A Network Function Parallelism-enabled MEC Framework for Supporting Low-Latency Services

Mengjie Liu*, Gang Feng*, *Senior Member, IEEE*, Yao Sun†, *Member, IEEE*, Nan Chen‡, *Member, IEEE*, Wei (andrew) Tan§

*National Key Lab. on Communications, University of Electronic Science and Technology of China.
†James Watt School of Engineering, University of Glasgow, United Kingdom.
‡ Dept. of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA.
§ Central Research Institute, Huawei Technologies, Co. Ltd Shanghai, China.
Email: fenggang@userstc.edu.cn

***Abstract*—Mobile edge computing (MEC) enables users to offload computing tasks to edge servers for provisioning low-latency and computation-intensive services. To manage heterogeneous resources and improve service flexibility, MEC is entailed by new technologies, *i.e.*, software defined networking (SDN) and network function virtualization (NFV), which allow services running on common commodity hardware instead of proprietary hardware. However, data processing via software on commodity servers may induce high latency due to limited processing capacity, which impedes the quality of service. Meanwhile, MEC is a resource-sharing system and thus fairness should be considered. In this paper, we propose a network function parallelism (NFP)-enabled MEC (NFPMec) framework for supporting low-latency services. To reap the potential benefits of the NFPMec, we formulate the fairness-aware throughput maximization problem (FTMP) with aim of maximizing the fairness-aware system throughput while satisfying the QoS requirements. We propose a relaxation-based generalized benders algorithm (RGBA) to decouple the FTMP into two sub-problems based on the non-linear convex duality theory. After relaxation, the sub-problems are solved by the Karush-Kuhn-Tucker (KKT) approach. The convergence of the RGBA is theoretically proved. The simulation results demonstrate that the proposed NFPMec outperforms SDN-enabled MEC networks in terms of resource utilization, service latency and system throughput.**

***Index Terms*—Network slicing, low-latency, MEC, network function parallelism, NFP-enabled MEC framework.**

I. Introduction

The fifth generation and beyond (5G/B5G) mobile communication networks are expected to support billions of intelligent, interoperable, and connected physical devices at edge networks, which will change all aspects of our lives [1]. As these physical devices usually have limited computation capability, some computing tasks are usually offloaded to edge-side cloud for processing, which is known as mobile edge computing (MEC) [2]. A typical example of MEC is Cloudlet (developed by Carnegie Mellon University [3]), which manages to provide computation, communication and cache resources for users via edge-side cloud rather than remote cloud.

Recently, MEC embraces the new technology, software defined networking (SDN)-enabled network slicing, to simplify network management, enhance service quality and improve resource utilization [4]. SDN is applied as a fundamental design principle toward 5G/B5G networks [5]. SDN decouples the control plane and the data plane, which allows network administrators to manage network services through virtualized network functions (NFs). Network function virtualization (NFV) enables an NF to be virtualized as a programmable software running on a virtual machine (VM), which is installed in a commodity server [6]. By using NFV, multiple data flows can share and scale NFs, including baseband processing unit functions (e.g., medium access control and radio link control) and evolved packet core functions (e.g., mobility management entity and serving gateway) in mobile networks [7]. The required NFs are chained as a logic service function chain (SFC) to provide a dedicated service. Data packets at the user side are aggregated into traffic flows according to service type, scheduled to traverse the required NFs and achieve desired service quality. Multiple SFCs share the computation, communication and cache resources of the substrate network, which should be properly sliced among traffic flows to achieve performance isolation by the contoller, which is called *network slicing*. However, the integration of network slicing and MEC has received much research interest [4], [8]–[10].

In spite of the benefit of flexibility provided by NFV, data processing via software on a commodity server may induce high latency due to limited processing capacity (e.g., Ananta Software Muxes running on commodity servers can add from 200μs to 1ms latency at 100 Kpps [11], [12]). Moreover, the SFC latency grows linearly with the length of the chain. This impedes provisioning low-latency services, such as e-health data analytic and smart transportation at edge networks. Fortunately, the authors of [12] proposed NF parallelism (NFP) technology as a solution to decreasing the SFC latency. Using

This work was supported by the Key Research and Development Projects under grant number 2020YFB1806804, National Science Foundation of China under grant number 61871099 and this work is supported by ZTE Industry-Academia-Research Cooperation Funds. The authors are with the school of the National Key Lab. on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313001, P. R. China.

NFP, 53.8% NF pairs can logically work in parallel, and 41.5% NF pairs can be paralleled without causing extra resource overhead [12]. In [11], we extend the principle of NFP to SFC parallelization graph (SPG) in a data centre, which is a logic entity to enable various low-latency services. In an SPG, parallel NFs can be organized according to the independence of NFs' processing. Packet processing latency can thus be significantly decreased as compared with that in the conventional sequential chain. Inspired by the similar idea of NFP, in this paper, we propose an NFP-enabled MEC framework, named NFPMec, to provide low-latency services in edge networks.

To reap the potential benefits of NFP and MEC, some key factors (*e.g.*, traffic flow rate, packet scheduling, packet retransmission and channel power allocation) influencing the latency of channel transmission, processing and queuing should be considered in the design of NFPMec. With the ever-growing amount of mobile devices, large traffic flow rate may deteriorate channel quality due to severe interference. The poor wireless channel quality may increase the number of automatic repeat requests and thus high channel transmission delay. The queuing latency may become longer when an NF is congested, especially when multiple flows are sharing the same NF. Therefore, it is necessary to jointly manage flow rate, packet scheduling and channel power allocation to decrease the latency. On the other hand, multiple users compete for the limited resources and thus fairness should be considered to achieve an equilibrium [13]. Hence, a comprehensive and fairness-aware control policy should be designed in the resource-sharing system. Many existing studies focus on the resource allocation for slices with guaranteed performance, while few studies focus on the fairness-ware policy design to achieve a desired trade-off between system revenue (e.g., system throughput) and performance isolation for slices.

To provide low-latency service by using our proposed NFPMec, we formulate an optimization problem with aim of maximizing the fairness-aware system throughput while guaranteeing the quality of service (QoS) requirements of individual slices. We devise a relaxation-based Generalized Benders algorithm (RGBA) to make the problem mathematically tractable. Numerical results demonstrate the advantage of incorporating NFP in MEC, the flexibility of introducing fairness metric, and the effectiveness of the proposed RGBA compared with benchmark algorithms. The main contributions of this work are listed as follows:

- We propose an NFP-enabled MEC framework, where a fairness-aware flow rate control scheme in conjunction with packet retransmission, power allocation and packet scheduling schemes to maximize the system throughout while satisfying the service quality of individual slices.
- We formulate a fairness-aware throughput maximization problem (FTMP) while taking into account heterogeneous resources (*i.e.*, wireless transmit power, computing resources and cache resources) and two-dimensional service requirements (*i.e.*, delay and successful transmission probability). The FTMP captures three trade-offs: 1) system throughput and service latency, 2) system throughput and reliability of the wireless transmission and 3) system throughput and fairness among multiple flows.
- We prove that FTMP is NP-hard and propose a relaxation-based Generalized Benders algorithm (RGBA) by exploiting non-linear duality theory. Using RGBA, non-convex FTMP can be solved by iteratively solving two convex sub-problems.

The rest of the paper is summarized as follows. Section II presents related work. Section III presents NFPMec. Section IV describes the system model and formulates the problem. The solution of the proposed optimization problem is elaborated in Section V. Section VI presents the simulation results. At last, Section VII concludes the paper.

II. Related Work

In the past decade, the fast development of air interface, multi-antenna, millimetre-wave and small-cell networks allows running computing services of mobile devices at the remote cloud data centre. However, the long propagation distance from the end user to the remote cloud results in long latency, which may significantly degrade the quality of experience of users. Mobile Edge Computing (MEC) [14] is proposed to equip cloud computing capabilities at network edge. In MEC, edge servers are distributed across the network and closely connected to edge nodes such as cellular base stations or wireless access points. As a result, mobile devices can access the edge servers for cloud services directly via Radio Access Network (RAN), which reduces service delay and improves user experience quality. An overview of MEC platform was presented in [14], including different MEC frameworks and the corresponding application scenarios. Taleb *et al*. surveyed the key enablers of MEC, such as VM, NFV and SDN [4]. Juan Liu *et al*. investigated a two-timescale stochastic optimization problem [15]. Baidya *et al*. recently proposed a content and computation-aware traffic flow control framework in an SDN-based edge network [8]. Zichuan Xu *et al*. formulated a novel QoS-aware task offloading problem in a mobile edge-cloud network that consists of a number of Cloudlets co-located with Access Points (APs) [16]. Jie Feng *et al*. proposed a novel framework for network slicing in MEC systems and investigated the system revenue escalation problem [9]. Cziva *et al*. considered an NF placement problem in an SDN-enabled edge network, taking the end-to-end latency of data packets into consideration [10]. Kuo *et al*. proposed an SFC embedding approach to maximize the traffic flow while considering the node service capability [17].

While enjoying the benefits of NFV and SDN, the negative effect of using commodity server should be minimized, as the processing speed of commodity hardware is slower than that of the dedicated hardware [12]. Parallel programming is widely used in computing systems to reduce the latency of running software on multi-core platform. One of the famous parallel computing model is the MapReduce proposed by Google. Inspired by the parallel programming, NF parallelization has

been proposed as a processing acceleration technique for SFC, including flow-level, packet-level and program-level NF acceleration techniques. At the flow level, Long Qu *et al.* proposed to establish parallel backup routing paths for an SFC to improve the reliability and reduce end-to-end delay of packets [18]. Mihai Dobrescu *et al.* explored software router architecture that parallelizes router functionality across multiple servers to improve throughput [19]. At the packet level, Chen Sun *et al.* [12] proposed that a packet and the copy of this packet could be simultaneously processed by two NFs, when the two NFs in the same service chain share no dependency and could work in parallel. More related work focusing on program-level NF parallelization by identifying dependency of the program codes of an NF can be found in [11].

Different from the existing studies, this paper proposes the NFPMec, which introduces a packet-level NFP-enabled MEC network for provisioning low-latency services. Intuitively, by jointly designing packet scheduling at network level and NFP at packet level, the proposed NFPMec framework outperforms conventional SDN-enabled MEC frameworks for provisioning low-latency services.

III. NFPMec Framework

Based on the network slicing framework specified by the 3rd generation partnership project (3GPP) [20], we propose the NFPMec framework as illustrated in Fig.1. There are four layers, *i.e.*, the control layer, service layer, network slice instance (NSI) layer and physical layer.

At the control layer, the centralized controller deployed in the MEC server controls the entire network, including the admission of network slice request, the management of network resources, and the design of control policy for the whole network. At the service layer, various services are provisioned by the service provider for users.

At the NIS layer, an NSI supports a specific type of service with a certain QoS requirement. In an NSI, the principle of NFP is extended to service parallelization graph (SPG), which is established according to the logical independence relationship (or parallel rules) between NFs. There are three logical components for constructing the SPG, *i.e.*, NFP policy specification scheme, NFP orchestrator and NFP infrastructure. Details of each component are given in [12]. A structure of NFs (instead of only a chain of NFs) can be constructed to fulfill the functionalities of the original SFC, with improved latency performance. The details of SPG construction can be found in [11].

