User Behavior-Aware Scheduling Based on Time-Frequency Resource Conversion

Hangguan Shan, *Member, IEEE,* Yani Zhang, Weihua Zhuang, *Fellow, IEEE,* Aiping Huang, *Senior Member, IEEE,* and Zhaoyang Zhang, *Member, IEEE*

***Abstract*—Time-frequency resource conversion (TFRC) is a recently proposed network resource allocation strategy. By exploiting user behavior, it withdraws and reutilizes spectrum resources strategically from connection(s) not being focused on by the user, to relieve network congestion effectively. In this work, we study downlink scheduling based on TFRC for an LTE-type cellular network, to maximize service delivery. The service scheduling of interest is formulated as a joint request, channel and slot allocation problem which is NP-hard. A deflation and sequential fixing based algorithm with only polynomial-time complexity is proposed to solve the problem. For practical implementation, we propose TFRC-enabled low-complexity yet online scheduling algorithms, which integrate prediction-based leaky bucket-like traffic shaping and modified Smith ratio or exponential capacity-based utility function. Furthermore, by establishing a charging model for the relationship between TFRC-enabled scheduling and its TFRC-disabled counterpart, we analytically study the benefits of integrating TFRC with scheduling. Simulation results not only verify the analysis of impact of key parameters on the performance improvement, but also corroborate the benefits of integrating TFRC with scheduling techniques in terms of quality-of-service provisioning and resource utilization.**

Keywords: Time-frequency resource conversion, context-aware resource allocation, virtual spectrum hole, scheduling, competitive ratio.

I. Introduction

With the rapid development of wireless communication technologies, more and more people and devices are seamlessly connected with each other. However, the exponentially increasing traffic volume, especially on-demand data services, from advanced user equipment units (UEs) [2], poses a significant challenge for both quality-of-service (QoS) provisioning for each user and revenue improvement for a network operator, mainly due to the scarcity of radio spectrum resources. To cope with the challenge, many enabling technologies for improving network efficiency emerge as required, e.g., small cell, network cooperation, massive multiple-input multiple-output (MIMO), full duplex, and context awareness [3,4]. Within these key technologies, context awareness has recently been attracting an increased attention due to its capability of providing more bandwidth-efficient and user-centric services [5].

In general, according to [5], "context refers to information characterizing the situation of an entity or a group of entities, and it provides information about the present status of the entities". Taking context-aware resource allocation for cellular networks as an example, context information (CI) defined first in [6,7] can be any knowledge on data transmissions collected by UE and exploited by a resource manager. It includes knowledge not only on traffic features for example data delivery deadline, application type (e.g., system applications without user interface or interactive applications), and request type (e.g., for immediate display or for caching), but also on user behaviors reflected for example in the set of active applications (i.e., the applications of focus). The knowledge of UEs' context information leads to new perspectives for resource management (called context-aware resource allocation) in wireless networks [8]–[14].

In our previous work [13], we propose a novel user behavior-aware network resource allocation strategy, time-frequency resource conversion (TFRC), to withdraw radio resources strategically from connection(s) not being focused on by the user, thus providing re-useable spectrum called "virtual spectrum hole". To implement the strategy, UEs need to periodically feed the base station (BS) with their CI on which connection is of current user focus. Whereby, the BS can withdraw and reutilize the radio resources for other connections temporally. The idea of TFRC is a new way of finding white space (from the perspective of user activity dimension), different from the conventional cognitive radio approach [15]. By integrating TFRC with call admission control, the newly proposed TFRC strategy not only reduces network congestion but also increases cell capacity effectively [13]. However, the optimal setting for TFRC-oriented call admission control suffers from the curse of dimensionality, because of Markov chain-based optimization in a high-dimensional space. To address the scalability issue of TFRC, in this work we extend the study of TFRC into the area of scheduling, thus further facilitating the implementation of the new method in a practical networking scenario.

In specific, considering downlink transmissions in an LTE-type cellular network, here we investigate TFRC-based scheduling for on-demand data service with hard deadline constraints. To satisfy the delay requirement of any request,

Manuscript received May 15, 2016; revised November 1, 2016 and February 9, 2017; accepted March 20, 2017. This work was supported in part by the National Natural Science Foundation of China under Grants 61571135 and 61201228; by the Zhejiang Provincial Public Technology Research of China under Grant 2016C31063; and by the research grant from the Natural Sciences and Engineering Research Council of Canada. The review of this paper was coordinated by Prof. Chadi Assi. *(Corresponding author: H. Shan.)*

H. Shan, A. Huang, and Z. Zhang are with the College of Information Science and Electronic Engineering and with Zhejiang Provincial Key Laboratory of Information Processing and Communication Networks, Zhejiang University, Hangzhou, 310027 China (e-mail: {hshan, aiping.huang, ning_ming}@zju.edu.cn).

Y. Zhang is with Meteorological Information and Network Center of Zhejiang Province, Hangzhou, 310017 China (e-mail: yanizhang1991@163.com).

W. Zhuang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, N2L 3G1 Canada (e-mail: wzhuang@uwaterloo.ca).

This work was presented in part at IEEE Globecom 2015 [1].

Digital Object Identifier ******

all data of the request must be transmitted from the BS to the user before a hard deadline. By exploiting the CI on which connection is of current user focus, the deadline for each request with TFRC turns to be a time-varying parameter, changing dynamically with the collected user behavior information, thus providing the scheduler more freedom to strategically allocate the limited radio resources. In particular, based on user behavior-aware scheduling, in this work we address the following three issues:

- **Q1**: What is the best performance of integrating TFRC with scheduling techniques?
- **Q2**: How to approach the aforesaid performance with only causal information?
- **Q3**: What is the performance gap between scheduling enabling and disabling TFRC with only causal information?

Firstly, **Q1** is addressed by formulating the problem of service scheduling of interest as a joint request, channel and slot allocation problem, for maximizing the total reward for completed requests. However, as the problem is NP-hard, we transform the original optimization problem (OP) by an usage-based pricing and penalty-based adjustment. Then, by exploring the structure of the transformed OP, a deflation and sequential fixing based algorithm with only polynomial-time complexity is proposed. Simulation with the proposed algorithm corroborates that scheduling with TFRC has great potential to improve not only QoS provisioning for each user (in increasing the number of delivered services) but also increase revenue for a network operator simultaneously.

Secondly, to answer **Q2** thus to facilitate the implementation of user behavior-aware scheduling, we study TFRC-enabled online and low-complexity scheduling, which does not need information of future service demand and time-varying channel capacity. In specific, using prediction-based leaky bucket-like traffic shaping, we propose two online TFRC-enabled resource allocation algorithms based on modified Smith ratio and exponential capacity-based utility functions, respectively. Simulation results show that, although their performance degrades to a certain extent as compared with the offline scheduling with a priori information, they achieve much better performance than their online TFRC-disabled counterparts.

Last but not least, to address **Q3**, we propose a charging model for the relationship between TFRC-enabled scheduling and its TFRC-disabled counterpart, based on which we derive the upper bound of performance improvement of the proposed TFRC-enabled online algorithms over their TFRC-disabled counterparts. Simulation results verify the analysis of impact of key parameters on the performance improvement.

The reminder of this paper is organized as follows. We discuss the related works in Section II. The system model is described in Section III. In Section IV, we present a uniform optimization framework for both TFRC-enabled and TFRC-disabled schedules. In Sections V and VI, both offline and online scheduling algorithms with and without TFRC are studied. In Section VII, an analytical model is presented to study the benefits of TFRC-based scheduling. Performance evaluation is provided in Section VIII. Finally, conclusions are given in Section IX.

II. Related Work

Resource allocation in 4G networks has been studied extensively in the literature (see [16,17] and references therein). As compared with context-aware resource allocation, to improve throughput, maintain user fairness, and provide QoS provisioning, previous context-unaware resource allocation techniques usually exploit only information on users' causal channel and/or queue state information. Yet, context-aware resource allocation uses more degrees of freedom from more diverse types of new context information collected from UEs to improve not only resource utilization but also user's QoS or quality of experience (QoE). In below, we provide a brief review on recent progress on context-aware resource allocation especially in cellular networks, and highlight the context information utilized respectively in each of those works.

