A Resoure Allocation Framework for Network Slicing with Multi-service Coexistence

Siqi Huang†, Yao Sun‡, Daquan Feng†, Wei Jiang†, Zongxiang Liu†
† The Shenzhen Key Laboratory of Digital Creative Technology, Shenzhen University, Shenzhen 518060, China
‡ James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom,
Email: Yao.Sun@glasgow.ac.uk

***Abstract*—Network slicing has been widely recognized as the architectural technology for 5G and beyond wireless network systems to provide tailored service for diverse applications by flexibly splitting and allocating various heterogeneous resources. However, it is still challenging to meet the strict delay requirements of a large number of delay-sensitive applications under traditional slicing architectures. One potential way to tackle this issue is to build network slicing upon *Mobile Edge Computing* (MEC) systems, where both communication and computing resources are integrated for providing customized service. As such, in this paper, we propose a framework, to jointly optimize communication and computing resources under the scenario of multi-service coexistence, with the objective to minimize the system cost while meeting the diverse QoS requirements. To make the original optimization problem more tractable, we decompose it into two convex sub-problems first. Then we obtain the optimal solutions of the two sub-problems respectively, and finally derive the optimal communication and computing resource allocation scheme based on the optimal solutions of these two sub-problems. Simulation results show that our proposed scheme significantly saves the system cost under various scenarios compared with other benchmarks.**

I. Introduction

In recent years, network slicing technology has been proposed to cope with the diverse application scenarios of future mobile networks by dividing the common physical infrastructure into multiple logically separate networks to provide tailored service for different demands [1]–[3]. With the emergence of a large number of delay-sensitive applications, such as industrial automation and control, *vehicle-to-everything* (V2X), *virtual reality/augmented reality* (VR/AR), etc [4]–[6], it is expected that 5G and beyond wireless network systems can support these applications with lower latency. However, it is still challenging under traditional slicing architectures.

Mobile edge computing (MEC), which migrates computing from centralized cloud computing to the edge of the network [7], has been envisaged as a promising paradigm to provide users with closer computing resources and better experience. With MEC, it can be more efficient to allocate computing resources for delay-sensitive applications to meet the QoS requirements.

To provide extremely low delay for future networks, building network slicing upon MEC can be expected as a promising way to jointly optimize communication and computing resources to fulfill the diverse requirements. Some relevant research work has been done mainly from the perspective of improving QoS of mobile users. Xiang *et al.* [8] proposed a mathematical model to effectively jointly allocate mobile network and edge computing resources to solve the resource allocation problem of multiple edge networks. Wang *et al.* [9] analyzed the long-term performance of edge network slicing and developed a resource orchestration mechanism to minimize network costs under the guarantee of quality of service (QoS). In [10], the operator's average revenue is optimized by jointly considering slice request admission in the long-term and resource allocation in the short-term. Generally, these existing works have investigated workload scheduling, resource orchestration, power allocation, and slice admission from the perspective of mobile users or operators. However, they have not considered the cost incurred by deploying network slicing under MEC architecture, which is a crucial prerequisite for operators to obtain more revenue.

Motivated by the above, in this paper, we propose a framework, to jointly optimize communication and computing resources with multi-service coexistence by building network slicing over MEC systems. The design objective is to minimize the system cost (i.e., bandwidth allocation cost, MEC server acquisition cost, and cloud computing capacity rental cost) while guaranteeing the QoS requirements of two typical services, i.e., *enhanced Mobile BroadBand* (eMBB) and *ultra-reliable low-latency communications* (URLLC) [11]. The main contributions of our work are as follows.

- In our framework, we dynamically allocate communication and computing resources to meet the diverse QoS requirements for achieving the coexistence of multi-service.
- We formulate the optimization problem of resource allocation with the objective of minimizing the system cost. Taking the slice type of eMBB and URLLC as an example, we derive the optimal communication and computing resource allocation scheme based on the optimal solutions of these two sub-problems.
- To achieve the isolation of eMBB slices and URLLC slices, as well as the stringent delay requirements of URLLC slices, we give priority to allocating sufficient computing resources for URLLC requests, and then allocate appropriate computing resources for eMBB requests.