At the physical layer, each user is connected to a wireless access point (AP), and each AP is connected to a cloudlet through a wired link. An example of the packet processing procedure is shown in the physical layer in Fig.1, where a packet is processed at NF f1 instantiated in the AP server; Simultaneously, it is copied at the AP server and then transmitted to NF f2 instantiated in the Cloudlet server for processing; Finally, the two packets that have been processed are merged at the destination node. The details of merging packets refer to [12].

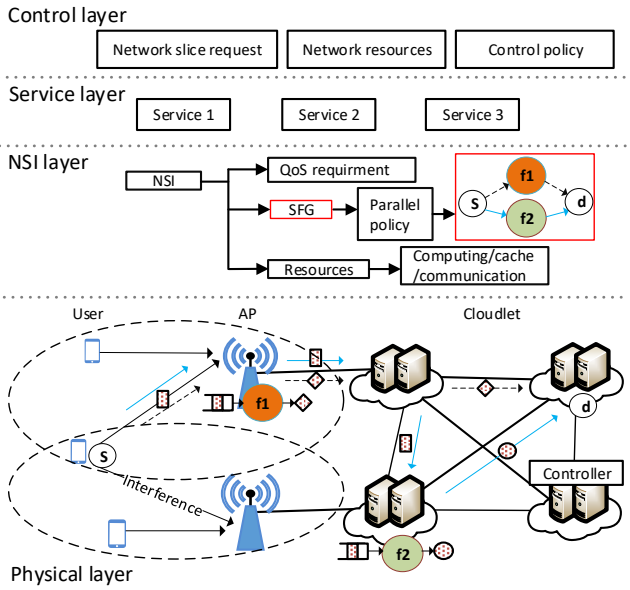


Fig. 1. Illustration of the NFP-enabled Mobile Edge Computing Framework

By using NFP, an SFC may introduce light-weight packet copying, which could consume extra link bandwidth resources [12]. In our previous research [11], we found that the cost of NFP, *i.e.*, bandwidth resource, hardly affects the end-to-end latency of packets. Moreover, we assume that each physical link has sufficient bandwidth capacity as given in assumption 7) [16]. Hence, in this paper, we do not consider the cost of NFP. Similar to [9], [10], [16], [17], we make the following assumptions:

1) A slice uses a full buffer traffic model[1].
2) A mobile device/user is associated with one slice and uses the band that has been allocated. A mobile device can allocate power to the band being used.
3) The destination node of data is at the AP/Cloudlet side, and thus a downlink transmission is not considered.
4) Nodes running VMs can support any type of NFs. The computing and cache capacity required by each type of NFs are fixed.
5) A service request is rejected if the flow rate of the service at the source node is controlled to be zero according to the designed RGBA based scheme.
6) A packet retransmission scheme under the statistical evaluation of channel quality is enabled to ensure packets to be successfully received by the AP [22].
7) Each physical link between Cloudlets has sufficient bandwidth capacity for supporting serving.

Based on the above assumptions, we consider the configuration policy of individual slices, which consists of packet rate control, computation offloading, transmit power allocation, packet retransmission and packet scheduling.

[1]There is an infinite amount of data bits awaiting to be transmitted in the output buffer associated with each data source [21].

For ease of presentation, we define some notations. Denote by $\mathbb{V}$ the set of servers, $\mathbb{I}$ the set of users, v_0^i the server of user $i \in \mathbb{I}$, and $\mathbb{S}$ the set of servers installed both in Cloudlets and APs. Thus $\mathbb{V} = \{v_0^i \cup \mathbb{S} | \forall i \in \mathbb{I}\}$. According to the assumptions, each mobile device/user has one slice request. Similar to [9], each slice request has a QoS requirement, *i.e.*, the latency requriemnt denoted by D_i. A new metric, successful transmission probability, is considered for slice request, which is introduced in Section IV. We denote by K_i the set of arriving packets per time slot of user i to be scheduled. We model the SPG as a weighted directed graph $\mathcal{G}_i = (\mathbb{J}_i, \mathbb{E}_i)$, where $\mathbb{E}_i$ is the set of virtual links connected to NFs and $\mathbb{J}_i$ is the set of NFs required by slice request $i \in \mathbb{I}$. The SPG design is based on the algorithm proposed in our previous work [11]. Denote by y_i the transmit power of user $i \in \mathbb{I}$. Set $y_i \leq Y_i^{max}$ where Y_i^{max} is the maximal power at user i. We define packet scheduling policy as a 0-1 variable $x_{v,k,i,j}$ with $x_{v,k,i,j} = 1$ indicating that packet $k \in K_i$ requiring NF $j \in \mathbb{J}_i$ is scheduled to server $v \in \mathbb{V}$ and $x_{v,k,i,j} = 0$ otherwise. We define flow rate control policy as a 0-1 variable $\overline{x}_{k,i}$ with $\overline{x}_{k,i} = 1$ indicating that packet $k \in K_i$ is admitted to the network and $\overline{x}_{k,i} = 0$ otherwise. We define the maximum number of packet retransmission trails of packet $k \in K_i$ as an integer variable $x_{k,i}^{(u)}$.

TABLE I
PARAMETERS AND VARIABLES

Notation	Parameter
Achievable uplink transmission rate between user i and AP a	$r_{i,a}$
Fairness tuning parameter of service i	a_i
Outage probability of user i	P_i^{out}
Server CPU processing speed of one packet at node v	C_v
Server cache capacity of node v	H_v
Set of arriving packets at user i to be scheduled	K_i
Set of Access Points	$\mathbb{A}$
Set of servers in Cloudlets	$\mathbb{S}$
Set of NFs required by service i	$\mathbb{J}_i$
Set of parallel group of service i	Z_i
Notation	**Variable**
Indicator variable that whether packet k of service i requiring NF j is scheduled to server v or not	$x_{v,k,i,j}$
Fraction of flow of service i requiring NF j scheduled to server v	$x_{v,i,j}$
The number of (re)transmissions of packet k of service i requiring NF j under link outages	$x_{k,i}^{(u)}$
The number of (re)transmissions of flow of service i requiring NF j under link outages	$x_i^{(u)}$
Transmission power of user i	y_i

IV. MODEL AND PROBLEM FORMULATION

In this section, we begin with presenting the models of wireless channel, packet retransmission, packet scheduling and end-to-end packet delay. Then, we formulate the FTMP problem. At last, we analyse the complexity of the problem.

A. Wireless Channel

Mobile devices using the same frequency bands suffer from co-channel interference. Denote by Q_i the set of users using the same channel frequency with user i.

According to Shannon theory, the achievable uplink transmission rate between user $i \in \mathbb{I}$ and the associated AP is given by

$$r_i(y) = Blog_2(1 + \gamma_i(y)), \tag{1}$$

where y is the set of $y_i, i \in \mathbb{I}$, B is the channel bandwidth, $\gamma_i(y) = \frac{A_i(y_i)}{B_i(y)}$ is the signal-to-interference-plus-noise ratio (SINR) of user i. Specifically, we have $A_i(y_i) = |h_{i,j}|^2 y_i$ and $B_i(y) = \sum_{j \neq i, j \in Q_i} |h_{i,j}|^2 y_j + \delta^2$, where $h_{i,j}$ is the channel gain between user i and AP j, and δ^2 is the average power of additive white Gaussian noise. During each signalling period, devices can periodically report the measured channel quality indicator (CQI) to the associated AP. Parameter $h_{i,j}$ can be updated based on the received CQI and power allocation policy [23].

Similar to [24] and [25], we assume the small-scale Rayleigh fading with path loss model for the wireless channel. At the physical layer, the packet successful transmission probability can be modelled as the channel link outage probability model [25], which is given by

$$\begin{aligned} P_i^{out}(y) =& \Pr[\gamma_i(y) < \gamma_{th}] \\ =& 1 - e^{-\gamma_{th}/\gamma_i(y)}, \end{aligned} \tag{2}$$

where γ_{th} is the predefined SINR threshold for receiving a bit [25]. Naturally, the successful transmission probability of a packet is given by $1 - P_i^{out}(y)$.

B. Packet Retransmission

To enhance system performance in terms of the transmission reliability on wireless channel, a retransmission scheme is incorporated for the AP to successfully receive the packet in NEPMec. When an outage occurs, the packet will stay at the head of the buffer and be retransmitted until it is successfully received [22]. We define the transmission trial of one packet as a Bernoulli random variable which takes the value 1 (success) with probability $1 - P_i^{out}(y_i)$ and the value 0 (failure) with probability $P_i^{out}(y_i)$. Denote by $s_{k,i}^l$ the lth transmission trial of packet $k \in K_i$, and thus a collection of Bernoulli random variable $s_{k,i}^l$ is independent and identically distributed. We define that the maximum value of l is $x_{k,i}^{(u)}$. If a packet needs to be uploaded to MEC servers, the AP shall successfully receive the packet, which means that the probability that a packet is successfully received by the AP under the packet retransmission scheme equals one. Thus, we have

$$x_{k,i}^{(u)}(1 - P_i^{out}(y)) = \begin{cases} 1, if & \sum_{v \in \mathbb{V}/\{v_0^i | \forall i \in \mathbb{I}\}} x_{v,k,i,j} > 0, \\ 0, if & \sum_{v \in \mathbb{V}/\{v_0^i | \forall i \in \mathbb{I}\}} x_{v,k,i,j} = 0. \end{cases} \tag{3}$$

We give a brief analysis about $x_{k,i}^{(u)}$ is given here. According to the equations of reliable transmission (3) and channel transmission delay (10) , when the channel outage probability is large, $x_{k,i}^{(u)}$ is large, resulting in a large channel transmission delay. To meet the delay requirement, packets shall be

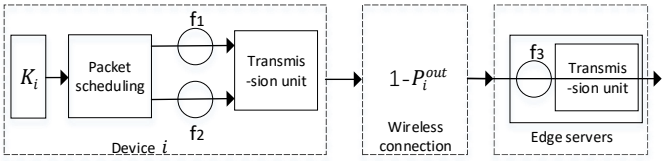


Fig. 2. The model of packet scheduling from a user to edge servers

scheduled to servers which have low processing and queuing delay when the channel transmission delay is large. If no such servers exist, the packets shall be rejected.

C. Packet Scheduling

Scheduling of packets enables efficient utilization of resources and load balancing in the NFPMec. Fig.2 gives an example of packet scheduling procedure from a user to servers in APs or Cloudlets. Given packet buffer K_i at user side, packet k, $k \in K_i$, requiring NF f1 and NF f2 is processed locally, and then uploaded to NF f3 for processing in the server at AP/Cloudlet side.