To copy with the unprecedented traffic growth generated from advanced user equipments, context-aware resource allocation is introduced in [6,7], where a *rate prediction-based serving time-aware* scheduler is proposed to facilitate efficient resource allocation among traffic flows. To provide green media delivery, Abou-zeid and Hassanein investigate the opportunities of exploiting *location awareness* from the perspectives of adaptive video quality planning, in-network caching, content prefetching, and long-term radio resource management [8]. Specifically, considering a multiuser multicell network, how *predicted user locations for video application* can minimize the required transmission airtime or total downlink BS power consumption is studied in [9,10]. By jointly utilizing information on *user location, channel quality, and network traffic load distribution*, Schoenen and Yanikomeroglu propose a new context-aware approach named user-in-the-loop (UIL), which treats a wireless network as a closed loop with the user as the system to control the spatial or temporal network traffic load distribution [11]. In [12], *social context unearthing similarities between users' interests, activities, and their interactions* is inferred from the social network profiles of the users and exploited to facilitate context-aware resource allocation for small cell networks with device-to-device communications. Proposed in [13] is a user behavior-aware network resource allocation strategy named time-frequency resource conversion, which explores the benefits of letting a resource manager be aware of *which connection(s) is currently focused on by the user*. Thus, re-useable spectrum withdrawn from those connections no being focused, so called "virtual spectrum hole", can be used to serve more new users or improve QoE of existing users. Furthermore, in recent work [14] on Internet service provisioning, with similar context information, users are classified into two states, user communicating state and user inactive state, based on which the proposed QoE-aware resource distribution framework can help an Internet resource owner differentiate really resource-starved clients from ordinary resource consumers.

Yet, to the best of our knowledge, very few such attempts except for [13,14] have been made in context-aware resource allocation techniques based on context information about user focus. Moreover, there is neither such user behavior information based scheduling techniques tailored for wireless

networks nor analytical framework with evaluation result available to unearth its impact on QoS provisioning for each user and revenue improvement for a network operator. The motivation behind this work is thus to explore the potential approach for context-aware scheduling and to evaluate the performance improvement of such user behavior-centric scheduling.

III. System Model

We consider the downlink transmission in an LTE-type cellular network, which employs orthogonal frequency division multiple access (OFDMA). The radio resource in the cell is partitioned into N orthogonal subchannels and the transmission is time slotted. In each time slot, the BS allocates the subcahnnels to the requests from all M users in the cell for a specific scheduling goal. Different from many existing scheduling works [16,17], we assume that, because of using the more and more advanced UE, any user can initiate multiple applications simultaneously. Therefore, each user can have multiple connections requiring data transmission at the same time [18]. Besides, we consider that the total scheduling time of interest consists of I successive slots each with a fixed duration T_F[1]. In each slot, each subchannel can be allocated for only one specific request and, by using adaptive modulation and coding, the maximum transmission capacity for the k^{th} request in slot i on subchannel n is $C_{i,n,k}$.

A. Request Model without TFRC

The request arrival process of each user in the cell is considered to be Poisson[2]. Denote the total requests arrived during the scheduling time as set $\mathbb{S} = \{s_1, s_2, ..., s_K\}$. Here, any request s_k can be described by a 6-tuple $(G_k, D_k, Q_k, \omega_k, \{\pi_{k,l}\}_{l=1}^{\eta}, \Pi_k)$, where G_k, D_k, and Q_k are the arrival time, the deadline, and the data amount of request s_k, respectively, ω_k denotes the reward (i.e., money) that system can obtain by finishing data transmission for s_k[3], $\pi_{k,l}$ represents the time duration during which the user switches his focus for the l^{th} time from s_k, with $l = \eta$ denoting the last focus switch, and Π_k is the total accumulated time that the user will focus on request s_k. The value of Π_k can be estimated according to the request's data amount and its usage history. For the time slotted model, a request arriving during a slot is equivalent to one arriving at the beginning of the next slot. Similarly, its deadline during one slot can be equivalently replaced by the beginning instant of that slot. Therefore, G_k and D_k can be rounded to two integers, i.e., $G_k \in \mathbb{Z}^+$, $D_k \in \mathbb{Z}^+$, $1 \le G_k \le D_k \le I$. When the BS has no information on which application is of user focus during a slot (i.e., scheduling without applying TFRC), the deadline of s_k for schedule should simply be

$$D_k = G_k + \Pi_k. \tag{1}$$

B. TFRC-Oriented Request Model

1) Review of User Behavior-Driven TFRC: The key idea of TFRC is to allocate radio resources mainly to the connection that a user focuses on [13]. Assume that a user opens multiple connections simultaneously, but he/she focuses on one or some of them at a specific time. By utilizing the user behavior information, the BS withdraws radio resources temporally and strategically from connections not being focused on by their users, providing reusable radio resource "virtual spectrum hole". By doing that, we can either allocate enough radio resources to the requests of user focus to improve user QoE or accommodate more users in the cell to increase cell capacity.

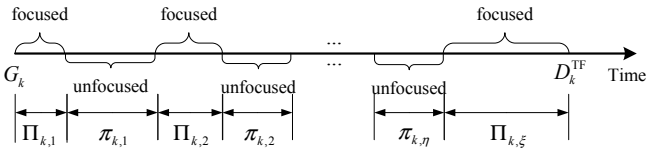


Fig. 1. Time-varying extension of request's deadline in applying TFRC.

2) TFRC-Oriented Offline Request Model: When implementing the TFRC strategy, UEs collect CI on which application is of current user focus, and feed it back to the BS every $\tau = qT_F$ seconds, where $q \in \mathbb{Z}^+$. Then, the BS can estimate the urgency of each request more accurately than that without TFRC, thus allocating its resources more efficiently. Compared with the request model without TFRC, the deadline of each request with TFRC may extend with every feedback of CI. Fig. 1 illustrates such extension of request s_k's deadline when applying TFRC, where $\Pi_{k,h}$ is the observed length of the h^{th} interval during which the user focuses on s_k, with $h = 1, 2, ..., \xi$. The time interval between two neighboring $\Pi_{k,h}$'s is the duration $\pi_{k,l}$ that the user focuses on other requests. As the total time that a user focuses on one request should not depend on whether or not the TFRC strategy is employed, we have $\sum_{h=1}^{\xi} \Pi_{k,h} = \Pi_k$. If there exist η intervals in total during which the user switches his focus from s_k, the final deadline of the request as shown in Fig. 1 is given by

$$D_k^{\mathrm{TF}} = G_k + \Pi_k + \sum\nolimits_{l=1}^{\eta} \pi_{k,l}. \tag{2}$$

By TFRC-oriented offline request model, we mean that the BS knows D_k^{TF} for any request s_k in $\mathbb{S}$ a priori at the beginning of the whole scheduling duration. Based on this model, we aim to study the upper bound performance of TFRC-based scheduling techniques.

3) TFRC-Oriented Online Request Model: In practice, the BS cannot forecast a request's final deadline, but to estimate the deadline online in every round of CI feedback as follows. Denote $D_k^{\mathrm{TF}}(i)$ as the estimated deadline of request s_k at slot i. When initiating the request, $i = G_k$, there is no CI collected yet, so $D_k^{\mathrm{TF}}(i) = D_k$. For other slots in its lifetime, we differentiate the following two cases. If the UE indicates that the user is not focusing on request s_k, the new deadline

[1]In offline scheduling with TFRC, the scheduling duration consists of multiple time slots, in which the channel quality and service demand are assumed to be known a priori. In online scheduling to be studied, the scheduling period is one time slot, corresponding to the subframe in LTE systems.

[2]The service for a request is simply referred to as request in this paper.

[3]Defining an accurate or practical reward model is a key and challenging issue in service pricing for mobile networks [19]. It depends on what charging scheme a network operator will use [20]. Yet, in this work we assume the reward of any request is preset and thus do not impose any constraint on the reward model.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS.

Symbol	Definition
N (M)	Subchannel (user) number
T_F	Slot duration
I	Slot number in the scheduling duration
s_k	The k^{th} request during scheduling duration
$C_{i,n,k}$ $(\hat{C}_{i,n,k})$	Transmission capacity (estimated transmission capacity) for s_k in slot i on subchannel n
G_k	Arrival time of s_k
D_k	Deadline of s_k with schedule disabling TFRC
D_k^{TF}	Deadline of s_k with offline schedule enabling TFRC
$D_k^{\mathrm{TF}}(i)$	Deadline of s_k estimated at slot i with online schedule enabling TFRC
Q_k	Data amount of s_k
$Q_k^{(r)}$	Remaining data amount of s_k with offline schedule after invoking sequential fixing algorithm or that with online schedule before current slot
$\tilde{Q}_k^{(r)}$	Remaining data amount of s_k with online schedule after current slot
ω_k	Reward that system obtains by finishing serving s_k before deadline
$\Pi_{k,h}$ $(\pi_{k,l})$	Length of the h^{th} (l^{th}) interval during which the user does (does not) focus on s_k
Π_k	Total accumulated time that the user focuses on s_k
τ	Context information's feedback period
$x_{i,n,k}$	Allocation variable indicating whether channel n is allocated to s_k in slot i
$\mathcal{X}$ $(\mathcal{X}^{\mathrm{TF}})$	A feasible scheduling scheme without (with) TFRC
$\mathbb{S}_i$ $(\mathbb{S}_i^{\mathrm{TF}})$	Feasible scheduling set in slot i without (with) TFRC
$\psi_{\mathcal{X},k}$ $(\psi_{\mathcal{X}^{\mathrm{TF}},k})$	Delivery indicator of s_k without (with) TFRC
A_k	Per-unit reward of s_k
Δ	Set of requests constrained by link rate prediction-based leaky bucket-like traffic shaping strategy
U_k	Utility of s_k in current slot
p_k	Remaining transmission slots of s_k given single channel with time-invariant channel capacity
$p_{\max}$	Maximum request transmission time
β_k	Critical time of s_k
$c_k(t)$	Deadline extension for s_k at time t
$c_{\max}$	Maximum request deadline extension
$\pi(k,a)$	Utility of serving s_k when it needs a slots to finish data transmission