II. SYSTEM MODEL

In our framework, the communication resources in RAN and the computing resources in MEC are sliced to meet the QoS requirements. In this work, we consider slicing requests from eMBB and URLLC, two major application scenarios of 5G and beyond wireless network systems, and the bandwidth and computing resources required by URLLC slices and eMBB slices are denoted by b^u, s^u, b^e and s^e, and the computing resources leased from remote cloud servers is denoted by ${s_c}^t$.

To cater the dynamic of the slice requests in practice, the timeslotted model is considered here, where time is divided into *long time slots* (LTSs) and *short time slots* (STSs) [12]. We denote LTS as L, and each LTS contains n STSs, denoted as $L = (t_1, t_2, ..., t_n)$. At the beginning of an LTS, the communication and computing resources (i.e., b^u, s^u, b^e and s^e) for two types of slices will be allocated. At the beginning of an STS, a decision will be made to determine the amount of requests offloaded to the remote cloud server. Here, we introduce a continuous variable $\alpha_i(t)\ \epsilon\ (0, 1)$ to denote the amount of requests processed on the edge server.

A. eMBB Slice

In this work, we assume that the users under eMBB slices share all bandwidth resources, and denote the set of UEs under eMBB slices as $I^E = \{1, 2, \cdots, I^e\}$. The corresponding *signal-to-noise ratio* (SNR) at UE i over the l-th LTS is

$$SNR_i^e(l) = \frac{P_i^e(l) \cdot h_i^e(l)}{\sigma_i^2(l)}, \tag{1}$$

where $P_i^e(l)$ is the transmission power from UE i under eMBB slices to the base station over the l-th LTS, $h_i^e(l)$ is the channel gain of UE i under eMBB slices over the l-th LTS, and $\sigma_i(l) \sim N(0, \sigma^2)$ is the Gaussian white noise in the channel of UE i under eMBB slices over the l-th LTS. Let the $r^e(l)$ be the achievable rate at LTS l, to ensure the successful data reception of all user [13], we have

$$r^e(l) \leqslant \min_{i \epsilon I^e} \{log(1 + SNR_i^e(l))\}. \tag{2}$$

According to the Shannon-Hartley formula [14], the transmission rate at LTS l is

$$R^e(l) = b^e \cdot r^e(l). \tag{3}$$

In this paper, we assume that bandwidth allocation and achievable rate are independent. In order to meet the transmission rate requirement of eMBB slices, the achievable rate of UE i at LTS l should satisfy

$$R^e(l) \geq R_s, \tag{4}$$

where R_s is the throughput requirement of eMBB slices. Then, the transmission delay of UE i at slot t is

$$D_{i,e}^T(t) = \frac{F_i^e(t)}{R^e(l)}, \forall\, t\, \epsilon\, l, \tag{5}$$

where $F_i^e(t)$ is the size of the data packet transmitted by UE i under eMBB slices in time slot t.

When the computational requests are offloaded to the MEC server, the $M/M/1$ queuing model is adopted to analyze the processing delay [15]. We use s_i^e to represent the computing resources allocated to user i, then the processing delay of UE i at slot t is

$$\begin{aligned} D_{i,e}^P(t) = &\frac{1}{s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)} \\ &+ \frac{1}{{s_c}^t - (1 - \alpha_i(t)) \cdot \lambda_i^e(t)}, \forall\, t\, \epsilon\, l, \end{aligned} \tag{6}$$

where $\lambda_i^e(t)$ is the task arrival of UE i under eMBB slices in time slot t. Thus the total delay of UE i under eMBB slices at slot t is

$$D_{i,e}(t) = D_{i,e}^T(t) + D_{i,e}^P(t) + d, \tag{7}$$

where d is the back-haul delay between the edge server and the remote cloud server. To meet the delay requirement of eMBB slices, which denoted by D_1, the delay constraint of UE i under eMBB slices at slot t is

$$D_{i,e}(t) \leqslant D_1. \tag{8}$$

B. URLLC Slice

In URLLC, the Shannon-Hartley formula is unavailable in the finite block-length channel coding regime. Instead, the achievable rate is derived in [16], which is