Constraint (4) indicates that packet k has to be scheduled to all required NFs to finish the service procedure. Constraint (5) ensures that packets processed by NF $j+1$ at the server have been processed by NF j in the same SPG, which avoids unnecessary round trip delay of wireless transmission and saves radio resources. Constraint(6) ensures that the packet can be scheduled only to the server with sufficient processing capacity, where C_v is the CPU processing speed for one packet at node v. Constraint (7) ensures that the packet can be scheduled only to the node with sufficient cache capacity, where w_j is the cache capacity required by NF j and H_v is the cache capacity of node v.

$$\overline{x}_{k,i} = \sum_{v \in \mathbb{V}} x_{v,k,i,j}, \forall j \in \mathbb{J}_i, \tag{4}$$

$$x_{v_0^i,k,i,j} - x_{v_0^i,k,i,j+1} \geq 0, \forall i, j, \tag{5}$$

$$C_v - \sum_{i,k} x_{v,k,i,j} \geq 0, \forall v, j, \tag{6}$$

$$H_v - \sum_{i,j,k} x_{v,k,i,j} w_j \geq 0, \forall v. \tag{7}$$

D. End-to-end Packet Delay

The delay of a packet traversing an SPG is defined as the sum of the processing delay, queueing delay and channel transmission delay. The propagation delay in links between two cloudlets is less than 0.1 ms [26]. Moreover, the number of hops between the source and sink nodes in an MEC network is much smaller than that in the data center. In an MEC network, compared with the end-to-end delay of packets traversing the slice, which ranges from 1 ms to 100 ms, the latency in backhaul links could be reasonably ignored [27].

$M/M/n$ queue model is adopted for deriving the queueing delay. The incoming packets are waiting in the queue of NF for processing. We denote by x' the vector of $x_{v,k,i,j}$, and by $d^q_{v,j}(x')$ the average queuing delay of each packet by node v for NF $j \in \mathcal{F}_i$. $d^q_{v,j}(x')$ is given by [16], i.e.,

$$d^q_{v,j}(x') = \frac{1}{C_v - \sum_{i \in \mathbb{I}} \sum_{k \in K_i} x_{v,k,i,j}}. \tag{8}$$

We denote by d_v the processing delay of one packet by an NF at server v with CPU processing speed C_v, which is given by

$$d_v = \frac{1}{C_v}. \tag{9}$$

The average channel transmission latency of one packet under packet retransmission is given by

$$d_i^{(u)} = \frac{x_{k,i}^{(u)}}{r_i(y)}. \tag{10}$$

For a sequential SFC, the delay of packet $k \in K_i$ is the summation of (8), (9) and (10), which is

$$T^1_{k,i}(x', y_i) = \sum_{j \in \mathbb{J}_i} \sum_{v: x_{v,k,i,j} > 0, k \in K_i} (d^q_{v,j}(x') + d_v) + d_i^{(u)}. \tag{11}$$

For an SPG, we denote by $\mathbb{J}_{i,l}^{(z)}$ the lth sub-chain in set z in the SPG of user i. The NFs in $\mathbb{J}_{i,l}^{(z)}$ are constructed in a sequential way. Meanwhile, sub-chain $\mathbb{J}_{i,l}^{(z)}$ can work in a parallel way with sub-chain $\mathbb{J}_{i,l'}^{(z)}$ in the same set z. We denote by $|Z_i|$ the number of such parallel sets in the SPG of user i. In an SPG, (11) can be rewritten as

$$T^2_{k,i}(x', y_i) = \sum_{z \in Z_i} \max_l \{ \sum_{j \in \mathbb{J}_{i,l}^{(z)}} \sum_{v: x_{v,k,i,j} > 0, k \in K_i} (d^q_{v,j}(x') + d_v) \} + d_i^{(u)}. \tag{12}$$

This is a generalized delay model which can represent the delay of both sequential SFCs and SPGs. Therefore, (12) will be applied in the following problem formulation.

E. Optimization Problem Formulation and Analysis

In this section, we formulate the problem with the objective of maximizing the fairness-aware system throughput. As the total average throughput of the flow is the number of admitted packets in a fixed duration, the total average throughput of user i is defined as $\eta_i = \sum_{k \in K_i} \overline{x}_{k,i}$. According to Lemma.2 in [28], we introduce the utility function to consider the fairness among service flows, which is given as

$$U_i(\eta_i, a_i) = \begin{cases} (1-a_i)^{-1} \eta_i^{1-a_i}, if \quad a_i \neq 1, \\ log(\eta_i), \quad if \quad a_i = 1, \end{cases} \tag{13}$$

where a_i is a fairness tuning parameter of user i. When $a_i = 0$, the problem is a throughput maximization problem. A greater a_i means that user i prefers more throughput than fairness or user i intends to pay more to get better service. Then, the fairness-aware throughput maximization problem (FTMP) is

formulated as the following **Problem 1** (P1).

$$P1: \max_{x_{k,i}^{(u)}, \overline{x}_{k,i}, x_{v,k,i,j}, y_i} \sum_i U_i(\eta_i, a_i) \tag{14}$$

$$s.t. \quad D_i - T_{k,i}^2(x', y_i) \geq 0, \forall i, \tag{14a}$$

$$(3) - (7), \tag{14b}$$

$$x_{k,i}^{(u)} \in \{0, 1, ...\}, \overline{x}_{k,i}, x_{v,k,i,j} \in \{0, 1\}, y_i \in [0, Y_i^{max}], \tag{14c}$$

where (14a) ensures that the delay requirement of each service is satisfied.

In this paper, we adopt (p,a)-proportional fairness for multiple users with NFP, which is a generalization of proportional fairness and max-min fairness [28].

Definition of (p,a)-proportional fairness: Let $p_1, \ldots, p_N$ and a be positive numbers, and then a vector of rates x^* is (p,a)-proportionally fair if it is feasible and for any other feasible vector x $\sum_i p_i \frac{x_i - x_i^*}{x_i^{*a}} < 0$.

Furthermore, consider the following optimization problem:

$$P: \quad \max g = \sum_{s \in S} U_s(x_s), \tag{15}$$

$$s.t. \quad A^T x \leq C, x \geq 0, \tag{16}$$

where a source-sink s is associated with a user; $U_s(x_s)$ is the utility associated with x_s; the objective is to maximize the aggregate utility of rates $x = \{x_s, s \in S\}$; the constraints are the network capacity constraints. According to Lemma 2 in [28], the vector x^* solves the problem P with the utility function if and only if x^* is (p,a)-proportionally fair. Since the problem P1 has the same structure as the problem P, the Lemma 2 [28] holds for the problem P1 as well.

Then, we prove that the formulated FTMP is NP-hard. The way of the proof is reducing the FTMP to a problem that is a special case or generalized instance of the FTMP (called SFTMP). Then, we reduce the SFTMP to a well-known NP-hard problem, *i.e.*, Bin Packing (BP) problem, as given in Proposition 1. The time used in these two reductions is both polynomial. Thus, the FTMP is also NP-hard. Formally, we give the definition of BP and SFTMP as following,

Definition 1. The BP is denoted by $BP = (Q, B)$, where Q is the item set and B is the bin set, items in Q of different sizes are to be packed into bins in B with fixed sizes so as to maximize the number of packed items. The size of bin $b \in B$ is R-dimensional value with $e_b = (e_{b,1}, \cdots, e_{b,R})$, where $e_{b,r}$ denotes the size of bin b in rth dimension. The size of item is $q \in Q$ with $m_q = (m_{q,1}, \cdots, m_{q,R})$, where $m_{q,r}$ denotes the size of item in rth dimension.

Definition 2. In the SFTMP, suppose that we are given an FTMP with known fairness factor a_i, $a_i \in [0, 1]$ and fixed power allocation scheme $\overline{y}$ (we design an algorithm to find $\overline{y}$ in polynomial time). The SFTMP is denoted by $SFTMP = (K, \mathbb{V}, D_i, C_v, H_v)$, where $K = \cup_{i \in \mathbb{I}} K_i$. SFTMP aims at maximizing the number of packets allowed to be inserted in to the network and guarantees latency requirements

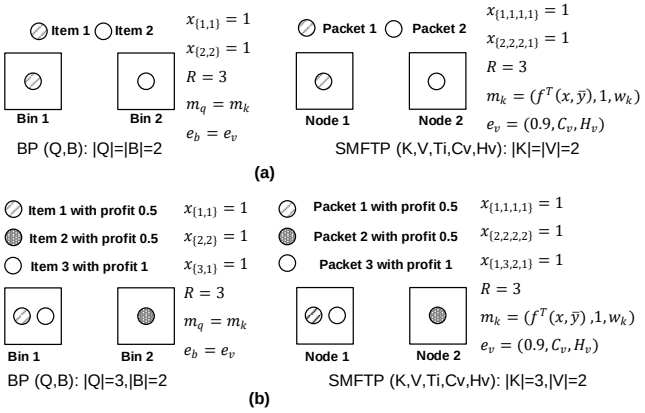


Fig. 3. Two examples of reduction from the BP to the SFTMP.

$(D_i, \forall i \in K_i)$ without exceeding server computing and cache capacity (C_v and $H_v, \forall v \in \mathbb{V}$).

Proposition 1. The reduction from SFTMP to BP uses polynomial time (BP$<_p$ SFTMP).

Proof: For each instance $BP = (Q, B)$, we construct an instance $SFTMP = (K, \mathbb{V}, T_i, C_v, H_v)$ as follows. As an example shown in Fig.4 (a), there are two packets requiring NF1 and NF2 respectively. First, set K such that packet k is in K if and only if item q is in Q, where q is called $k's$ corresponding item in Q. The size of packet k is the same as $k's$ corresponding item in Q. Set $R = 3$, as there are three constraints in SFTMP, *i.e.*, latency, computing capacity and cache capacity. Set $m_{k,1} = f^T(x, \overline{y}, D_i)$, $m_{k,2} = 1$ and $m_{k,3} = w_k$, where $f^T(x, \overline{y}, D_i)$ is a function that takes value 1 if $D_i < T(x, \overline{y}), k \in \mathbb{K}_i$ and 0 otherwise; and $w_k = w_j$ means that packet k requires NF j. Set $\mathbb{V}$ such that server v is in $\mathbb{V}$ if and only if bin z is in B. The size of each bin is the same as its corresponding item. For the bin size, set $e_{v,1} = 0.9$, $e_{v,2} = C_v$ and $e_{v,3} = H_v$. The reasons of setting $e_{v,1} = 0.9$ is that a scheduling solution x is not feasible if packet k is scheduled to node v under x violating delay constraint, *i.e.*, $f^T(x, \overline{y}, D_i)$ is set to 1 which is larger than 0.9 if $D_i < T(x, \overline{y}), k \in \mathbb{K}_i$. Clearly, in this instance $SFTMP = (K, \mathbb{V}, T_i, C_v, H_v)$, the corresponding items and bins in $BP = (Q, B)$ can be constructed in polynomial time. In addition, let x_{BP} with $x_{q,b} = 1$ be a solution to $BP = (Q, B)$ and x with $x_{v,k,i,j} = 1$ be a solution to SFTMP such that packet k is scheduled into node v if and only if item q is packed into bin b. Obviously, x is a feasible solution to $SFTMP$ if and only if x_{BP} is the solution to BP. Fig.4 (b) shows an example of the extension version of BP, which is used to show a more generalized reduction. There are two packets, one requiring NF1 and NF2, and the other only requiring NF1. Items in Q of different sizes are to be packed into bins in B with fixed given sizes to maximize the sum of profit of packed items. Based on that the profit of packet k is set to $1/|\mathbb{J}_i|$, we set the profit of these two packet to $1/2$ and set the profit of the latter packet to 1. Then, an extension version of BP is

reduced to the SFTMP. ■

V. The Proposed Algorithm for Throughput Maximization

In this section, we present the relaxation-based Generalized Benders algorithm (RGBA) to solve the FTMP and prove the convergence of RGBA. The time complexity of the proposed RGBA is also analysed.