estimate can be updated by $D_k^{\mathrm{TF}}(i) = D_k^{\mathrm{TF}}(i-1) + \tau$, where τ is the feedback period. As τ can be much smaller (e.g., several milliseconds) than a human's normal attention span (lasting for at least several seconds [21]–[23]), we assume that the user will remain unfocused on s_k at least until the next feedback. Otherwise, if the user is focusing on s_k or no information is sent to the BS, a conservative strategy is to maintain the same deadline with $D_k^{\mathrm{TF}}(i) = D_k^{\mathrm{TF}}(i-1)$. To sum up, when applying TFRC, the BS can schedule request and allocate resource according to the following estimation of each request's time-varying deadline:

$$D_k^{\mathrm{TF}}(i) = \begin{cases} D_k, & i = G_k \\ D_k^{\mathrm{TF}}(i-1) + \tau, & i \in \left(G_k, D_k^{\mathrm{TF}}(i-1)\right) \ \& \\ & \text{not focus on } s_k \\ D_k^{\mathrm{TF}}(i-1), & i \in \left(G_k, D_k^{\mathrm{TF}}(i-1)\right) \ \& \\ & (\text{focus on } s_k \text{ or no CI}). \end{cases} \tag{3}$$

As many symbols are used in this paper, we summarize the important ones in Table I.

IV. Problem Formulation

In this section, we study how to integrate TFRC in the transmission schedule. The objective of service provisioning here either with or without TFRC is to maximize the total reward of delivered services over the scheduling duration of interest. In the following, to obtain an understanding of the benefits of integrating TFRC into scheduling techniques, we provide a uniform optimization framework for both TRFC-enabled and TFRC-disabled schedule, assuming that the scheduler has the complete information of future request and time-varying channel capacity.

Let $x_{i,n,k}$ denote an allocation variable which equals 1 if channel n is allocated to request s_k in slot i and 0 otherwise. Then, a feasible scheduling scheme without TFRC and that with TFRC, both over the whole scheduling duration, can be denoted as the following two allocation sets, respectively,

$$\mathcal{X} = \{x_{i,n,k} | n = 1, 2, ..., N, G_k \le i \le D_k, \forall s_k \in \mathbb{S}\}$$
$$\mathcal{X}^{\mathrm{TF}} = \left\{x_{i,n,k} | n = 1, 2, ..., N, G_k \le i \le D_k^{\mathrm{TF}}, \forall s_k \in \mathbb{S}\right\}.$$

The key difference between the two feasible scheduling schemes lies in that, with the collected user behavior information, the real deadline of any request (say request s_k) and the feasible scheduling set in any slot (say slot i) extend, respectively, from D_k to D_k^{TF} and from $\mathbb{S}_i = \{s_k | G_k \le i \le D_k\}$ to $\mathbb{S}_i^{\mathrm{TF}} = \left\{s_k | G_k \le i \le D_k^{\mathrm{TF}}\right\}$. For a specific $\mathcal{X}$ $(\mathcal{X}^{\mathrm{TF}})$, define $\psi_{\mathcal{X},k}$ $(\psi_{\mathcal{X}^{\mathrm{TF}},k})$ as a delivery indicator of request s_k, which equals 1 if request s_k is served before its deadline D_k (D_k^{TF}) and 0 otherwise. Based on channel allocation results, $\psi_{\mathcal{X},k} = 1$ $(\psi_{\mathcal{X}^{\mathrm{TF}},k} = 1)$ if $\sum_{i=G_k}^{D_k} \sum_{n=1}^{N} x_{i,n,k} \cdot C_{i,n,k} \ge Q_k$ $(\sum_{i=G_k}^{D_k^{\mathrm{TF}}} \sum_{n=1}^{N} x_{i,n,k} \cdot C_{i,n,k} \ge Q_k)$, and $\psi_{\mathcal{X},k} = 0$ $(\psi_{\mathcal{X}^{\mathrm{TF}},k} = 0)$ otherwise. The optimal service scheduling of interest with or without TFRC can be formulated as a joint request, channel and slot allocation problem:

$$\text{(OP1)} \quad \max_{\Lambda} \quad \sum\nolimits_{s_k \in \mathbb{S}} \omega_k \psi_{\Lambda,k} \tag{4a}$$
$$\text{s.t.} \quad x_{i,n,k} \in \{0,1\}, \ \forall i, \ \forall n, \ \forall s_k \in \Phi_i \tag{4b}$$
$$x_{i,n,k} = 0, \ \forall i, \ \forall n, \ \forall s_k \notin \Phi_i \tag{4c}$$
$$\sum\nolimits_{s_k \in \Phi_i} x_{i,n,k} \le 1, \ \forall i, \ \forall n \tag{4d}$$

where $\Phi_i = \mathbb{S}_i$ and $\Lambda = \mathcal{X}$ for scheduling without TFRC, $\Phi_i = \mathbb{S}_i^{\mathrm{TF}}$ and $\Lambda = \mathcal{X}^{\mathrm{TF}}$ for scheduling with TFRC. In specific, the objective function (4a) accumulates the reward from all requests delivered data before deadline[4]. Constraints (4b) and (4c) imply that each request can only be scheduled in its lifetime. Constraint (4d) means that in any slot each subchannel can only be allocated to no more than one request.

The optimal schedules with and without TFRC have the same mathematical structure. Yet, recall that for any slot as TFRC increases the feasible scheduling set, a TFRC-enabled scheduler has a larger space to search for a better scheduling solution, thus can work better. The increase of feasible set depends on two factors, namely the prevalence level of advanced UEs and the usage behavior with the advanced UEs; so, the performance improvement by TFRC hinges upon them as well. However, to find the performance gap accurately, the time complexity of the formulated OP is an issue.

[4]In scheduling theory, the objective of the studied scheduling problem falls into a category of minimizing weighted number of tardy jobs [24], with which the BS not only tries to gain more revenue but also simultaneously satisfies more users by finishing delivering their data.

Proposition 1: OP1 is an NP-hard problem.

Proof: We prove the NP-hardness by reducing an NP-hard single-machine preemptive scheduling problem to the OP. Consider the special case of OP1 for: 1) a single channel scenario, $N = 1$, and denoting $x_{i,n,k} = x_{i,k}$, $C_{i,n,k} = C_{i,k}$; and 2) a homogeneous channel scenario with equal link rate of any requests over any slots, and letting $C_{i,k} = C$. Then, the total transmission time of request s_k, denoted as p_k, equals $\lceil Q_k/C \rceil$ slots. Taking TFRC-enabled schedule as an example, the feasible allocation scheme can be reformed as $\mathcal{X}^{\mathrm{TF}} = \left\{x_{i,k} | G_k \leq i \leq D_k^{\mathrm{TF}}, \forall s_k \in \mathbb{S}\right\}$. Accordingly, the OP can be transformed into a *time-based* formulation:

$$\text{(OP2)} \max_{\mathcal{X}^{\mathrm{TF}}} \quad \sum\nolimits_{s_k \in S} \omega_k \phi_{\mathcal{X}^{\mathrm{TF}},k} \tag{5a}$$

$$\text{s.t.} \quad x_{i,k} \in \{0,1\},\ \forall i,\ \forall s_k \in \mathbb{S}_i^{\mathrm{TF}} \tag{5b}$$

$$x_{i,k} = 0,\ \forall i,\ \forall s_k \notin \mathbb{S}_i^{\mathrm{TF}} \tag{5c}$$

$$\sum\nolimits_{s_k \in \mathbb{S}_i^{\mathrm{TF}}} x_{i,k} \leq 1,\ \forall i \tag{5d}$$

where $\phi_{\mathcal{X}^{\mathrm{TF}},k}$ is a new *timeslot-based* delivery indicator which equals 1 if slots allocated to s_k satisfy $\sum_{i=G_k}^{D_k^{\mathrm{TF}}} x_{i,k} = p_k$ and 0 if $\sum_{i=G_k}^{D_k^{\mathrm{TF}}} x_{i,k} < p_k$. In scheduling theory, OP2 is equivalent to a single-machine preemptive scheduling problem, for minimizing the sum of the weights of the late jobs, with integer request times (G_k), processing times (p_k), and deadlines (D_k^{TF}) (formal notation: $1|preemption, G_k| \sum \omega_k(1 - \phi_{\mathcal{X}^{\mathrm{TF}},k})$), which is NP-hard [24,25]. Therefore, according to the property of reducibility, OP1 is also NP-hard. ∎

Due to the high complexity, in the next section, we focus on designing an efficient algorithm to solve OP1. The solution helps to measure the performance gap between TFRC-enabled scheduler and its TFRC-disabled counterpart, and is used as a benchmark to evaluate the performance of user behavior-aware online scheduling algorithms in Section VI.

