. M. Leung, "Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 154–169, 2021.

[13] H. Yu, T. Taleb, and J. Zhang, "Deterministic latency/jitter-aware service function chaining over beyond 5g edge fabric," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2148–2162, 2022.

[14] R. Vinayakumar, K. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *Proc. IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sept. 2017, pp. 2353–2358.

[15] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proc. Python in Science Conference (SciPy)*, Jan. 2008, pp. 11 – 15.