A. Relaxation-based Generalized Benders Algorithm

The FTMP is an NP-hard problem which is reduced from the BP or the extension version of BP. The common approximate algorithms for integer problems are not feasible, such as Next-fit, First-fit, Best-fit and Worst-fit, as the FTMP is formulated as a mixed-integer problem. An approach is to transform the FTMP to a relaxation-based problem with continuous variables, which has a linear objective function and nonlinear constraint functions. By exploring and exploiting the structure of nonlinear constraint (14a), we propose the relaxation-based generalized benders (RGBA) algorithm based on the non-linear convex duality theory, which decouples the FTMP that is non-convex into two convex sub-problems. To solve the FTMP, we shall first determine a feasible SPG for the service request by the SPG construction algorithm in [11].

The proposed RGBA consists of four steps. In **STEP 1**, the FTMP is transformed to the relaxation-based FTMP (RFTMP) with continuous variables. In **STEP 2**, we decompose the RFTMP into two subproblems. In **STEP 3**, we solve the sub-problem 1 by Karush-Kuhn-Tucker (KKT) approach [29], and the sub-problem 2 by convex approximation and KKT approach. In **STEP 4**, we determine the value of $x_{k,i}^{(u)}$, and whether each $x_{v,k,i,j}$ and $\overline{x}_{k,i}$ should be 0 or 1 based on the relaxed solutions. The RGBA is summarized in Algorithm 1.

STEP 1: Before presenting the RFTMP, we define some parameters as follows. We denote by λ_i the average flow rate ($\lambda_i = |K_i|$) of service i, $x_{v,i,j}$ the fraction of flow of service i requiring NF j scheduled to server v, and $\overline{x}_i$ the fraction of flow of service i admitted into the network, where λ_i is constant and $x_{v,i,j}$, $\overline{x}_i \in [0,1]$. By relaxing $x_{k,i}^{(u)}$, $\sum_{k\in K_i} x_{v,k,i,j}$ and $\sum_{k\in K_i} \overline{x}_{k,i}$ in the original FTMP to $x_i^{(u)} (x_i^{(u)} \in R^+)$, $\lambda_i x_{v,i,j}$ and $\lambda_i \overline{x}_i$, respectively, we get the following RFTMP.

$$RFTMP: \max_{x_i^{(u)}, \overline{x}_i, x_{v,i,j}, y_i} \sum_i U_i(\eta_i, a_i) \tag{17}$$

$$s.t. \quad D_i - T_i^2(x'', y_i) \geq 0, \forall i, \tag{17a}$$

$$(3) - (7), \tag{17b}$$

$$x_i^{(u)} \in R^+, \overline{x}_i, x_{v,i,j} \in [0,1], y_i \in [0, Y_i^{max}], \tag{17c}$$

where x'' is the vector of $x_{v,i,j}$, $T_i^2(x'', y_i) = \sum_{z\in Z_i} \max_l \{\sum_{j\in \mathbb{J}_{i,l}^{(z)}} \sum_{v:\lambda_i x_{v,i,j}>0} (\frac{1}{C_v - \sum_{i\in\mathbb{I}} \lambda_i x_{v,i,j}} + d_v)\} + \frac{x_i^{(u)}}{r_i(y)}$; constraint (3) is relaxed to $x_i^{(u)}(1 - P_i^{out}(y)) = 1$, if $\sum_{v\in\mathbb{V}/\{v_0^i|\forall i\in\mathbb{I}\}} x_{v,i,j} > 0$ and $x_i^{(u)}(1 - P_i^{out}(y)) = 0$, if $\sum_{v\in\mathbb{V}/\{v_0^i|\forall i\in\mathbb{I}\}} x_{v,i,j} = 0$; constraint (4) is relaxed to $\overline{x}_i = \sum_{v\in\mathbb{V}} x_{v,i,j}, \forall j \in \mathbb{J}_i$; the other constraints in (17b) are correspondingly relaxed, and we do not list them.

STEP 2: We introduce to use the generalized benders (GB) to decompose the RFTMP. We exploit the special structure of $P1$, *i.e.*, $P1$ is not concave in x and y jointly, but fixing y renders the problem concave in x, where x is the vector of variables $x_{v,i,j}$, $\overline{x}_i$ and $x_i^{(u)}$, and y is the vector of variables y_i. We notice that the RFTMP is non-concave since constraint $(17a)$ is non-concave (the second factor at the RHS is non-concave in delay model (12)). However, if we fix y_i in (12), (17a) is concave due to the property that if $g^1(x)$ and $g^2(x)$ are convex, $h(x) = \max\{g^1(x), g^2(x)\}$ is convex. Based on the above property of of $P1$, we can project the RFTMP into the master problem (P2) by using GB approach.

$$P2: \max_{y\in Y, z} z \tag{18}$$

$$s.t. \quad z \leq \text{supremum}_{x\in X}\{f(x,y) + u^t G(x,y)\}, u \geq 0, \tag{18a}$$

$$\text{supremum}_{x\in X}\{\omega^t G(x,y)\} \geq 0, \omega \in \Omega, \tag{18b}$$

$$\Omega = \{\omega \in R^m | \omega \geq 0, \sum_{i=1}^{m} \omega_i = 1\}, \tag{18c}$$

where $f(x,y) = \max_{x,y} \sum_i U_i(\eta_i, a_i)$ is the objective of the RFTMP; $G(x,y) = (g_1(x,y), \cdots, g_m(x,y))$ is the vector of constraints of the RFTMP; (18b) ensures that all constraints are satisfied, *i.e.*, $y \in Y \cap V, V := \{y : G(x,y) \geq 0 \text{ for some } x \in X\}$. The explicit expression of $G(x,y)$ can be found in Appendix B. Theorem 1 provides theoretical support that the $P2$ is equivalent to the original RFTMP.

Theorem 1. The RFTMP can be constructed to an equivalent problem with the form of $P2$.

Proof: The proof is given in Appendix. ■

Solving the RFTMP directly is hard, but we can use the non-linear dual method to solve the $P2$ effectively. By using the dual method, the procedure of solving $P2$ that is equivalent to iteratively solving $P3$ and $P4$. The iterative algorithm involving the $P3$ and $P4$ proceeds until an acceptable tolerance is reached. The formal statement of using the dual method is given as below.

1) Let an initial point $\overline{y}$ satisfy $G'(x,y) \geq 0$ with some $x \in X$, an initial ω_0 satisfy (18c) and choose a convergence tolerance parameter ϵ. Set $p = 1$, $q = 0$, $LBD = 0$.
2) Next we solve $P3$ (the first sub-problem), which is given by

$$P3: \ L^*(y;u): \ = \text{supremum}_{x\in X}\{f(x,y) + u^t G(x,y)\}, \quad y \in Y, u \geq 0, \tag{19}$$

When solving $P3$ with fixed y, *i.e.*, $y = \overline{y}$, we need to determine u. We obtain an optimal or near-optimal multiplier vector $\overline{u} \in U(\overline{y})$ (*e.g.*, by using interior-point algorithm), where $U(y)$ is the set of optimal solutions to the dual of $P1$. If no such u exists, a near-optimal solution

satisfying (11-1) in [30] can be calculated. Then, we can get function $L^*(y;\overline{u})$. Increase p by 1 and put $u^p=\overline{u}$. If $L^*(\overline{y};\overline{u})>LBD$, set $LBD=L^*(\overline{y};\overline{u})$ as a lower bound of the optimal value of the RFTMP.

3) Denote $L_*(y;\omega) := \text{supremum}_{x\in X}\{\omega^t G(x,y)\}, y \in Y, \omega \geq 0$. Determine $\tilde{\omega}$ by $\text{superemum}_{x\in X}\{\tilde{\omega}G'(x,\overline{y}) \leq 0\}$ in (18c) and the function $L_*(y;\tilde{\omega})$. Increase q by 1 and set $\omega_q=\tilde{\omega}$. Solve the current master problem (the second sub-problem) with known u^p

$$P4: \max_{y\in Y,z} z, \tag{20}$$

$$s.t. \quad z \leq L^*(y;u_p), \tag{20a}$$

$$L_*(y;\omega_q) \geq 0. \tag{20b}$$

We obtain an optimal solution $(\tilde{y},\tilde{z})$, where $\tilde{z}$ is an upper bound on the optimal value of the RFTMP. If $LBD \geq \tilde{z}-\epsilon$, the algorithm terminates; else, $\overline{y}=\tilde{y}$ and go to 2).

STEP 3: The details of solving $P3$ and $P4$ are shown as bellow.

Solve P3: $P3$ aims at maximizing the system throughput at both user side and edge server side by optimally scheduling the traffic flow with fixed power allocation. Since $P3$ is a concave problem on X with fixed y, solving $P3$ is trivial. In this study, we use KKT approach to solve it.

Solve P4: However, solving $P4$ is not easy, as the function $L^*(y;u)$ and $L_*(y;\omega)$ have no explicit forms. Fortunately, we can observe that $f(x,y)$ and $G(x,y)$ in $L^*(y;u)$ and $L_*(y;\omega)$ are linearly separate. Based on this observation, we can get the explicit closed-form of $L^*(y;u)$ and $L_*(y;\omega)$ as

$$L^*(y;u) = C_1^* + u^t G_1'(x) + u^t G_2'(y), \tag{21}$$

$$L_*(y;\omega) = C_2^* + \omega^t G_2'(y), \tag{22}$$

where $C_1^* = \text{supremum}_{x\in X}\{f_1'(x)+u^tG_1'(x)\}$ and $C_2^* = \text{supremum}_{x\in X}\{\omega^t G_1'(x)\}$. The derivation is shown in Appendix. Because $f_1'(x)$ and $G_1'(x)$ are linear, we can easily get C_1^* and C_2^*. Then, $P4$ is rewritten as

$$\begin{aligned}&\max_{y\in Y,z} z\\ &z \leq C_1^* + u^t G_2'(y),\\ &C_2^* + \omega^t G_2'(y) \geq 0.\end{aligned}$$

Due to the constant part C_1^* in the first constraint, $P4$ has an equivalent form of

$$\begin{aligned}&\max_{y\in Y} u^t G_2'(y)\\ &C_2^* + \omega^t G_2'(y) \geq 0.\end{aligned}$$

Based on the above proposition, $P4$ under $u^p = (u_1^{p,i}, u_2^{p,i}, \cdots, u_m^{p,i})$ is rewritten as the following $P5$.

$$P5: \max_{y\in Y} \sum_{i\in\mathbb{I}} (u_{j,1}^{(i)} g_{1,2}^{\prime,(i)}(y) + u_{j,2}^{(i)} g_{2,2}^{\prime,(i)}(y)), \tag{23}$$

$$s.t.\ C_2^* + \sum_{i\in\mathbb{I}} (\omega_{j,1}^{(i)} g_{1,2}^{\prime,(i)}(y) + \omega_{j,2}^{(i)} g_{2,2}^{\prime,(i)}(y)) \geq 0, \tag{23a}$$

where $g_{1,2}^{\prime,(i)}(y) = -\frac{1}{B(1-P_i^{out}(y))log_2(1+\frac{A_i(y)}{B_i(y)})}$ and $g_{2,2}^{\prime,(i)}(y) = 1-P_i^{out}(y) = e^{-\gamma_{th}\frac{B_i(y)}{A_i(y)}}$ according to Appendix B.