V. Offline Algorithm Design

A. Usage-based Pricing and Penalty-based Adjustment

To solve OP1, we utilize usage-based pricing and penalty-based adjustment to transform the original optimization problem. With usage-based pricing [26,27], the BS gains a reward for the consumed radio resources in delivering every bit of a request. As such, the total reward of the BS from serving requests in $\mathbb{S}$ is

$$J_1 = \sum\nolimits_{k=1}^{K} \sum\nolimits_{i=1}^{I} \sum\nolimits_{n=1}^{N} A_k C_{i,n,k} x_{i,n,k} \tag{6}$$

where $A_k = \omega_k / Q_k$ represents the per-unit reward from s_k. If we simply maximize J_1, users in the system suffer from increased risk that the served request may not be finished before deadline, as the BS tends to serve requests generating a large per-unit reward.

To not only improve system reward but also increase user satisfaction, we use a penalty-based approach to adjust the scheduling objective J_1. Intuitively, for any request at its deadline, the more the data is pending for transmission, the less the user satisfaction is [28], and thus the more the refund (denoted as $p_1(k)$) should be, i.e.,

$$p_1(k) = \alpha A_k [Q_k - \sum\nolimits_{i=1}^{I} \sum\nolimits_{n=1}^{N} C_{i,n,k} x_{i,n,k}]^+ \tag{7}$$

where $[x]^+ = \max\{x, 0\}$ and α is a non-negative value to weight the tradeoff between system reward J_1 and user satisfaction. Yet, the penalty in this form does not improve user satisfaction efficiently, as stated in the following proposition.

Proposition 2: Scheduling by maximizing $J_1 - \sum_{k=1}^{K} p_1(k)$ has performance close to that of maximizing usage-based system reward J_1. The difference between them reduces as requests' data size increases or per-slot link capacity reduces, and the mean of the gap reduces to zero if all requests have equal average link capacity and time-frequency resources are divided into the granularity of unit bit per channel allocation.

We prove Proposition 2 in Appendix A. The performance of scheduling by maximizing $J_1 - \sum_{k=1}^{K} p_1(k)$ will be studied in Section VIII. An alternative penalty is to let the refund (denoted as $p_2(k)$) for the service of unfinished data transmission increases with the data transmission time (i.e., the time a user has spent on the unfinished service) and the amount of the delivered data,

$$p_2(k) = \alpha A_k \sum\nolimits_{i=1}^{I} \sum\nolimits_{n=1}^{N} C_{i,n,k} x_{i,n,k} \tag{8}$$

when $\sum_{i=1}^{I} \sum_{n=1}^{N} C_{i,n,k} x_{i,n,k} < Q_k$. The rationality of the new penalty can be understood if we define the following penalty-based reward function for request schedule

$$J_2 = J_1 - \sum\nolimits_{k=1}^{K} (1 - \psi_{\Lambda,k})\, p_2(k). \tag{9}$$

At $\alpha = 1$, (9) is reduced to $J_2 = \sum_{s_k \in \mathbb{S}} \psi_{\Lambda,k} \sum_{i=1}^{I} \sum_{n=1}^{N} A_k C_{i,n,k} x_{i,n,k}$. That is, the BS still only collects reward but with usage-based pricing from completed requests, and OP1 can thus be transformed into:

$$\text{(OP3)} \max_{\Lambda} \quad \sum_{s_k \in \mathbb{S}} \psi_{\Lambda,k} \sum_{i=1}^{I} \sum_{n=1}^{N} A_k C_{i,n,k} x_{i,n,k} \tag{10a}$$

$$\text{s.t.} \quad \text{(4b)},\ \text{(4c)},\ \text{(4d)} \tag{10b}$$

$$\sum_{i=1}^{I} \sum_{n=1}^{N} C_{i,n,k} x_{i,n,k} \leqslant Q_k, \forall s_k \in \Phi_i \tag{10c}$$

Here, we add constraint (10c) in OP3 to limit channel allocation, therefore the BS cannot claim a reward larger than $\omega_k = Q_k A_k$ from s_k. However, this transformation in general is not equivalent to OP1, unless all requests can fully utilize the link capacity of their allocated channels. By checking the Hessian matrix of the relaxed function of (10a), we know that OP3 is a non-convex non-linear integer OP, which is still difficult to address. By exploiting the similar structure between the objective function of OP3 and the usage-based system reward J_1, next we propose a deflation and sequential fixing based algorithm to solve it efficiently.

B. Deflation and Sequential Fixing based Revenue Boosting Algorithm

The principle of deflation is straightforward [29]: At first, all requests are scheduled together in allocating resources to maximize usage-based system reward J_1; the request, which has not completed data transmission but increases J_2 the most if reallocating all of its radio resources to other incomplete ones, is sequentially dropped; thereafter, resource allocation

among incomplete requests repeats until a feasible solution finds all the existing requests delivered before deadline or no increase of J_2 occurred via dropping any incomplete request. So to apply the deflation approach to OP3, we first solve the following 0-1 integer linear programming (LP) problem

$$\begin{aligned}(\text{OP4}) \quad \max_{\mathcal{X}} \quad & J_1 && (11a)\\ \text{s.t.} \quad & (4b),\ (4c),\ (4d) && (11b)\\ & \sum_{i=1}^{I}\sum_{n=1}^{N} C_{i,n,k}x_{i,n,k} \leqslant Q_k, \forall s_k \in \mathbb{S}_f && (11c)\end{aligned}$$

Here, we take scheduling without TFRC as an example, but the algorithms presented in below can be applied to scheduling with TFRC as well. In OP4, the objective is to maximize the total usage-based system reward, and $\mathbb{S}_f$, which initially equals $\mathbb{S}$, is the set of requests to be scheduled in iteration. The size of $\mathbb{S}_f$ decreases with the operation of the deflation approach. As 0-1 integer LP problem is NP-hard in general [30], we require an efficient solver to address OP4. Our main idea to solve OP4 is to fix the values of the $x_{i,n,k}$ variables sequentially through solving a number of relaxed LP problems, with each iteration setting at least one binary value for some $x_{i,n,k}$. Specifically, during the first iteration, we relax all binary variables $x_{i,n,k}$ to continuous ones $0 \leq x_{i,n,k} \leq 1$. Such a relaxation leads to the following upper-bounding problem formulation

$$\begin{aligned}(\text{OP5}) \quad \max_{\mathcal{X}} \quad & J_1 && (12a)\\ \text{s.t.} \quad & (4c),\ (4d),\ (11c) && (12b)\\ & x_{i,n,k} \in [0,1], \forall i, \forall n, \forall s_k \in \mathbb{S}_f && (12c)\end{aligned}$$

which is a standard LP and can be solved in polynomial time. Upon solving this LP, we have a solution with each $x_{i,n,k}$ between 0 and 1. Among all the variables, we set the one with the largest value (say x_{i^*,n^*,k^*}) to 1. As a result of this fixing, by (4d), we can also fix $x_{i^*,n^*,q} = 0$ for $s_q \in \mathbb{S}_{i^*}$ and $q \neq k^*$. Further, by checking (11c), we can remove s_{k^*} from $\mathbb{S}_f$ and set its remaining unfixed variables to 0, if its data transmission ends. Then, all the terms in the LP involving these fixed variables can be removed and a new LP can be formulated. In the second iteration, we solve the new LP and then fix some additional variables based on the same process. The algorithm ends until all variables are fixed. We summarize the proposed sequential fixing (SF) algorithm in Algorithm 1.

Algorithm 1: Sequential fixing algorithm

Input: Total feasible request set $\mathbb{S}_f$, schedulable request set in slot i $\{\mathbb{S}_i\}_{i=1}^{I}$, channel number N, request number K
Output: $\{x_{i,n,k}, \forall i, \forall n, \forall s_k \in \mathbb{S}_f\}$
1. Formulate OP5 with $0 \leq x_{i,n,k} \leq 1$;
2. Solve OP5 and derive the solution set $\mathcal{X}'$;
3. Choose the largest x_{i^*,n^*,k^*} in $\mathcal{X}'$ and set $x_{i^*,n^*,k^*} = 1$;
4. Set $x_{i^*,n^*,q} = 0$ for any $q \neq k^*$;
5. If $\sum_{i=1}^{I}\sum_{n=1}^{N} x_{i,n,k^*}C_{i,n,k^*} \geqslant Q_{k^*}$, $\mathbb{S}_f = \mathbb{S}_f - \{s_{k^*}\}$ and set its unfixed variables $x_{i,n,k^*} = 0$; otherwise, go to step 6;
6. If all the $x_{i,n,k}$ variables are fixed, output scheduling result and end algorithm; otherwise, go to step 7;
7. Formulate the updated LP problem and find the new optimal variable set $\mathcal{X}'$, then go to step 3.