$$r_i^u(l) = \log(1 + SNR_i^u(l)) - \sqrt{\frac{C_i^u(l)}{n_i^u}} \cdot Q^{-1}(\varepsilon) \log e, \tag{9}$$

where $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function, ε is the transmission error probability, n_i^u is the length of block code and $C_i^u(l)$ is the $channel\ dispersion$[2] of UE i under URLLC slices at LTS l, given by

$$C_i^u(l) = 1 - \frac{1}{(1 + SNR_i^u(l))^2}. \tag{10}$$

Similarly, the transmission rate of UE i under URLLC slices at LTS l is

$$R_i^u(l) = b_i^u(l) \cdot r_i^u(l). \tag{11}$$

And the transmission delay of UE i under URLLC slices at slot t is

$$D_{i,u}^T(t) = \frac{F_i^u(t)}{R_i^u(l)}, \forall\, t\, \epsilon\, l. \tag{12}$$

In this paper, all URLLC requests are processed on the edge server, thus the processing delay is

$$D_{i,u}^P(t) = \frac{1}{s_i^u - \lambda_i^u(t)}, \forall\, t\, \epsilon\, l. \tag{13}$$

The total delay of UE i under URLLC slices at slot t is

$$D_{i,u}(t) = D_{i,u}^T(t) + D_{i,u}^P(t). \tag{14}$$

To meet the delay requirement of URLLC slices, which denoted by D_2, the delay constraint of UE i under URLLC slices at slot t is

$$D_{i,u}(t) \leqslant D_2. \tag{15}$$

C. System Cost

The system cost consists of bandwidth consumption, deploying edge servers and renting remote cloud instances. From [17], we know the cost of deploying edge servers increases with the computing resources, which can be expressed as

$$C(s^u) = c_s \cdot (s^u)^\theta, \tag{16}$$

where c_s and θ are constants, representing the linear and exponential relationship between the cost and resources, respectively, and $c_s > 0$, $\theta \geqslant 1$. And the cost of renting on-demand cloud instances can be expressed as

$$C({s_c}^t) = c_s \cdot {s_c}^t. \tag{17}$$

Similarly, the cost of allocating bandwidth can be expressed as

$$C(b^u) = c_b \cdot (b^u)^\theta, \tag{18}$$

where c_b is constant, representing the linear relationship between cost and resources, and $c_b > 0$. Thus the system cost is given by

$$C(b,s) = V \cdot [C(b^u) + C(b^e)] + C(s^u) + C(s^e) + \sum_{t=1}^{n} C({s_c}^t), \tag{19}$$

where V is a factor used to strike the trade-off between the cost of allocating bandwidth and the cost of allocating computing resources.

D. Problem Formulation

In this work, we focus on how to allocate bandwidth and computation resources to satisfy the QoS requirements of eMBB slices and URLLC slices at the lowest cost. The optimization problem is formulated as

$$\begin{aligned} \min_{\alpha_i(t), b^u, b^e, s^u, s^e, {s_c}^t} \quad & C(b,s) \\ s.t. \quad & \alpha_i(t) \epsilon (0,1), \forall\, t \,\epsilon\, l, \\ & (2), (4), (8), (15). \end{aligned} \tag{20}$$

III. Problem Solution

In this work, we assume that the two types of slices use different frequency of wireless bandwidth. For computing resources, we have taken priority to URLLC slices. Based on above, the isolation between URLLC slices and eMBB slices can be achieved.

A. Analysis for URLLC Slice

Since URLLC slices is delay-sensitive, and renting cloud resources will introduce back-haul delay, it is assumed that all URLLC requests are processed on the edge server, with higher priority compared with eMBB requests, then we have the optimization problem for URLLC slices as follows.

$$\begin{aligned} \min_{b^u, s^u} \quad & V \cdot C(b^u) + C(s^u) \\ s.t. \quad & \sum_{i \epsilon I^U} b_i^u \leqslant b^u, \\ & \sum_{i \epsilon I^U} s_i^u \leqslant s^u, \\ & D_{i,u}(t) \leqslant D_2. \end{aligned} \tag{21}$$