Observe that (23) is non-concave and (23a) is also non-concave due to SINR $\frac{A_i(y)}{B_i(y)}$. We cannot use the general concave optimization methods to solve this problem. To solve this issue, we first relax constraint (23a) to (24a) by setting y_i in $B_i(y)$ to y_j^m that is the maximum transmit power of user $j \in Q_i$, *i.e.*, $B_i(y) = \sum_{j\neq i, j\in Q_i} |h_{j,a_i}|^2 y_j^m + \delta^2$. In other words, when considering the SINR threshold of all users, we assume that the maximum interference is received by each user. To decrease the influence of this assumption on the solution to this problem, the protection ratio for receiving a bit could be set to a suitable value that is smaller than the predetermined one. Then, we rewrite $P5$ as $P6$, and introduce Proposition 2.

$$P6: \max_{y\in Y} \sum_{i\in\mathbb{I}} (u_{j,1}^{(i)} g_{1,2}^{\prime,(i)}(y) + u_{j,2}^{(i)} g_{2,2}^{\prime,(i)}(y)), \tag{24}$$

$$s.t.\ y \in \tilde{Y}. \tag{24a}$$

Proposition 2. *The corresponding constraint set $\tilde{Y}$ in (24a) is a non-empty concave set.*

Proof: The proof is given in Appendix. ∎

After converting the constraint of $P5$ to a concave one, we need to convert the objective (24) to a concave one. Indeed, $P6$ is a concave-convex sum-of-functions-ratio problem (CCSP) belonging to fractional programming (FP) (*i.e.*, a family of optimization problems containing ratio term(s), *e.g.*, SINR $r_i(y) = \frac{A_i(y)}{B_i(y)}$ from multiple interfering links) [31]. We can convert the original non-convex problem $P6$ into a convex problem by using FP theory [31]. The idea of FP theory is to tackle the FP problem by introducing a set of auxiliary variables and decoupling the numerator and denominator in $r(y)$. In the following, we first prove that $P6$ is a CCSP with Proposition 4. Then, $P6$ is transform into an equivalent problem, $P7$. In this way, we can use Algorithm 1 proposed in [31] to solve $P6$.

Proposition 3. *$P6$ belongs to CCSPs.*

Proof: A problem belongs to CCSPs if and only if the following conditions are satisfied, (1) Numerators $A_i(y)$ in $r_i(y)$ are all concave functions; (2) Denominators $B_i(y)$ in $r_i(y)$ are all convex functions; (3) The constraint set $\tilde{Y}$ is a non-empty concave set; (4) A sequence of functions $f_i(\cdot)$ in the objective function $\sum_{i\in\mathbb{I}} f_i(r_i(y))$ are not only non-decreasing, but also concave.

It is easy to observe that $A_i(y)$ and $B_i(y)$ are both linear functions. As a linear function is convex and also concave, (1) and (2) are satisfied. (3) is also satisfied according to the relaxed constraint (24a). Now consider (4). Let $f_i(r_i) = u_{j,1}^{(i)}\hat{g}_{2,2}^{(i)}(r_i) + u_{j,2}^{(i)}\hat{g}_{2,2}^{(i)}(r_i)$ where $\hat{g}_{1,2}^{(i)}(r_i) = -\frac{1}{B\hat{g}_{2,2}^{(i)}(r_i)log_2(1+r_i)}$ and $\hat{g}_{2,2}^{(i)}(r_i) = e^{-\gamma_{th}/r_i}$ according to (24). We can easily find that $f_i(r_i)$ is non-decreasing. The proof of concavity of $f_i(r_i)$ is omitted here due to limited pages. As the sum of concave functions is also concave, (4) is satisfied.

Algorithm 1 Relaxation-based Generalized Benders Algorithm

Input: Operator $\{\mathbb{V}, H_v, C_v\}$, User $\{\mathbb{I}, Z_i, Y_i^{max}, D_i, K_i\}$
Output: x, the vector of $x_i^{(u)}, \overline{x}_i, x_{v,i,j}$ and y, the vector of y_i
Initialization SPG construction algorithm [11].
Step 1 Problem relaxation
Step 2 and 3 Initialize $y \leftarrow \overline{y}$,$p \leftarrow 0$, $q \leftarrow 0$, $\epsilon \leftarrow \tilde{\epsilon}$
repeat
  $u_p \leftarrow \overline{u}$ (solve dual of $P1$)
  $p = p + 1$
  $x \leftarrow L^*(\overline{y}; \overline{u})$ (solve $P3$)
  repeat
    $\tilde{w} \leftarrow w \in \{w | \text{superemum}_{x \in X}\{\tilde{\omega} G'(x, \tilde{y}) \leq 0\}, w \in \Omega\}$
    $q \leftarrow q + 1$
    $w_q \leftarrow \tilde{w}$
    $\eta_i^* \leftarrow (26)$
    $\overline{y} \leftarrow y$ (solve $P7$ by Algorithm 1 in [31])
  until Convergence
  Return $\overline{y}$
  $\tilde{z} \leftarrow C_2^* + \sum_{i \in \mathbb{I}} f_i(s_i(\overline{y}, \eta_i^*))$
until Convergence
Step 4 Algorithm 2

■

According to Proposition 3, we can apply quadratic transform to $P6$ and rewrite it as P7 as below.

$$P7: \max \sum_{i \in \mathbb{I}} f_i(s_i(y, \eta_i)) = \sum_{i \in \mathbb{I}} (u_{j,1}^{(i)} \hat{g}_{1,2}^{(i)}(s_i(y, \eta_i)) + u_{j,2}^{(i)} \hat{g}_{2,2}^{(i)}(s_i(y, \eta_i)), \quad (25)$$

$$s.t.\ y \in \tilde{Y}, \eta_i \in \mathbb{R}, i \in \mathbb{I}, \quad (25a)$$

where $s_i(y, \eta_i) = 2\eta_i \sqrt{A_i(y)} - \eta_i^2 B_i(y)$ [31]. Then, we have the following critical propositions.

Proposition 4. *The optimal η_i has closed form as*

$$\eta_i^* = \sqrt{A_i(y)}/B_i(y), \forall i \in \mathbb{I}. \quad (26)$$

Proof: Fix y and set the first derivative of the objective (25) in $P7$ to zero, then we can get $\eta_i^* = \sqrt{A_i(y)}/B_i(y), \forall i \in \mathbb{I}$. ■

Based on Proposition 3 and 4, we have Proposition 5.

Proposition 5. *$f_i(s_i(y, \eta_i))$, $\forall i \in \mathbb{I}$, is concave in y for a fixed η_i.*

Proof: When η_i is fixed, due to the concavity of each $A_i(y)$, the convexity of each $B_i(y)$, and the concave square-root function $(\sqrt{\cdot})$, the quadratic transform $s_i(y, \eta_i) = 2\eta_i \sqrt{A_i(y)} - \eta_i^2 B_i(y)$ is concave in y for fixed η_i. Furthermore, as $f_i(\cdot)$, $\forall i$ is concave and non-decreasing according to Proposition 3, we have $f_i(s_i(y, \eta_i))$, $\forall i \in \mathbb{I}$, is concave in y for a fixed η_i. ■

Therefore, $P7$ is a concave optimization problem over y, which can be solved by using convex optimization methods such as the KKT and the interior-point algorithm.

STEP 4: After we get x by Algorithm 1, the next step to determine $x_{v,k,i,j}$, $\overline{x}_{k,i}$ and $x_{k,i}^{(u)}$ is presented in Algorithm 2. First, we determine $x_{k,i}^{(u)}$ by $x_{k,i}^{(u)} = \lceil x_i^{(u)} \rceil$ to satisfy constraint (3). We define $\overline{K}_{v,i,j} = \lceil \lambda_i x_{v,i,j} \rceil$. For $k \in \{1, 2, ..., \overline{K}_{v,i,j}\}$, $x_{v,k,i,j}$ is set to 1, and for $k \in \{\overline{K}_{v,i,j} + 1, \overline{K} + 2, ..., |K_i|\}$, $x_{v,k,i,j}$ is set to 0. To satisfy constraint (4), $x_{v,k,i,j}$ is set to 1 for $k \in \{\overline{K}_{v,i,j} + 1, \overline{K}_{v,i,j} + 2, ..., \overline{K}_{v,i,j} + \delta\}$ if $\delta = \sum_{v \in \mathbb{V}} x_{v,k,i,j} - \sum_{v \in \mathbb{V}} x_{v,k,i,j'} > 0, j \neq j'$. Then, constraint (4) is satisfied and $\overline{x}_{k,i}$ is set to $\sum_{v \in \mathbb{V}} x_{v,k,i,j}$. Then, we sort $\lambda_i \overline{x}_{k,i}$ in a decreasing order. Denote by A the threshold for rounding which equals the largest $\lambda_i \overline{x}_{k,i}$. While constraints (17a), (6) and (7) are not satisfied, for $k \in K_i$, $\overline{x}_{k,i}$ with $\lambda_i \overline{x}_{k,i} = A$, $\overline{x}_{k,i} = 1$ is set to 0, and corresponding $x_{v,k,i,j}$ with $x_{v,k,i,j} = 1, \forall j \in \mathbb{J}_i, v \in \mathbb{V}$ is set to 0. After $\overline{x}_{k,i}$ with $\lambda_i \overline{x}_{k,i} = A$ is determined, if there still exist violated constraints, a new iteration continues until all constraints are satisfied.