Algorithm 2: Deflation & sequential fixing based revenue boosting algorithm

Input: Overall request set $\mathbb{S}$, schedulable request set in slot i $\{\mathbb{S}_i\}_{i=1}^{I}$, total channel number N, total request number K
Output: $\{x_{i,n,k}, \forall i, \forall n, \forall s_k \in \mathbb{S}\}$
1. Initialize: The feasible request set $\mathbb{S}_f = \mathbb{S}$, the dropped request set $\mathbb{S}_p = \emptyset$, and the resource fragment set $\mathbb{R} = \emptyset$;
2. Solve OP4 by the SF algorithm with input $(\mathbb{S}_f, \{\mathbb{S}_i\}_{i=1}^{I}, N, K)$, and derive the solution set $\mathcal{X} = \{x_{i,n,k}, \forall n, \forall i, \forall s_k \in \mathbb{S}\}$;
3. If all requests are completed or only a single incomplete request exists (i.e., $|\mathbb{S}_f| = 0$ or 1 after step 2), end algorithm and output $\mathcal{X}$; otherwise go to step 4;
4. For each incomplete request $s_v \in \mathbb{S}_f$, find set $\mathbb{S}_c(s_v)$ in which element $s^*_{v_j}$ is sequentially selected according to (13);
5. Find request $s_{k_1} = \arg\max_{s_v \in \mathbb{S}_f} \{\Delta J_2(s_v)\}$ based on (14);
6. If s_{k_1} has a positive value of $\Delta J_2(s_{k_1})$, update $\mathbb{S}_f = \mathbb{S}_f - \{s_{k_1}\} - \mathbb{S}_c(s_{k_1})$, $\mathbb{S}_p = \mathbb{S}_p \bigcup \{s_{k_1}\}$, $\mathcal{X}$ by allocating the resources of s_{k_1} or those from $\mathbb{R}$ to the requests one-by-one in $\mathbb{S}_c(s_{k_1})$, and $\mathbb{R}$ by deleting the newly allocated resources and adding resource fragments that are withdrew but without reallocation, then go to step 4; otherwise go to step 7;
7. If $\mathbb{R} = \emptyset$ or $\mathbb{S}_p = \emptyset$, end algorithm and output $\mathcal{X}$; otherwise, add new resource fragments to $\mathbb{R}$ by withdrawing all the resources of requests in $\mathbb{S}_f$ and thus update $\mathcal{X}$ by letting $x_{i,n,k} = 0, \forall s_k \in \mathbb{S}_f$, then go to step 8;
8. Find $s_{k_2} = \arg\max_{s_k \in \mathbb{S}_p} \left\{ A_k Q_k \left| \sum_{(i,n)\in\mathbb{R}}^{i\in[G_k,D_k]} C_{i,n,k} \geq Q_k \right. \right\}$;
9. If s_{k_2} is empty, end algorithm and output $\mathcal{X}$; otherwise, update $\mathcal{X}$ by sorting the time-frequency resources in $\mathbb{R}$ in decreasing order of C_{i,n,k_2} and allocating them to s_{k_2} sequentially as long as $\sum_{(i,n)\in\mathbb{R}}^{i\in[G_k,D_k]} C_{i,n,k_2}x_{i,n,k_2} < Q_{k_2}$;
10. Update $\mathbb{R} = \mathbb{R} - \{(i,n)\,|x_{i,n,k_2} = 1\}$ and $\mathbb{S}_p = \mathbb{S}_p - \{s_{k_2}\}$, then go to step 8.

Based on Algorithm 1, we propose a deflation and sequential fixing based revenue boosting (DSFRB) algorithm to solve the original OP, as detailed in Algorithm 2, which can be separated into the following three parts.

First, in steps 1 to 3, we invoke the SF algorithm to derive the initial allocation set $\mathcal{X}$ to maximize the total usage-based system reward J_1. If all requests are completed or only a single incomplete request exists after invoking the SF algorithm, in general we have obtained a "best-effort" solution.

Second, in steps 4 to 6, if multiple incomplete requests exist after initial resource allocation, we continue to increase J_2 by sequentially looking for a single incomplete request which, if reallocating all of its radio resources to other incomplete requests, increases J_2 the most. To measure the benefit of reutilizing the resources of any incomplete request, say s_v, we propose the following strategy to reallocate these resources (i.e., setp 4). Let $Q_k^{(r)}$ denote the remaining data amount of request s_k after invoking the SF algorithm. We select requests (denoted as $s^*_{v_j}$'s) one by one from $\mathbb{S}_f$ to share the resources of s_v, according to the strategy given in (13), where $\mathbb{R} = \left\{ (i,n) \left| \sum_{i\in[G_k,D_k], s_k\in\mathbb{S}} x_{i,n,k} = 0 \right. \right\}$ is the set of resource fragments that are withdrawn from incomplete requests but without reallocation, $\mathbb{S}_{v,j} = \{s^*_{v_1}, s^*_{v_2}, ..., s^*_{v_{j-1}}\}$ is the set consisting of requests that can finish data transmission by applying resources to s_v, with $\mathbb{S}_{v,1} = \emptyset$. The strategy ensures that only requests with chances to finish data delivery can share

$$s^*_{v_j} = \arg \max_{s_{v_j} \in \mathbb{S}_f - \mathbb{S}_{v,j}} \left\{ Q_{v_j} A_{v_j} \left| \begin{array}{l} x_{i,n,v} = 1 \ \text{ or } \ (i,n) \in \mathbb{R} \\ x_{i,n,v_t} = 0, 1 \le t \le j-1 \\ \sum_{i \in [G_{v_j}, D_{v_j}]} C_{i,n,v_j} \ge Q^{(r)}_{v_j} \end{array} \right. \right\} \tag{13}$$

the resources. Further, the larger the reward such a request can contribute, the higher the priority it will be allocated resources. Here, to exploit multiuser diversity, when allocating channels to such a request, we can always assign the channel of the best quality from all available ones to the request. However, recalling that we can choose any incomplete request from $\mathbb{S}_f$ to fulfill resource reallocation, we propose another strategy to determine the request whose resources should be withdrawn first and when to end the resource reallocation (i.e., steps 5 and 6). To this end, we represent the total benefits of reallocating radio resources of an incomplete request, say s_v, by

$$\Delta J_2(s_v) = \sum\nolimits_{s^*_{v_j} \in \mathbb{S}_c(s_v)} Q_{v_j} A_{v_j} \tag{14}$$

where $\mathbb{S}_c(s_v)$ is the final set composed of all incomplete requests sequentially found based on (13). Then, the one with the largest contribution (denoted as s_{k_1} in Algorithm 2) is selected first to withdraw its radio resources. Such an iteration stops if reallocating the resources of any existing incomplete request has no positive increase in J_2. After reallocating the radio resources of a selected incomplete request, resource fragments can appear, because part of the selected request's resources may be withdrawn but without new allocation. To increase algorithm efficiency, when further withdrawing and reutilizing the resources from other incomplete requests, these fragments are aggregated for reallocation. All the requests with resources being withdrawn are kept in a dropped request set, $\mathbb{S}_p$. The second part of the DSFRB algorithm plays a key role in the algorithm, reducing as much as possible the performance degradation due to the difference between maximizing usage-based system reward J_1 and user satisfaction-oriented system reward J_2.

Third, in steps 7 to 10, we check whether the remaining resources can serve any dropped request in $\mathbb{S}_p$, in which the one generating the largest reward will be served first and allocated the channel which it fits best according to link quality (see steps 8 and 9). Obviously, for a high algorithm efficiency, both resource fragments (i.e., in $\mathbb{R}$) and those held by incomplete requests (i.e., $\in \mathbb{S}_f$) should be treated as remaining resources to deliver requests in $\mathbb{S}_p$. It can be proved that resources in $\mathbb{R}$ are not sufficient to deliver any request in $\mathbb{S}_f$. Hence, in step 8, we only check requests in $\mathbb{S}_p$. The third part of the DSFRB algorithm helps to maximize radio resource utilization, thus further improving the algorithm performance.

As compared with the algorithm which always drops the request with minimal complete ratio proposed in [1], the newly proposed algorithm adopts a new deflation strategy, dropping incomplete request which improves revenue the most, which is shown in Section VIII to greatly improve network performance. In addition, as the SF algorithm is invoked only once in the new algorithm, the time complexity of the new algorithm as analyzed next is further reduced.