To solve this problem, we can consider the scenario of single-user at first. From (12), (13), (14) and (15), we can know that the computing resources required to meet the delay requirements of UE i under URLLC slices are at least

$$s_i^u \geqslant \frac{1}{D_2 - \frac{F_i}{b_i^u \cdot r_i^u(l)}} + \lambda_i(l). \tag{22}$$

To minimize the cost, it holds that

$$s_i^u = \frac{1}{D_2 - \frac{F_i}{b_i^u \cdot r_i^u(l)}} + \lambda_i(l). \tag{23}$$

From the above formula, we can obtain that s_i^u is a function of b_i^u. In addition, the transmission delay should not exceed the delay requirement,

$$\frac{F_i}{b_i^u \cdot r_i^u(l)} < D_2. \tag{24}$$

Further the bandwidth need to meet the delay requirements of UE i under URLLC slices,

$$\frac{F_i}{D_2 \cdot r_i^u(l)} < b_i^u. \tag{25}$$

According to (16), (18) and (23), the optimization problem for each user is transformed into searching the minimum value of the function of b_i^u within the constraints, which is

$$\begin{aligned} f(b_i^u) &= V \cdot c_b \cdot (b_i^u)^\theta + c_s \cdot \left(\frac{1}{D_2 - \frac{F_i}{b_i^u \cdot r_i^u(l)}} + \lambda_i(l)\right)^\theta \\ s.t. \quad & \frac{F_i}{D_2 \cdot r_i^u(l)} < b_i^u \leq \bar{b}_i^u. \end{aligned} \tag{26}$$

Proposition 1. $f(b_i^u)$ is a convex function over b_i^u within the constraints.

Proof: See Appendix A.

Since the cost is a convex function over b_i^u, we can search the b_i^u that minimize the cost by binary search, and the time complexity is $O(\log N)$. Finally we obtain the optimal solution of the optimization problem for URLLC slices by adding up the optimal cost of each user.

B. Analysis for eMBB Slice

For eMBB slices, our objective is to allocate bandwidth of RAN, computing resources provided by edge hosts and remote cloud instances at lowest cost. At first, in RAN, according to (2), (3) and (4), when the cost is at its lowest, we have

$$R^e(l) = R_s = b^e \cdot \min_{i \epsilon I^E} r_i^e(l). \tag{27}$$

Further the optimal b^e can be solved by above formula. Then the optimization problem for eMBB slices can be simplified as

$$\begin{aligned} \min_{\alpha_i(t), s^e, {s_c}^t} \quad & V \cdot C(s^e) + \sum_{t=1}^{n} C({s_c}^t) \\ s.t. \quad & \alpha_i(t) \epsilon (0,1), \forall\, t \,\epsilon\, l, \\ & (2), (4), (8). \end{aligned} \tag{28}$$

The problem can be solved as follows:

Step 1. Determine s_i^e.

According to (6), to ensure the stability of queues, we have

$$s_i^e > \max_{t\epsilon l} \alpha_i(t) \cdot \lambda_i^e(t). \tag{29}$$

Since $\alpha_i(t) \ \epsilon \ (0,1)$, it can be deduced that the computing resources allocated to UE i under eMBB slices are at least

$$s_i^e \geq \max_{t\epsilon l} \lambda_i^e(t). \tag{30}$$

Here, we assume that $\max_{t\epsilon l} \lambda_i^e(t)$ can be predicted according to the maximum value of $\lambda_i^e(t)$ in previous LTSs. As the cost decreases with the reduction of computing resources, to minimize the cost, we have

$$s_i^e = \max_{t\epsilon l} \lambda_i^e(t). \tag{31}$$

Then the computing resources of edge hosts s_i^e remain unchanged through all time slots at current LTS.

Step 2. Make the offloading decision by determining ${s_c}^t$.

There are two situations upon the computing resources of edges hosts s_i^e. The first one is that s_i^e are sufficient, and all requests are processed on the edge server, while guaranteeing the delay requirement, it can be expressed as

$$\frac{1}{s_i^e - \lambda_i^e(t)} \leq D_1 - d - D_{i,e}^T(t). \tag{32}$$

In this situation, ${s_c}^t = 0$.