Algorithm 2 The algorithm of determining $x_{v,k,i,j}$, $\overline{x}_{k,i}$ and $x_{k,i}^{(u)}$

Input: $x_i^{(u)}, \overline{x}_i$ and $x_{v,i,j}$
Output: $x_{v,k,i,j}$, $\overline{x}_{k,i}$ and $x_{k,i}^{(u)}$
$x_{k,i}^{(u)} \leftarrow \lceil x_i^{(u)} \rceil$, $\overline{K}_{v,i,j} \leftarrow \lceil \lambda_i x_{v,i,j} \rceil$
for all $v \in \mathbb{V}, i \in \mathbb{I}, j \in \mathbb{J}_i, k \in \{1, 2, ..., \overline{K}_{v,i,j}\}$ **do**
  $x_{v,k,i,j} \leftarrow 1$
end for
for all $k \in \{\overline{K}_{v,i,j} + 1, \overline{K} + 2, ..., |K_i|\}$ **do**
  $x_{v,k,i,j} \leftarrow 0$
end for
if $\sum_{v \in \mathbb{V}} x_{v,k,i,j} - \sum_{v \in \mathbb{V}} x_{v,k,i,j'} > 0, j \neq j'$ **then**
  $\delta = \sum_{v \in \mathbb{V}} x_{v,k,i,j} - \sum_{v \in \mathbb{V}} x_{v,k,i,j'}$
  for all $k \in \{\overline{K}_{v,i,j} + 1, \overline{K}_{v,i,j} + 2, ..., \overline{K}_{v,i,j} + \delta\}$ **do**
    $x_{v,k,i,j} \leftarrow 1$
  end for
end if
$\overline{x}_{k,i} \leftarrow \sum_{v \in \mathbb{V}} x_{v,k,i,j}$
while Constraints (17a), (6) and (7) are not satisfied **do**
  $A = \max_{i \in \mathbb{I}}\{\lambda_i \overline{x}_{k,i}\}$
  if $\lambda_i \overline{x}_{k,i} = A, \overline{x}_{k,i} = 1$ **then**
    $\overline{x}_{k,i} \leftarrow 0$ and corresponding $x_{v,k,i,j} \leftarrow 0$
  end if
end while

B. Convergence and Complexity Analysis

According to Theorem 2.5 in [30], several conditions should be satisfied to guarantee the convergence of Algorithm 1. The first condition is that the feasible region of both x and y is non-empty and compact. Obviously, both X and Y are closed and bounded, and thus they are compact. The second condition is that f' and G' are concave over x for fixed $y \in Y$ and continuous on $X \times Y$. This is true as mentioned in subsection V-A. The third condition is that the multiplier vectors u of the dual of $P3$ is non-empty. This condition is satisfied once $P3$

has an optimal solution. As $P3$ is a concave programming, an optimal solution exists when the algorithm converges. As validated in [31], we can solve $P6$ (that is equivalent to $P3$) using the proposed algorithm and enable it converge to a stationary point in a finite steps. Above all, the convergence of Algorithm 1 is proved.

We analyse the time complexity of the proposed RGBA here. The maximum number of iterations, defined as L_1 of the inner layer and L_2 of the outer layer of the RGBA, is either set in advance or determined by appropriate stopping rule. The complexity of SPG construction is $O(|\mathbb{I}|\,|\mathbb{J}_i|^3 R)$, where R is the number of parallel rules for the service request. The complexity of the interior-point algorithm for solving P1 is $O\left(n^{3.6}\right)$, where n is the number of variables. We have $n = 3\,|\mathbb{I}| + \sum_{i \in <} |\mathbb{J}_i|\,|\mathbb{V}|$, where $|\mathbb{I}|$ is the number of users and $|\mathbb{V}|$ is the number of servers. The complexity of the KKT approach for solving P3 and the Algorithm 1 in [33] for solving P7 are polynomial time, *i.e.*, $O\left(n^{p1}\right)$ and $O\left(n^{p2}\right)$, respectively, where P1 and P2 are positive numbers independent of n. Thus, the running time of the RGBA is $L_2(O\left(n^{p2}\right) + L_1(O\left(n^{3.6}\right) + O\left(n^{p1}\right))) + O(|\mathbb{I}|\,|\mathbb{J}_i|^3 R)$.

VI. Performance Evaluation and Discussions

In this section, we compare the performance of the proposed RGBA with the optimal solution by brute force. As the running time of brute force algorithm is accepted for the problem with a small size, we consider a small-scale network, which has less than 10 servers. We also compare the performance of the RGBA with an upper bound by solving $P1$, where (14a) is omitted and the network is a large-scale one with more than 20 servers. On the other hand, to verify the effectiveness of incorporating NFP in the MEC framework, we compare the performance of the proposed RGBA in the NFPMec with that in the MEC. To demonstrate the effectiveness of the RGBA in small-scale NFPMec networks (RSN) and the RGBA in large-scale NFPMec networks (RLN), four benchmark algorithms are employed as below.

1) The algorithm of getting an optimal solution to P1 in the small-scale NFPMec (OSN): In the small-scale network, the optimal solution to P1 is obtained by using brute-force search algorithm.
2) The algorithm of getting an upper bound solution to a relaxed-based P1 in the large-scale NFPMec (ULN): In the large-scale network, the optimal solution of a relaxed linear problem is used as an upper bound of the solution to P1. This problem is set by omitting (14a) in P1.
3) The RGBA of getting a solution to P1 in the large-scale/small-scale MEC network without deploying the NFP (RLM/RSM): In the large-scale network, the sequential SFC structure is considered as the NFP is not deployed, which means that the delay model (11) rather than the general delay model (12) is used in P1.
4) The algorithm of getting an optimal solution to P1 in the small-scale MEC network without deploying the NFP (OSM): In the case of the small-scale network, the brute-force is used as that in the OSN. The delay model is (11) without deploying the NFP.

We examine the performance in terms of service flow rates, delay requirements and fairness factors and service latency, as well as the running time of the above algorithms. Similar to [32], [33], other simulation parameters are set as given in Table II. We conduct the simulations in in a simulation platform based on MATLAB R2019b.

TABLE II
Simulation parameters

Parameter	Value
Channel bandwidth	10 MHz
Delay requirements	$2-5$ ms
Fairness factor	$0-1$
Maximum transmit power of a user	1 W
Noise density	10^{-15} W/Hz
NF processing overhead at one node	$7-9$ MB/Mbps
Number of users	10
Number of servers	8
Path loss	3.5
Packet size	264 bits
Processing rate of a user	1 bit/us
Scale parameter of Rayleigh distribution	1
SNIR threshold	25dB

Fig.4 shows the system throughput as a function of the service flow rate λ. In the small-scale network, as shown in Fig.4 (a), we can see that the system throughput of both OSN and OSM increases linearly with service flow rate at the beginning. Then, the knees of the curves of OSN and OSM occur. This is because the available resources are reduced with the service flow rate. The knee of the curve of OSM occurs earlier than that of OSN, which demonstrates that the NFP improves the system throughput compared with the MEC network without deploying NFP. The advantage of NFP can also be observed by comparing the curves of the RSN/RLN and the RSM/RLM, as shown in Fig.4 (a) and (b). Moreover, the RGBA achieves close-to-optimal throughput when the flow rate is larger than 6 Mbps, whereas the RGBA has a much lower complexity than the optimal brute force algorithm.

Fig.5 shows the system throughput as a function of delay requirements. In the small-scale network as shown in Fig.5 (a), for the same requirement of service latency, the RSM and the OSM achieve lower throughput than the RSN and the OSN, respectively. As can be observed, the system throughput of four algorithms equals to zero at very low delay requirements due to the limited capacity of network resources. In the large-scale network as shown in Fig.5 (b), it can be observed that the system throughput of the RLN increases with the delay requirement, and is close to the upper bound by using ULN when the delay requirement reaches 4.5ms. It is demonstrated that the proposed RGBA can be used for the service with low-latency requirements in the NFPMec.

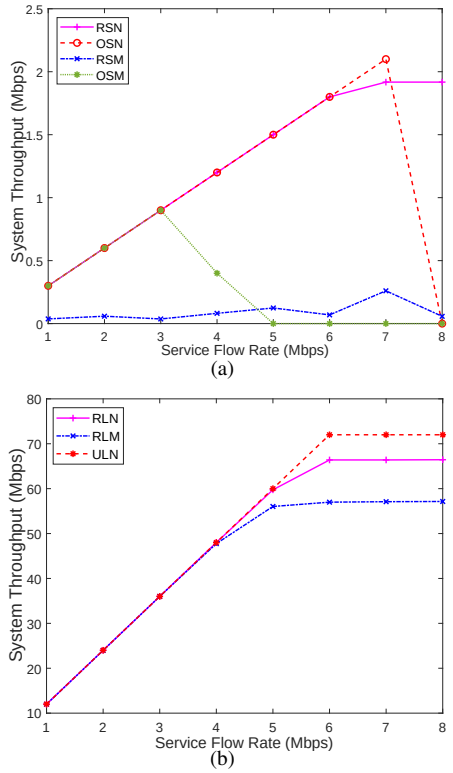


Fig. 4. Throughput when the service flow rate varies from 1 to 8 Mbps

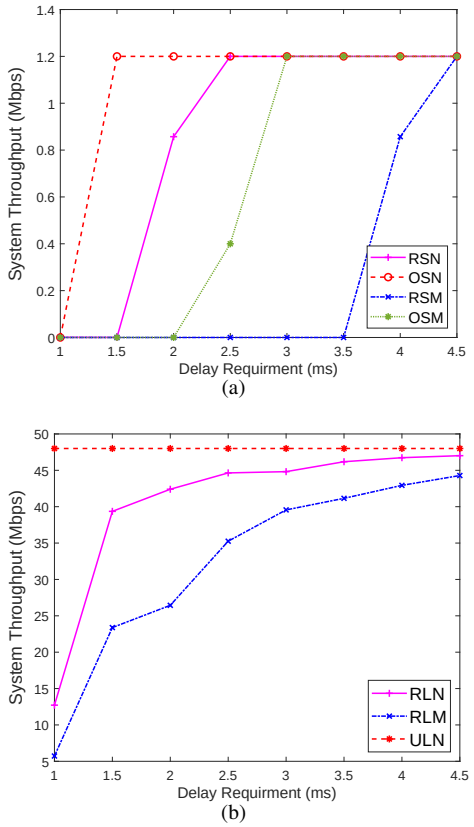


Fig. 5. Throughput when the delay requirement varies from 1 to 4.5 ms

Fig.6 shows the comparison between different components of the end-to-end packet delay obtained by using RLN and RLM respectively in the large-scale network , where the delay requirement of all services is set to 3ms. As shown in the figure, the average queueing and processing delay (Q&P delay) of both the RLN and the RLM increases with the service flow rate. The Q&P delay of RLM is twice as much as that of the RLN, which indicates that the NFP efficiently decreases the end-to-end delay of packets in the NFPMec. The average retransmission delay (Ret delay) keeps stable in both RLM and RLN, as we solve (25) to ensure the packets can be successfully received in the fixed number of retransmission trials.