C. Complexity Analysis

The time complexity of the SF algorithm mainly depends on that in solving LP. If the interior point method is applied to solve LP OP5, the time complexity for performing arithmetic operations is $\mathcal{O}\left((INK)^{3.5} L^2\right)$, where (INK) and L are the dimension of the problem and the number of bits in the input [31]. Because the SF algorithm allocates at least one time-frequency resource unit (i,n) in solving one such LP problem, at most IN rounds of iterations are needed (with IN denoting the maximum number of time-frequency resource units) before ending the algorithm, where each round solves one IN LP problem. Therefore, the time complexity of the SF algorithm is at most $\mathcal{O}\left((INK)^{3.5} L^2 (IN)\right)$.

The analysis of time complexity of the DSFRB algorithm can be separated into three parts. In steps 1 to 3, we simply invoke the SF algorithm; therefore, the time complexity is that of the SF algorithm. For steps 4 to 6, we focus on the number of comparisons to be made. In step 4, for any incomplete request s_v we first find all elements $s^*_{v_j}$'s which compose set $\mathbb{S}_c(s_v)$. Specifically, to find $s^*_{v_1}$, given at most $K-1$ other incomplete requests, we require $K-1$ times of comparisons to decide whether each of them can be finished if allocating the s_v's radio resources to it, and then need $K-2$ times of comparisons to identify the one generating the largest reward (see (13)), resulting in total $(K-1)+(K-2)$ times of comparisons. By the same token, for $s^*_{v_j}$, at most $(K-j)+(K-j-1)$ times of comparisons are needed. Therefore, given incomplete request s_v, to find out $\mathbb{S}_c(s_v)$ thus to calculate $\Delta J_2(s_v)$ (see (14)), the total maximum comparison number is $\sum_{j=1}^{K-1} (K-j)+(K-j-1) = (K-1)^2$. As the maximum number of such incomplete requests is K, to find s_{k_1} in step 5, we look for $\mathbb{S}_c(s_v)$ and calculate $\Delta J_2(s_v)$ for each of them, needing in total at most $K(K-1)^2 + (K-1)$ times of comparisons, where the second term is from the comparisons of these $\Delta J_2(s_v)$'s. Then, in step 6, to reallocate the radio resources of s_{k_1} sequentially to the selected requests in $\mathbb{S}_c(s_v)$, for any request in $\mathbb{S}_c(s_v)$ sorting the channels according to channel quality (i.e., data rate) adds $(IN)^2$ times of comparisons in the worst case; then, checking whether the request has finished data transmission while allocating it the resource units one-by-one adds at most another IN times of comparisons. As the maximum size of $\mathbb{S}_c(s_v)$ is $K-1$, the total comparison number in step 6 is thus no more than $\left((IN)^2 + IN\right)(K-1)$. Finally, in steps 4 to 6, every time when we drop an incomplete request, at least another incomplete request can be finished. Given K initial

incomplete requests, there are at most $K/2$ rounds of resource reallocation, resulting in time complexity in the worst case of $\mathcal{O}\left(\left(\left(K(K-1)^2+(K-1)\right)+\left((IN)^2+IN\right)\right)K/2\right)$ for steps 4 to 6. By applying similar analysis, we can find that steps 7 to 10 have time complexity in the worst case of $\mathcal{O}\left(\left(((K-1)+(K-2))+\left((IN)^2+IN\right)\right)(K-1)\right)$.

VI. Online Algorithm Design

To compute a good schedule for OP1, the proposed DS-FRB algorithm requires the complete knowledge of all future request demands and channel capacities. However, such information in practice is not known a priori. In order to achieve efficient resource allocation for on-demand data services exploiting user behavior information, in this section, we present efficient online algorithms based on TFRC-oriented online request model in Section III-B3.

Here, we choose to design new online algorithms based on Smith ratio algorithm and exponential capacity algorithm [32], because they not only show good competitive ratio performance[5] but also have tractable analysis framework [33], based on which we will analytically study the benefits of integrating TFRC with scheduling in the next section. Yet, as the original algorithms are proposed for scheduling with a single machine of fixed processing ability, directly applying them in a multiple-channel time-varying-channel capacity scenario can be inefficient. A new strategy to take both multiple channels and time-varying channel capacities into account is required.

With the collected user behavior information, requests that can be scheduled in any slot i must be in set $\mathbb{S}_i^{\mathrm{On-TF}} = \left\{ s_k |\, G_k \le i \le D_k^{\mathrm{TF}}(i) \right\}$. When traffic load is heavy, allocating resources to every request in $\mathbb{S}_i^{\mathrm{On-TF}}$ incurs the risk of reducing not only revenue of a network operator but also quality of user experience, as requests may be served without considering their unique properties (in terms of channel quality, reward value, and remaining data amount). A link rate prediction-based leaky bucket-like traffic shaping strategy is proposed here to address the issue.

For simplicity, we use the same symbol, $Q_k^{(r)}$, to denote the remaining data amount of request s_k before current slot, and $\tilde{Q}_k^{(r)}$ to denote that after current slot. Assume that the future link rate of a channel for a given request can be estimated by its mean value of the previous link rates over the same channel, thus denoting the predicted link rate as $\hat{C}_{i,n,k} = E\left[C_{t,n,k} | t < i\right]$. Such a prediction can be applied to a scenario of stationary or low mobility users; however, as user mobility increases, averaging the link rates within an appropriately selected time duration improves the accuracy of the prediction. Then, the proposed link rate prediction-based leaky bucket-like traffic shaping strategy further constrains requests to be scheduled in slot i in the following set

[5]For a preemptive scheduling problem with bounded processing time and arbitrary weights, any deterministic online algorithm has competitive ratio at least $k/\ln k$, but the Smith ratio algorithm has ratio $\Theta(k)$ and the exponential capacity algorithm has ratio $\mathcal{O}(k/\ln k)$ [32].

Algorithm 3: Modified Smith ratio/exponential capacity algorithm

Input: $\left\{ G_k, D_k^{\mathrm{TF}}(i), Q_k, \omega_k, Q_k^{(r)}, \forall s_k \in \mathbb{S}_i^{\mathrm{On-TF}} \right\}$

Output: $\left\{ x_{i,n,k}, \tilde{Q}_k^{(r)}, \forall s_k \in \mathbb{S}_i^{\mathrm{On-TF}} \right\}$

1. Initialize $x_{i,n,k} = 0$, $\tilde{Q}_k^{(r)} = Q_k^{(r)}$, for $s_k \in \mathbb{S}_i^{\mathrm{On-TF}}$, $n = 1$;
2. While $n \le N$ do
3. Calculate Δ;
4. If $\Delta = \emptyset$, end algorithm; otherwise go to step 5;
5. Calculate U_k, for $s_k \in \Delta$;
6. Find $k^* = \arg\max_{s_k \in \Delta} \{U_k\}$;
7. Update $x_{i,n,k^*} = 1$, $\tilde{Q}_{k^*}^{(r)} = \max\left\{ \tilde{Q}_{k^*}^{(r)} - C_{i,k^*}^n, 0 \right\}$, $n = n+1$;
8. End while.

$$\Delta = \left\{ s_k |\, s_k \in \mathbb{S}_i^{\mathrm{On-TF}}, \tilde{Q}_k^{(r)} > 0, \sum_{m=n}^{N} C_{i,m,k} + \sum_{j=i+1}^{D_k^{\mathrm{TF}}(i)} \sum_{n=1}^{N} \hat{C}_{j,n,k} \ge \tilde{Q}_k^{(r)} \right\}. \tag{15}$$

Here, we assume that channels are sequentially allocated to requests. When channels with index less than n have been allocated, request s_k will be in Δ if and only if it is in $\mathbb{S}_i^{\mathrm{On-TF}}$ and it is possible to complete the service before its deadline by using all available channels (i.e., channels n to N in current slot and all channels in any future slot before its deadline). Further, in (15), for a request without allocating any channel, we have $\tilde{Q}_k^{(r)} = Q_k^{(r)}$. Based on the strategy, we propose our modified Smith ratio or exponential capacity-based online algorithm, as given in Algorithm 3, where U_k in step 7 represents the utility of request s_k. For the modified Smith ratio (MSR) algorithm, U_k is given by

$$U_k = \omega_k C_{i,n,k} / Q_k. \tag{16}$$

For the modified exponential capacity (MEC) algorithm, U_k is given by

$$U_k = \omega_k C_{i,n,k} \left(1 - \frac{\ln\left(\max_{s_k \in \Delta} Q_k\right)}{\max_{s_k \in \Delta} Q_k}\right)^{\tilde{Q}_k^{(r)} - 1}. \tag{17}$$

Different from the original Smith ratio or exponential capacity algorithm, the modified one incorporates link rate $C_{i,n,k}$ so that channel quality becomes a factor in determining channel allocation. For the MSR algorithm, a request with a higher reward or a smaller size or a better channel quality can obtain a higher utility. For the MEC algorithm, as the utility function incorporates the remaining data amount $\tilde{Q}_k^{(r)}$, a request with less remaining data also tends to acquire a higher utility. Whereby, the algorithms can iteratively allocate channels to requests in descending order of their utilities, until the channels are used up. Both algorithms can be applied to TFRC-disabled scenarios if we replace $\mathbb{S}_i^{\mathrm{On-TF}}$ by $\mathbb{S}_i$ in the algorithms.