The second one is that s_i^e are not sufficient, thus some requests should be processed on the remote cloud server. In this situation, an offloading decision should be made to minimize ${s_c}^t$, i.e., search the optimal $\alpha_i(t)$ that minimizes ${s_c}^t$. According to (6), (7), (8),we have

$$\begin{aligned} \frac{1}{s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)} + \frac{1}{{s_c}^t - (1-\alpha_i(t)) \cdot \lambda_i^e(t)} \\ \leq D_1 - D_{i,e}^T(t) - d. \end{aligned} \tag{33}$$

When obtain the optimal ${s_c}^t$, it holds that

$$\begin{aligned} \frac{1}{s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)} + \frac{1}{{s_c}^t - (1-\alpha_i(t)) \cdot \lambda_i^e(t)} \\ = D_1 - D_{i,e}^T(t) - d. \end{aligned} \tag{34}$$

From this we can deduce that

$$\begin{aligned} {s_c}^t = \frac{\lambda_i^e(t) - s_i^e}{1 - D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t))} \\ - \frac{D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)) \cdot (\lambda_i^e(t) - \alpha_i(t) \cdot \lambda_i^e(t))}{1 - D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t))}. \end{aligned} \tag{35}$$

Proposition 2. ${s_c}^t$ is a convex function over $\alpha_i(t)$, When $\alpha_i(t) \epsilon [0, \frac{D_{i,e}^P(t) \cdot s_i^e - 1}{D_{i,e}^P(t) \cdot \lambda_i^e(t)}]$.

$Proof$: See Appendix B.

Since ${s_c}^t$ is a convex function over $\alpha_i(t)$, we can search the $\alpha_i(t)$ that minimize ${s_c}^t$ by binary search, and the time complexity is $O(\log N)$. Finally we can obtain the optimal cost of each user according to (16), (17), (18).

Step 3. Obtain the optimal solution.

Knowing the optimal bandwidth and the optimal computing resources allocated to each user, we can obtain the optimal solution of the optimization problem for eMBB slices by adding up the optimal cost of each user.

Table I Simulation Parameters

Parameter	Value	Parameter	Value
N_o	-174 dBm/Hz	B	20 MHz
n_i	168 $\sim$ 336 bytes	p	20 w
F_i^e	420 kB	F_i^u	500 bytes
ε	10^{-5}	D_1	400 ms
l	60 s	t	1 s
V	0.1	θ	1

IV. SIMULATION RESULTS

In this section, we present simulation results to evaluate the performance of the proposed scheme.

A. Simulation Parameters

We consider a scenario where a base station with multiple edge servers is connected to a remote cloud via the Internet, and the back-haul delay d between the edge server and the remote cloud server is 50 milliseconds. In the simulation, flat fading channel is adopted as the wireless transmission channel model, and the path loss from the device to the base station is $\rho = 128.1 + 37.6 \log(d)\ dB$, where d is the distance from the device to the base station in km [13]. The shadow fading is $\delta \sim N(0, 8)\ dB$. As for the cost, the average cost for mobile traffic is $\$2.68/GB$ [18]. An edge server with $9.6\ GHz$ costs \$3000, which can be used for about 3 years. The cost of renting a cloud instance ($AmazonEC2$) is $\$0.0208/GHz$ per hour [17]. Other simulation parameters are shown in Table 1.

B. Performance of the Proposed scheme

In this simulation, we use Sine Traffic Model as that in [13], which is widely used to describe traffic distribution in cellular network. The following three benchmarks are used for comparison.

Benchmark 1, both URLLC requests and eMBB requests are partly processed on the edge server, and the others are processed by renting remote cloud instances.

Benchmark 2, URLLC requests are all processed on the edge server, while eMBB requests are all processed by renting remote cloud instances.

Benchmark 3, both URLLC requests and eMBB requests are processed by renting remote cloud instances.

Fig. 1 shows system cost under different number of UEs for the four schemes. From this figure, we can see that the cost increases with number of UEs for all the four schemes, and the proposed scheme can always achieve the minimum system cost compared with the benchmark. This is because we jointly optimize computing and communication resources to minimize the system cost, and give priority to URLLC slices requests when allocating computing resources, resulting in a significant cost reduction.