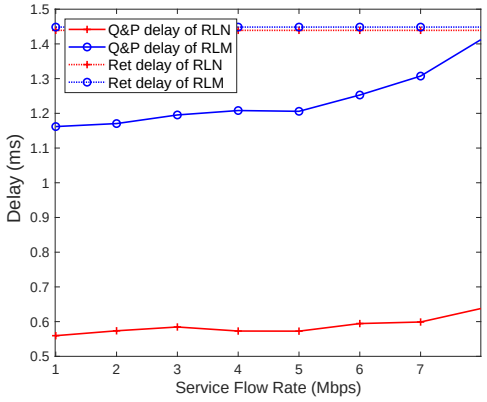


Fig. 6. End-to-end delay of one packet when the service flow rate varies from 1 to 8 Mbps

A. System Throughout and Delay

Fig.7 shows the throughput of slice as a function of the service flow rate. To demonstrate the impact of both fairness metric and delay requirement on throughput performance in the NFPMec, we consider that there are only two slices in the system with different delay requirements and fairness metrics. In system 1, slice 1 has a pair of delay requirement and fairness metric, i.e., ($D1 = 3.0$ms $a1 = 0.8$), and slice 2 has a pair of ($D2 = 2.0$ms $a2 = 0.8$). In system 2, slice 1 and slice 2 have two pairs of ($D1 = 3.0$ms $a1 = 1.0$) and ($D2 = 2.0$ms $a2 = 0.1$) respectively. In system 3, slice 1 and slice 2 have two pairs of ($D1 = 3.0$ms $a1 = 0.1$) and ($D2 = 2.0$ms $a2 = 1.0$) respectively. In Fig.7 (a), the throughput of all slices first increases with the service flow rate and then remain stable. In system 1, both two slices have the the same fairness metric ($a1 = a2 = 0.8$), and slice 2 with more rigid delay requirement ($D2 < D1$) achieves higher throughput than slice 1. In system 2, the fairness metric of slice 1 is 1.0, which is higher than the fairness metric 0.1 of slice 2, and the delay requirements of both slices are the same as that in the system 1. Comparing the system 1 and the system 2, the throughput of slice 1 is increased, whereas that of slice 2 is decreased. The similar phenomenon is presented by comparing the throughput of two slices in system 1 and system 3 as shown in Fig.7 (b). The fairness metric largely increases the flexibility for both user

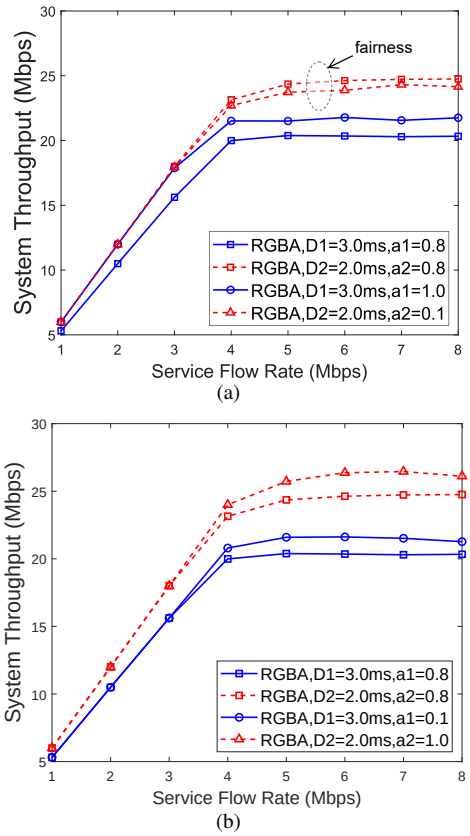


Fig. 7. Throughput when the service flow rate varies from 1 to 8 Mbps

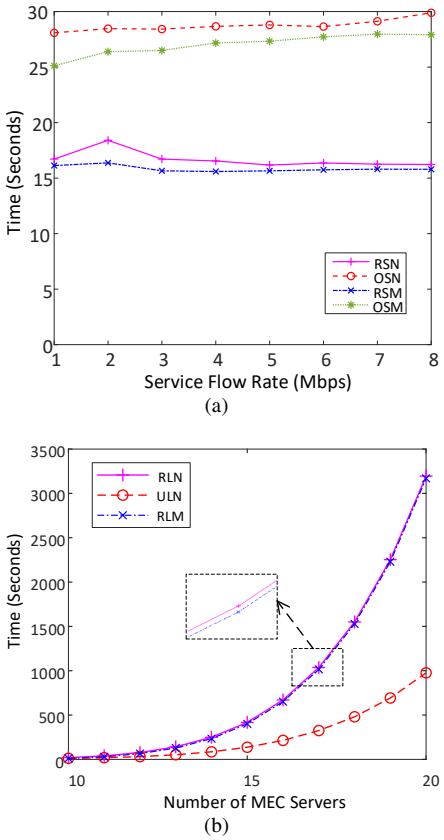


Fig. 8. Time versus the service flow rate,

and service provider to provide fairness-aware and low-latency services.

Fig.8 shows the running time of RSN, RSM, RLN, RLM and ULN to get suboptimal solutions and OSM and OSN to get optimal solutions in different cases respectively by using the simulation parameters in Table II. Fig.8.(a) show that the RGBA can obtain the optimal solution efficiently when the size of substrate network and the length of SFC is moderately (10 users and 8 servers) . Fig.8.(a)-(b) shows that the SPG construction algorithm has only trivial impact on the running time of the RGBA. In Fig.8.(b), we can see that the running time of RLN, ULN and RLM increases fast as the number of servers. This is consistent with the complexity analysis.

VII. Conclusions

In this study, we have proposed the NFPMec framework for serving low-latency services. We have formulated the FTMP to achieve a desired trade-off between system revenue and performance isolation for slices. The FTMP is solved to maximize the system throughput with bounded system resource capacity while meeting the delay requirements of users. We have proposed the RGBA to solve the FTMP by equivalently solving two sub-problems, which can be solved much easier after convex approximation. The Numerical results have proved the advantage of introducing the NFP into MEC networks, as well as the effectiveness of the proposed RGBA compared with the four benchmark algorithms. This framework can provide efficient and fairness-aware traffic management for the services with low-latency requirements in future MEC.

Appendix A
Proof of Theorem 1

We prove it by showing that the RFTMP has satisfied the necessary conditions for holding Theorem 2.2 and Theorem 2.3 in [30]. Theorem 2.2 and Theorem 2.3 are demonstrated based on non-linear duality theory, which demonstrate that transformation manipulations applied to the RFTMP yield an equivalent problem in the form of $P2$.

There are two conditions for holding Theorems 2.2 in [30]. 1) Fix y in the RFTMP, $G(x, y)$ is concave on X. We observe that $(14a)$ is concave on X since the first factor at the RHS of (12) is convex on X when fixing y and the rest constraints in $G(x, y)$ is linear. Above all, $G(x, y)$ is concave on X when fixing y. 2) $Z_y := \{z \in R^m | G(x, y) \geq z \text{ for some } x \in X\}$ is closed for each $y \in Y$. We observe that each element of $G(x, y)$ is continuous on X for each fixed y in Y and X is closed. Hence, Z_y must be closed for each $y \in Y$.

There are two conditions for holding Theorems 2.3 in [30]. 1) $v(\overline{y})$ is finite where $v(\overline{y}) := \text{supremum}_{x \in X} f(x, \overline{y})$, and $G(x, \overline{y}) \geq 0$. We observe that since X is closed, $v(\overline{y})$ is finite when fixing y to $\overline{y}$. 2) $G(x, \overline{y})$ and $f(x, \overline{y})$ are continuous on closed X, and the ϵ-optimal ($\epsilon \geq 0$) solution of the

optimization problem with fixed y named as $P*$ as shown below is non-empty.

$$P*: \max_{x\in X} f(x,y),$$
$$s.t.\ G(x,y)\geq 0.$$

The optimization problem $P*$ is indeed a flow scheduling problem with fixed power allocation. There exists an optimal solution to the flow scheduling problem if and only if the constraints $G(x,\overline{y})$ are satisfied for some $x\in X$, X is non-empty. When condition 2) is satisfied, Theorems 2.3 is satisfied. Theorems 2.3 in [30] also demonstrates that the optimal value of P4 equals that of its dual that is $v^{(d)}(y) = infimum_{u\geq 0}[supremum_{x\in X} f+u^tG(x,y)], y\in Y\cap V$. This concludes the proof.

Appendix B
Derivation of closed-form of $L^*(y;u)$ and $L_*(y;\omega)$

Based on the observation, $f(x,y)$ and $G(x,y)$ are linearly separate. Specifically,

$$f(x,y) = f_1'(x)+f_2'(y),$$
$$g_j(x,y) = g_{j,1}'(x)+g_{j,2}'(y),$$

where $f_2'(y)=0$. We denote $G_1'(x)=\{g_{j,1}'(x)|1\leq j\leq m\}$ and $G_2'(y)=\{g_{j,2}'(y)|1\leq j\leq m\}$. We thus have $L^*(y;u)$ and $L_*(y;\omega)$ in an explicit form

$$L^*(y;u) = \text{supremum}_{x\in X}\{f_1'(x)+u^tG_1'(x)\}+u^tG_2'(y),$$
$$L_*(y;\omega) = \text{supremum}_{x\in X}\{\omega^tG_1'(x)\}+\omega^tG_2'(y),$$

where $g_{j,2}'(y) = (g_{j,2}^{',(1)}(y),\cdots,g_{j,2}^{',(i)}(y),\cdots,g_{j,2}^{',(|\mathbb{I}|)}(y)), 1\leq j\leq m$ and

$$g_{1,2}^{',(i)}(y) = -\frac{1}{B(1-P_i^{out}(y))log_2(1+\frac{A_i(y)}{B_i(y)})},$$
$$g_{2,2}^{',(i)}(y) = 1-P_i^{out}(y) = e^{-\gamma_{th}\frac{B_i(y)}{A_i(y)}},$$
$$g_{j,2}^{',(i)}(y) = 0, 3\leq j\leq m.$$

Appendix C
Proof of the concavity of constraint (23a)

By relaxing $B_i(y)$ to $\tilde{B}^m = B_i(y^m)$, $g_{1,2}^{',(i)}(y)$ and $g_{2,2}^{',(i)}(y)$ are rewritten as

$$\tilde{g}_{1,2}^{',(i)}(y) = f(y)g(y),$$
$$\tilde{g}_{2,2}^{',(i)}(y) = e^{\frac{-\gamma_{th}B^m}{A_i(y)}},$$

where

$$f(y) = -B^{-1}e^{\frac{-\gamma_{th}B^m}{hy}}, h=|h_{i,a_i}|^2,$$
$$g(y) = (log(1+\frac{hy}{B^m}))^{-1}.$$

Then, the derivatives of $f(y)$ and $g(y)$ are given by

$$f'(y) = \Delta_1 y^{-2}e^{\frac{B\Delta_1}{y}}, \Delta_1 = \gamma_{th}B^m(Bh)^{-1},$$
$$g'(y) = -h(B^m+hy)^{-1}(log(1+\frac{hy}{B^m}))^{-2}.$$