For the time complexity of Algorithm 3, we find out that the MSR or MEC algorithm has time complexity $\mathcal{O}\left(N(INK + (K-1))\right)$ in the worst case. We omit the straightforward proof.

VII. Performance Analysis

In this section, based on the proposed online algorithms, we analytically explore the benefits of integrating TFRC with scheduling techniques. Specifically, we analyze the performance gap between TFRC-enabled and TFRC-disabled MSR

or MEC algorithm, by deriving the ratio of the total reward achieved by TFRC-enabled schedule to that by TFRC-disabled counterpart. Similar to the definition of competitive ratio used in comparing the performance between an offline algorithm and its online version, we refer to such derived ratio as *TFRC-oriented competitive ratio* (TOCR). For analysis traceability, we consider only a scenario of one single channel with time-invariant channel capacity. Notice that the solution of OP2 is the optimal schedule for the problem of interest here.

A. Preliminaries

The analysis is built on the charging scheme proposed in [32,33] which can be used to analyze many online algorithms satisfying both the monotonicity and validity properties. Before introducing the two properties, we define some notations for the following analysis.

For a request s_k, denote $q_k(t)$ as its remaining transmission time at time t. In a clear context, we denote it as q_k. A request without any service has $q_k = p_k$, where p_k equal to $\lceil Q_k/C \rceil$ is defined in the proof of Proposition 1. We say a request s_k is pending for a TFRC-disabled algorithm at time t if it has not been completed but still has a chance to be completed, i.e., $G_k \leq t$ and $t + q_k(t) \leq D_k$. As each request has its service deadline, we define $\beta_k = \max\{\nu : \nu + q_k(t) = D_k\}$ as the critical time for request s_k, if TFRC is disabled. In general, the critical time of a request can be viewed currently as the latest time that must be used to serve the request, otherwise it cannot be completed. Further, let $c_k(t) = D_k^{\mathrm{TF}}(t) - D_k$ denote the deadline extension for s_k at time t when user behavior is exploited by the resource manager, which is simply denoted as c_k in a clear context. Among all requests arrived in the scheduling duration of interest, let $c_{\max}$ and $p_{\max}$ denote the maximum request deadline extension and the maximum request transmission time, respectively. As time is divided into slots, every request s_k naturally contains p_k units. These units can be denoted by (k, a), for $1 \leq a \leq p_k$, for the unit of s_k whose transmission started when there were a units remaining. With each unit (k, a), we define *utility*[6] $\pi(k, a)$, whose exact value depends on ω_k and a, and can be different from algorithm to algorithm.

The properties of an analyzed algorithm, defined in [32], are given as follows.

- ρ-monotonicity: If the algorithm schedules (k, a) with $a > 1$ at time t but (k', a') at time $t + 1$, then it holds that $\rho\pi(k', a') \geq \pi(k, a)$, where $\rho \in (0, 1)$.
- Validity: If the algorithm schedules a unit (k, a) at time t but a request s_j is pending for the algorithm, then it holds that $\pi(k, a) \geq \omega_j/p_j$.

Informally, the monotonicity property means that, between two consecutive moments in which the algorithm serves some requests, the utilities of scheduled units are increasing. The validity property ensures that the utility of unit (k, a) is large enough to receive the penalty from a unit of pending request s_j. To apply the charging scheme of [32] to the analysis of TOCR for both MSR and MEC algorithms, we need the following proposition to be held.

[6]To differentiate from the capacity of a wireless channel, we name $\pi(k, a)$ the *utility* (rather than the *capacity* as used in [32]) of unit (k, a).

Proposition 3: Given one single channel with time-invariant channel capacity, both MSR and MEC algorithms, with or without TFRC, satisfy ρ-monotonicity and validity properties.

The proof of the proposition is straightforward, as integrating TFRC with scheduling techniques only impacts the deadline of each request, which has no relationship with the two properties of both algorithms. Furthermore, both the basic Smith ratio algorithm and the exponential capacity algorithm without TFRC have been proved to follow the two prosperities in [32] and it is easy to check that, for the single-channel time-invariant-channel capacity scenario, the proposed prediction-based leaky bucket-like traffic shaping strategy does not make the modified algorithms violate the two properties.

B. Analysis of TFRC-Oriented Competitive Ratio

The original charging scheme for analyzing competitive ratio of an online algorithm follows the outline: for every request s_j completed by the offline optimal algorithm we consider its p_j units. Each unit of the request charges ω_j/p_j to some request s_k completed by the online algorithm. If the charging schemes satisfy the property that every request s_i completed by the online algorithm receives a total charge of at most $R\omega_i$, we can claim R-competitiveness for the online algorithm. The approach is adopted here to analyze TOCR, where the TFRC-disabled (TFRC-enabled) schedule corresponds to the online (offline) algorithm studied in [32].

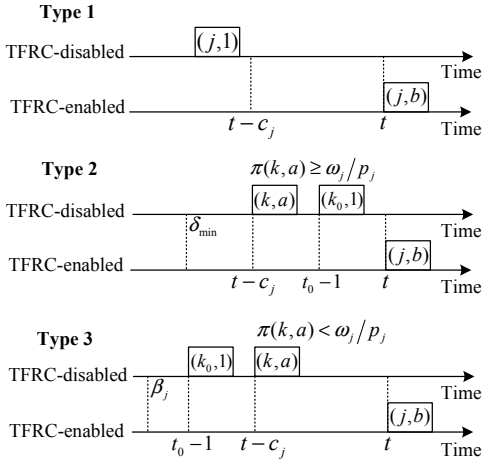


Fig. 2. An illustration of the charging scheme.

As shown in Fig. 2, by comparing the utility gained by a TFRC-enabled schedule and that gained by a TFRC-disabled schedule, we distinguish three types of charges in our charging scheme. A unit scheduled by either algorithm is referred to as a block. Let (j, b) be a unit of request s_j scheduled by the TFRC-enabled schedule at time t.

- **Type 1**: If a TFRC-disabled schedule has already completed $(j, 1)$ before $t - c_j$, then the TFRC-disabled schedule pays ω_j/p_j to the TFRC-enabled schedule.
- **Type 2**: Otherwise if a TFRC-disabled schedule scheduled a request unit (k, a) at time $t - c_j$ and its utility satisfies $\pi(k, a) \geq \omega_j/p_j$, then s_{k_0} (the first request completed by the TFRC-disabled schedule from time $t - c_j$ on) pays ω_j/p_j to the TFRC-enabled schedule.

- **Type 3**: In the last case, the TFRC-disabled schedule has served (k, a) by time $t - c_j$ and its utility satisfies $\pi(k, a) < \omega_j/p_j$. Thus, according to validity property, request s_j is not pending anymore from time $t - c_j$ on (though it is not completed). Let β_j denote s_j's critical time when disabling TFRC. The first completed request s_{k_0} by the TFRC-disabled schedule from time β_j on should pay ω_j/p_j to (j, b) in the TFRC-enabled schedule.

All the three types of charges result from inefficient resource utilization of the TFRC-disabled schedule. Specifically, in the first case, TFRC-disabled schedule is charged because it served request s_j too early to ignore the future chances to serve the request, as compared with the TFRC-enabled schedule that takes advantage of time-varying request deadline information. In the second case, compared with the TFRC-enabled counterpart, request s_{k_0} must be served (and finished) too early, thus again not utilizing future opportunities to schedule the request. In the last case, the TFRC-disabled schedule fails to finish request s_j but its TFRC-enabled counterpart keeps scheduling the request, thus the request is allocated resources and the first completed request s_{k_0} by the TFRC-disabled schedule from s_j's critical time on should be charged. Next we derive the upper bound of each type of the charges that a request s_{k_0} completed by TFRC-disabled schedule should pay to its TFRC-enabled counterpart.

Lemma 1: The total type 1 charge that request s_{k_0} completed by a TFRC-disabled schedule should pay to its TFRC-enabled counterpart is at most ω_{k_0}.

We prove Lemma 1 in Appendix B.

Lemma 2: The total type 2 charge that request s_{k_0} completed by a ρ-monotone and valid TFRC-disabled schedule should pay to its TFRC-enabled counterpart is at most $\pi(k_0, 1)[1/(1-\rho) + c_{\max}]$.

We prove Lemma 2 in Appendix C.

Lemma 3: The total type 3 charge that request s_{k_0} completed by a ρ-monotone and valid TFRC-disabled schedule should pay to its TFRC-enabled counterpart is at most $(c_{\max} + p_{\max} - 1)\pi(k_0, 1)$.