In the next experiment, we compare the system cost under different delay requirements of URLLC slices (i.e., D_2), while the throughput requirement of the eMBB slices remains unchanged at each LTS, i.e., $R_s = 6\ Mb/s$. From Fig.2, we can see that the system cost decreases with D_2 and remains almost unchanged when $D_2 > 0.2\ s$. Since a tighter delay requirement need allocate more resources to meet, the system

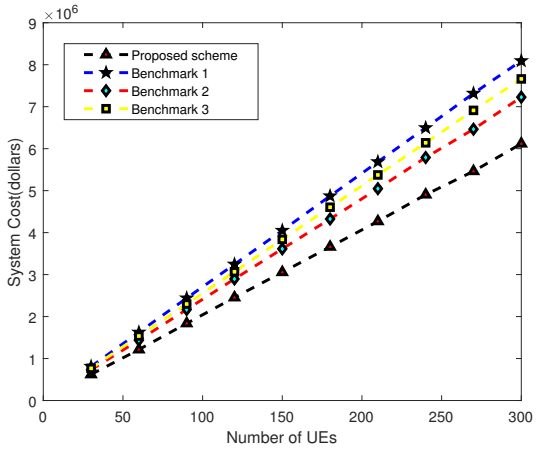

Fig. 1: System cost over number of UEs

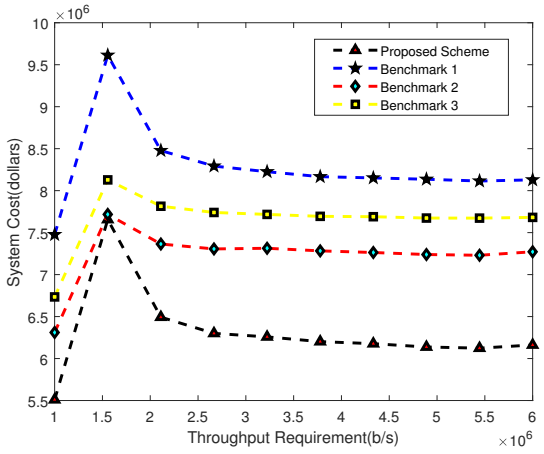

Fig. 3: Change R_s for eMBB Slice

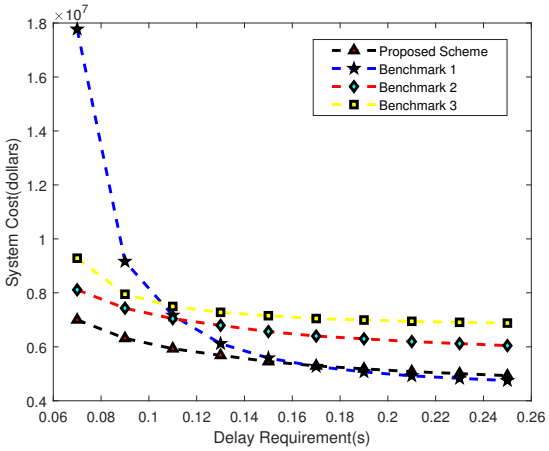

Fig. 2: Change D_2 for URLLC Slice

cost decreases with the delay requirements. When $D_2 > 0.2\ s$, the delay requirement becomes looser, and the main factor affecting the cost becomes number of UEs rather than the QoS requirements, so the system cost remains almost unchanged. Comparing with the benchmarks, we can observe that the proposed scheme still maintains a low system cost under different delay requirements, only when the delay requirement becomes looser, the system cost of benchmark 1 and the proposed scheme are about the same. Owing to the high price of remote cloud instances, all URLLC slices requests are processed on MEC servers in our proposed scheme, so that it is no need to rent numerous cloud instances to meet the stringent delay requirement for URLLC slices requests, which greatly reduces costs.

In Fig. 3, we show the system cost under different throughput requirements of eMBB slices (i.e., R_s), while the delay requirement of the URLLC slices remains unchanged at each LTS, i.e., $D_2 = 0.1\ s$. As shown in Fig. 3, the system cost first rise sharply with the increase of R_s, and reach a peak when R_s approximately equal to $1.6\ Mb/s$, then drop rapidly, and finally remain almost unchanged when $R_s > 3\ Mb/s$. The reason for this is that when the throughput requirements gradually become loose, the cost will first increase with the allocated bandwidth, and then decrease sharply with the allocated computing resources. When balanced, the system cost remains almost unchanged. Comparing with the benchmark, it can be seen that the proposed scheme still maintains the lowest system cost under different throughput requirements, for the reason that the communication and computing resources can be allocated more flexible and efficient according to the QoS requirements in our proposed scheme.