The second derivatives of $f(y)$ and $g(y)$ are given by

$$f''(y) = -2\Delta_1 y^{-3}e^{\frac{B\Delta_1}{y}} - B(\Delta_1)^2y^{-4}e^{\frac{B\Delta_1}{y}},$$
$$g''(y) = h^2(B^m+hy)^{-2}(log(1+\frac{hy}{B^m}))^{-2}(2(log(1+\frac{hy}{B^m}))^{-1}+1).$$

As $(\tilde{g}_{1,2}^{',(i)}(y))'' = f''(y)g(y)+2f'(y)g'(y)+g''(y)f(y)$, we have

$$(\tilde{g}_{1,2}^{',(i)}(y))'' = -2\Delta_1y^{-3}e^{\frac{B\Delta_1}{y}}(log(1+\frac{hy}{B^m}))^{-1}$$
$$-B(\Delta_1)^2y^{-4}e^{\frac{B\Delta_1}{y}}(log(1+\frac{hy}{B^m}))^{-1} - 2\Delta_1y^{-2}e^{\frac{B\Delta_1}{y}}h(B^m$$
$$+hy)^{-1}(log(1+\frac{hy}{B^m}))^{-2} - B^{-1}e^{\frac{B\Delta_1}{y}}h^2(B^m+hy)^{-2}$$
$$*(log(1+\frac{hy}{B^m}))^{-2}(2(log(1+\frac{hy}{B^m}))^{-1}+1).$$

In the same way, $(\tilde{g}_{2,2}^{',(i)}(y))''$ can be obtained by $(\tilde{g}_{2,2}^{',(i)}(y))'' = (d/dy(e^{\frac{k}{y}}))' = (-y^{-2}ke^{\frac{k}{y}})' = ky^{-4}e^{\frac{k}{y}}(k+2y) = -B\Delta_1y^{-4}e^{\frac{-B\Delta_1}{y}}(-B\Delta_1+2y)$, where $k = -\gamma_{th}B^mh^{-1} = -B\Delta_1$. Thus, we have (25).

It can be easily observed that $\Gamma_{2,3,4} < 0$. According to fundamental inequalities $log(1+x)\geq x/(1+x)$ and $e^x\geq(x+1)$, C is a positive constant and normalize B such that $0<B<1$. Γ_1 is also negative as

$$2B^{-1}ye^{\frac{2B\Delta_1}{y}} + (2y-B\Delta_1)log(1+\frac{hy}{B^m})$$
$$> 2B^{-1}y(\frac{2B\Delta_1}{y}+1)+(2y-B\Delta_1)\frac{hy}{B^m+hy}$$
$$= \frac{C+2yB^{-1}(B^m+hy+hyB+(2\gamma_{th}-0.5\gamma_{th}B)B^m)}{B^m+hy}$$
$$> 0.$$

As the second derivative is always negative, the corresponding constraint set $\tilde{Y}$ in (24a) is concave. This ends the proof.

References

[1] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[2] "Mobile edge computing—A key technology towards 5G," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[3] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," in *6th International Confer. Mobile Computing, Applicat. and Services*, 2014, pp. 1–9.

[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 2017.

[5] "Software-defined networking: The new norm for networks," *White Paper, Open Networking Foundation*, 2012.

$$(\tilde{g}'^{,(i)}_{1,2}(y))'' + (\tilde{g}'^{,(i)}_{2,2}(y))'' = \underbrace{-B\Delta_1 y^{-4} e^{\frac{-B\Delta_1}{y}} (log(1+\frac{hy}{B^m}))^{-1} \left(2B^{-1} y e^{\frac{2B\Delta_1}{y}} + (2y - B\Delta_1) log(1+\frac{hy}{B^m})\right)}_{\Gamma_1}$$

$$\underbrace{-B(\Delta_1)^2 y^{-4} e^{\frac{B\Delta_1}{y}} (log(1+\frac{hy}{B^m}))^{-1}}_{\Gamma_2} \underbrace{-2\Delta_1 y^{-2} e^{\frac{B\Delta_1}{y}} h(B^m+hy)^{-1} (log(1+\frac{hy}{B^m}))^{-2}}_{\Gamma_3}$$

$$\underbrace{-B^{-1} e^{\frac{B\Delta_1}{y}} h^2 (B^m+hy)^{-2} (log(1+\frac{hy}{B^m}))^{-2} (2(log(1+\frac{hy}{B^m}))^{-1}+1)}_{\Gamma_4}. \quad (25)$$

[6] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[7] R. Wen, G. Feng, W. Tan, R. Ni, S. Qin, and G. Wang, "Protocol function block mapping of software defined protocol for 5g mobile networks," *IEEE Trans. Mobile Computing*, vol. 17, no. 7, pp. 1651–1665, 2018.

[8] S. Baidya, Y. Chen, and M. Levorato, "eBPF-based content and computation-aware communication for real-time edge computing," in *IEEE INFOCOM - Conf. Computer Commun. Workshops*, April, 2018, pp. 865–870.

[9] 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 Trans. Veh. Technology*, vol. 69, no. 7, pp. 7863–7878, 2020.

[10] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal vNF placement at the network edge," in *IEEE INFOCOM-Conf. Computer Commun.*, April, 2018, pp. 693–701.

[11] M. Liu, G. Feng, J. Zhou, and S. Qin, "Joint two-tier network function parallelization on multicore platform," *IEEE Trans. Network and Service Management*, vol. 16, no. 3, pp. 990–1004, 2019.

[12] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "NFP: Enabling network function parallelism in NFV," in *Pro. ACM Conf.of Special Interest Group on Data Commun.*, 2017, pp. 43–56.

[13] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommunications*, vol. 8, pp. 33–37, 1997.

[14] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. IEEE Int. Conf. Intell. Syst. Control (ISCO)*, 2016.

[15] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *IEEE International Symp. Inform. Theory (ISIT)*, July 2016, pp. 1451–1455.

[16] Z. Xu, W. Liang, M. Jia, M. Huang, and G. Mao, "Task offloading with network function requirements in a mobile edge-cloud network," *IEEE Trans. Mobile Computing*, vol. 18, no. 11, pp. 2672–2685, 2018.

[17] J. Kuo, S. Shen, H. Kang, D. Yang, M. Tsai, and W. Chen, "Service chain embedding with maximum flow in software defined network and application to the next-generation cellular network architecture," in *IEEE INFOCOM-Conf.Computer Commun*, 2017, pp. 1–9.

[18] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," *IEEE Trans. Network and Service Management*, vol. 14, no. 3, pp. 554–568, 2017.

[19] M. Dobrescu, N. Egi, K. Argyraki, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, and S. Ratnasamy, "Routebricks: Exploiting parallelism to scale software routers," in *Proc. of the ACM SIGOPS 22nd Symposium on Operating Syst. Principles*. New York, NY, USA: Association for Computing Machinery, 2009, p. 15–28. [Online]. Available: https://doi.org/10.1145/1629575.1629578

[20] "Study on management and orchestration of network slicing for next generation network (release 15)," *3GPP (2018) Technical Specification Group Services and System Aspects; Telecommunication Management; 3GPP TR 28.801 V15.1.0.*

[21] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tutorials*, vol. 22, no. 2, pp. 905–929, 2020.

[22] D. Xu, J. Wang, T. Cao, C. Yang, and B. Xia, "Performance analysis for wireless stochastic networks with dynamic traffic and packet retransmission," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2370–2380, 2020.

[23] A. Mourad, L. Brunel, A. Okazaki, and U. Salim, "Channel quality indicator estimation for ofdma systems in the downlink," in *IEEE 65th Veh. Technology Conference*, 2007, pp. 1771–1775.

[24] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .I. Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, 1997.

[25] M. O. Hasna and M. Alouini, "Outage probability of multihop transmission over Nakagami fading channels," *IEEE Commun. Lett.*, vol. 7, no. 5, pp. 216–218, 2003.

[26] C. Sun, C. She, and C. Yang, "Energy-efficient resource allocation for ultra-reliable and low-latency communications," in *IEEE Globecom 2017*, 2017.

[27] 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.

[28] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[29] D. Tabak and B. C. Kuo, *Optimal control by mathematical programming*. SRL Publishing Company, 1971.

[30] A. M. Geoffrion, "Generalized benders decomposition," *Journal of optimization theory and applications*, vol. 10, no. 4, pp. 237–260, 1972.

[31] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Processing*, vol. 66, no. 10, pp. 2616–2630, 2018.

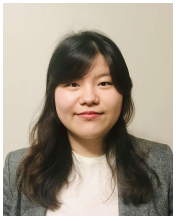
[32] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *IEEE INFOCOM 2018-Conf. Comput. Commun.*, 2018, pp. 207–215.

[33] Y. Wang, Y. Gu, and X. Tao, "Edge network slicing with statistical QoS provisioning," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1464–1467, 2019.

Mengjie Liu received the B.S. degree in Communication Engineering from the Sichuan University, in 2016. She now is pursuing her Ph.D. degree in National Key Lab of Science and Technology on Communications, University of Electronic Science and Technology of China (UESTC). She went on a one-year exchange to the BBCR lab at the University of Waterloo, Canada. She has been served as TPC member for several international conferences, such as GLOBECOM 2022, ICCT 2019. Her research interests include the network function parallelization, resource allocation in wireless/wired networks and machine learning.

Gang Feng received his BEng and MEng degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong in 1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University in December 2000 as an assistant professor and was promoted as an associate professor in October 2005. At present he is a professor with the National Laboratory of Communications, University of Electronic Science and Technology of China. Dr. Feng has extensive research experience and has published widely in computer networking and wireless networking research. His research interests include resource management in wireless networks, next generation cellular networks, etc. Dr. Feng is a senior member of IEEE.

Yao Sun received the B.S. degree in Mathematical Sciences, and the Ph.D. degree in Communication and Information System from University of Electronic Science and Technology of China (UESTC), in 2014 and 2019, respectively. Dr. Sun has published widely in wireless networking research area, and received the IEEE ComSoc TAOS Best Paper Award in 2019 ICC. He has been the guest editor for special issues of several international journals. He has been served as TPC member for number of international conferences, including GLOBECOM 2020, WCNC 2019, ICCT 2019. His research interests include intelligent wireless networking, network slicing, blockchain system, internet of things and resource management in mobile networks.



Nan Chen received her Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada in 2019, and her Bachelor degree in electrical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China. She was a post-doctoral research fellow at the University of Waterloo from 2019 to 2020. She is currently working for Tennessee Tech University, Cookeville, TN, USA. Her research interests include electric vehicle charging/discharging scheme design, resource allocation in smart grid, and machine learning application in cyber-physical systems.

Wei (andrew) Tan is a principle research scientist and architect at Central Research Institute, Huawei Technologies, Co. Ltd Shanghai, China. He received his Ph.D. in 2008 from Harbin Institute of Technology. He has published over 20 papers in refereed journals and conferences proceedings and more than 300 patents. His research interests include design of communication networks, blockchain and AI in wireless network. He is also very active in global standards for 5G including 3GPP RAN2 RAN3 SA5.