V. Conclusion

In this paper, we propose a framework, to jointly optimize communication and computing resources under the scenario of multi-service coexistence. Specifically, we investigate how to build network slice over MEC architecture, and formulate this as an optimization problem to minimize system cost while guaranteeing the QoS requirements of different service. We take the slice type of eMBB and URLLC as an example, and derive the optimal solution of communication and computing allocation between the two slices. Simulation results demonstrate that our proposed scheme significantly saves the system cost under various scenarios compared with other benchmarks.

Appendix

A. Proof of Proposition 1

Based on (26), the second derivative of $f(b_i^u)$ over b_i^u can be given by

$$\begin{aligned} f(b_i^u)'' &= V \cdot \theta \cdot (\theta - 1) c_b \cdot (b_i^u)^{\theta - 2} \\ &+ \theta \cdot (\theta - 1) \cdot c_s \cdot \Big(\frac{1}{D_2 - \frac{F_i}{b_i^u \cdot r_i^u(l)}} + \lambda_i(l)\Big)^{\theta - 2} \\ &\cdot \Big(\frac{-F_i \cdot r_i^u(l)}{(D_2 \cdot b_i^u \cdot r_i^u(l) - F_i)^2}\Big)^2 \\ &+ \theta \cdot c_s \cdot \Big(\frac{1}{D_2 - \frac{F_i}{b_i^u \cdot r_i^u(l)}} + \lambda_i(l)\Big)^{\theta - 1} \\ &\cdot \Big(\frac{2 \cdot D_2 \cdot F_i \cdot r_i^u(l)^2 (D_2 \cdot b_i^u \cdot r_i^u(l) - F_i)}{(D_2 \cdot b_i^u \cdot r_i^u(l) - F_i)^4}\Big). \end{aligned} \tag{36}$$

From (24), we have

$$D_2 \cdot b_i^u \cdot r_i^u(l) > F_i. \tag{37}$$

Thus

$$f(b_i^u)'' > 0. \tag{38}$$

This completes the proof of Proposition 1.

B. Proof of Proposition 2

According to (6), we have

$$\frac{1}{s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)} < D_{i,e}^P(t). \tag{39}$$

When

$$\frac{1}{s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)} = D_{i,e}^P(t), \tag{40}$$

we can obtain the upper bound of $\alpha_i(t)$, which is $\frac{D_{i,e}^P(t) \cdot s_i^e - 1}{D_{i,e}^P(t) \cdot \lambda_i^e(t)}$. Let

$$f(\alpha_i(t)) = \frac{\gamma(\alpha_i(t))}{\kappa(\alpha_i(t))}, \tag{41}$$

where $\gamma(\alpha_i(t))$ and $\kappa(\alpha_i(t))$ are given by

$$\begin{aligned}\gamma(\alpha_i(t)) = \lambda_i^e(t) - s_i^e - D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)) \\ \cdot(\lambda_i^e(t) - \alpha_i(t) \cdot \lambda_i^e(t)),\end{aligned} \tag{42}$$

$$\kappa(\alpha_i(t)) = 1 - D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)). \tag{43}$$

Then the first derivative of $\gamma(\alpha_i(t))$ and $\kappa(\alpha_i(t))$ are

$$\gamma(\alpha_i(t))' = -D_{i,e}^P(t) \cdot ((2\alpha_i(t) - 1) \cdot \lambda_i^e(t)^2 - s_i^e \cdot \lambda_i^e(t)), \tag{44}$$

$$\kappa(\alpha_i(t))' = D_{i,e}^P(t) \cdot \lambda_i^e(t). \tag{45}$$

The second derivative of $\gamma(\alpha_i(t))$ and $\kappa(\alpha_i(t))$ are

$$\gamma(\alpha_i(t))'' = -2D_{i,e}^P(t) \cdot \lambda_i^e(t)^2, \tag{46}$$

$$\kappa(\alpha_i(t))'' = 0. \tag{47}$$

Thus the second derivative of $f(\alpha_i(t))$ is given by

$$\begin{aligned}f(\alpha_i(t))'' &= \frac{\kappa(\alpha_i(t))^3 \cdot \gamma(\alpha_i(t))''}{\kappa(\alpha_i(t))^4} \\ &- \frac{2\kappa(\alpha_i(t)) \cdot \kappa(\alpha_i(t))' \cdot \gamma(\alpha_i(t))' \cdot \kappa(\alpha_i(t))}{\kappa(\alpha_i(t))^4} \\ &+ \frac{2\kappa(\alpha_i(t)) \cdot \kappa(\alpha_i(t))' \gamma(\alpha_i(t)) \cdot \kappa(\alpha_i(t))'}{\kappa(\alpha_i(t))^4} \\ &= \frac{-2D_{i,e}^P(t) \cdot \lambda_i^e(t) \cdot (1 - D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)))}{(1 - D_{i,e}^P(t)(s_i^e - \alpha_i(t) \cdot \lambda_i^e(t)))^4}.\end{aligned} \tag{48}$$

Since $1 - D_{i,e}^P(t)(s_i^e - \alpha_i(t)) < 0$, there exists

$$f(\alpha_i(t))'' > 0. \tag{49}$$

This completes the proof of Proposition 2.

REFERENCES

[1] Y. Sun, S. Qin, G. Feng, L. Zhang, and M. Imran, "Service provisioning framework for ran slicing: User admissibility, slice association and bandwidth allocation," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[2] Y. Sun, W. Jiang, G. Feng, P. V. Klaine, L. Zhang, M. A. Imran, and Y. C. Liang, "Efficient handover mechanism for radio access network slicing by exploiting distributed learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2620–2633, 2020.

[3] Y. J. Liu, G. Feng, Y. Sun, S. Qin, and Y. C. Liang, "Device association for ran slicing based on hybrid federated deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 731–15 745, 2020.

[4] C. Zheng, D. Feng, S. Zhang, X. Xia, G. Qian, and G. Y. Li, "Energy efficient v2x-enabled communications in cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 554–564, 2019.

[5] D. Feng, C. She, K. Ying, L. Lai, Z. Hou, T. Q. S. Quek, Y. Li, and B. Vucetic, "Toward ultrareliable low-latency communications: Typical scenarios, possible solutions, and open issues," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 94–102, 2019.

[6] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A tutorial on ultra-reliable and low-latency communications in 6g: Integrating domain knowledge into deep learning," 2021.

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[8] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5g networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[9] Y. Wang, Y. Gu, and X. Tao, "Edge network slicing with statistical qos provisioning," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1464–1467, 2019.

[10] J. Feng, Q. Pei, F. R. Yu, X. Chu, J. Du, and L. Zhu, "Dynamic network slicing and resource allocation in mobile edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7863–7878, 2020.

[11] D. Feng, L. Lai, J. Luo, Y. Zhong, C. Zheng, and K. Ying, "Ultra-reliable and low-latency communications: applications, opportunities and challenges," *SCIENCE CHINA Information Sciences*, vol. 64, no. 2, pp. 120 301–, 2021.

[12] H. Zhang and V. W. S. Wong, "A two-timescale approach for network slicing in c-ran," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6656–6669, 2020.

[13] J. Tang, B. Shim, and T. Q. S. Quek, "Service multiplexing and revenue maximization in sliced c-ran incorporated with urllc and multicast embb," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 881–895, 2019.

[14] W. Chen and L. Han, "Time-efficient task caching strategy for multi-server mobile edge cloud computing," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019, pp. 1429–1436.

[15] Y. Niu, B. Luo, F. Liu, J. Liu, and B. Li, "When hybrid cloud meets flash crowd: Towards cost-effective service provisioning," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 1044–1052.

[16] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[17] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. S. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[18] "Mobile traffic cost," http://finance.eastmoney.com/news/1355,20180306-840167074.html, 2018.