We prove Lemma 3 in Appendix D.

Based on Lemmas 1-3, given a single channel with time-invariant channel capacity, we obtain the upper bound of TOCRs for both MSR and MEC algorithms in the following two theorems.

Theorem 1: The TOCR of the Smith ratio-based algorithm is at most $2c_{\max} + 2p_{\max}$.

Proof: We use the charging scheme to prove the theorem. Each request s_{k_0} completed by the TFRC-disabled Smith ratio-based algorithm receives in total at most ω_{k_0} type 1 charge, $\pi(k_0, 1)[1/(1-\rho) + c_{\max}]$ type 2 charge, and $(c_{\max} + p_{\max} - 1)\pi(k_0, 1)$ type 3 charge, respectively. Thus, the TOCR of the Smith ratio-based algorithm is bounded by

$$\frac{\omega_{k_0} + \pi(k_0, 1)[1/(1-\rho) + c_{\max}] + (c_{\max} + p_{\max} - 1)\pi(k_0, 1)}{\omega_{k_0}} \tag{18}$$

Finally, as the Smith ratio-based algorithm is $\frac{p_{\max}-1}{p_{\max}}$-monotone and valid for $\pi(k, a) = \omega_k/a$ [32], by substituting $\rho = \frac{p_{\max}-1}{p_{\max}}$ and $\pi(k_0, 1) = \omega_{k_0}$ into (18), we obtain the result. ■

Theorem 2: The TOCR of the exponential capacity-based algorithm is at most $2c_{\max} + p_{\max} + \frac{p_{\max}}{\ln(p_{\max})}$.

Proof: The proof of Theorem 2 is similar to that for Theorem 1. The exponential capacity-based algorithm is $\left(1 - \frac{\ln(p_{\max})}{p_{\max}}\right)$-monotone and the utility of (k, a) is $\pi(k, a) = \omega_k\left(1 - \frac{\ln(p_{\max})}{p_{\max}}\right)^{a-1}$ [32]. ■

Based on the analytical results, the performance gap between TFRC-enabled schedule and TFRC-disabled schedule increases with the maximum deadline extension $c_{\max}$ and the maximum processing time $p_{\max}$. However, the impact of the two parameters on the algorithms is different. We verify the analytical results by simulation in the next section.

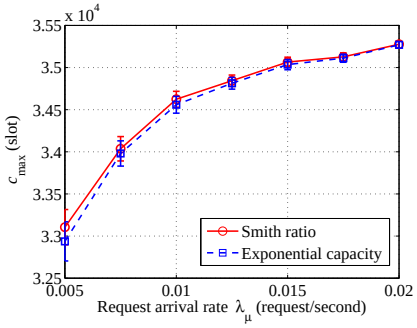
VIII. Numerical Results and Discussion

In this section, we evaluate the performance of the proposed TFRC-enabled scheduling techniques and verify the theoretical analysis. In the simulation, we assume that each of the M users in the cell initiates new requests according to a Poisson process with rate λ_u. The data size Q_k and lifetime Π_k of each request are uniformly distributed within $[Q_{\min}, Q_{\max}]$ and exponentially distributed with mean μ, respectively. The reward of each request's unit data (A_k) is assumed to be uniformly distributed within $[1, 10]$. To capture the main feature of user focus, in the simulation we allow each user to only focus on one foreground application at any instant, but multiple applications can be in transmission in the background simultaneously. Moreover, the user is set to keep focusing on the foreground application until it ends or a new connection arrives, while the context information on user behavior is fed back from UE to BS once every slot of 100 ms. For all simulation results, we perform the simulation for 200 runs, average the results, and obtain the 95% confidence intervals.

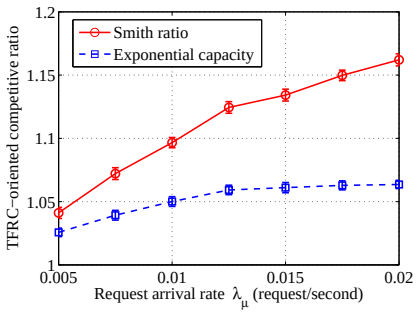
A. TFRC-Oriented Competitive Ratio

We first perform simulations to verify the analytical results of TFRC-oriented competitive ratio for both Smith ratio and exponential capacity-based algorithms. As shown in Section VII, there are two key parameters, the maximum request deadline extension $c_{\max}$ and the maximum request transmission time $p_{\max}$, impacting the performance gap. Yet, unearthing the impact of $c_{\max}$ is not so straightforward as for $p_{\max}$, as we cannot control the value of $c_{\max}$ directly. So, for the impact of $c_{\max}$, we first explore its relationship with other parameters. Specifically, considering the scenario of 10 users sharing one single channel with time-invariant channel capacity, we study the impact of traffic load (thus $c_{\max}$ indirectly) and request size (equivalent to $p_{\max}$) on this performance gap. Here, each simulation run sustains a network time of 1 hour.

Fig. 3 shows the impact of traffic load (request arrival rate λ_u) with $C = 10^4$bps, $Q_{\min} = 0.1$Mbits, $Q_{\max} = 0.9$Mbits, and $\mu = 60$s. From Fig. 3(a), it can be observed that the maximum deadline extension $c_{\max}$ for each algorithm increases as the traffic load increases. Recall that TOCR increases with $c_{\max}$ which, according to Fig. 3(a), implies that TOCR or the performance gap between TFRC-enabled schedule and its TFRC-disabled counterpart should increase with traffic load as well. By comparing the total rewards that both types of

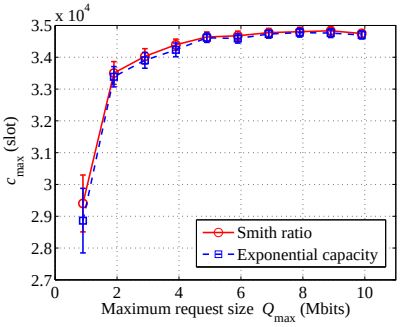


(a) $c_{\max}$ vs. λ_μ.

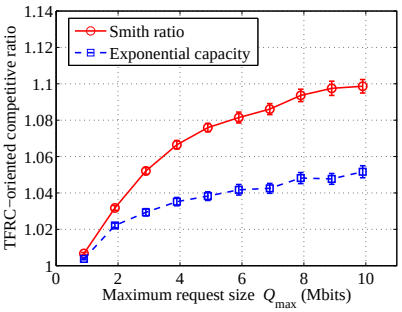


(b) TFRC-oriented competitive ratio vs. λ_μ.

Fig. 3. Impact of λ_μ on $c_{\max}$ and TFRC-oriented competitive ratio.



(a) $c_{\max}$ vs. $Q_{\max}$.



(b) TFRC-oriented competitive ratio vs. $Q_{\max}$.

Fig. 4. Impact of $Q_{\max}$ on $c_{\max}$ and TFRC-oriented competitive ratio.

schedule obtain in the simulations, Fig. 3(b) corroborates the aforesaid relationship between the traffic load and TOCR.

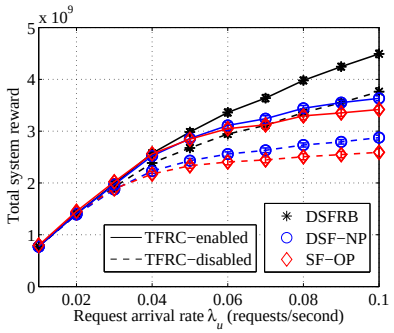
In Fig. 4, we study the impact of request size. To accommodate more data traffic, we increase channel capacity to 10^5bps. Per user request arrival rate (λ_u) is fixed to 0.0125 requests/s. The minimum request size keeps unchanged, while the maximum request size changes according to the mean request size which varies from 0.5Mbits to 5Mbits. Other parameters are set the same as those in Fig. 3. From Fig. 4(a), we observe that the maximum deadline extension $c_{\max}$ also increases with the request size. So, the performance gap between TFRC-enabled schedule and its counterpart increases with the request size, which has been verified in Fig. 4(b). Further, it is noted from Fig. 4(a) that $c_{\max}$ first increases quickly before $Q_{\max} = 4$Mbits but then converges thereafter as $Q_{\max}$ keeps increasing. Yet, from Fig. 4(b) it is clear that the increasing speed of TOCR for both algorithms reduces slightly as $Q_{\max}$ increases. This is because the performance gap between TFRC-enabled schedule and its counterpart not only depends on the maximum deadline extension $c_{\max}$ but is proportional to the maximum request size $Q_{\max}$ (i.e., $p_{\max}$ in Theorems 1 and 2). The convergence of $c_{\max}$ in Fig. 4(a) is due to the bounded mean lifetime of each request and the limited network simulation time (1 hour). From Fig. 3(b) and Fig. 4(b), the performance improvement of Smith ratio algorithm due to TFRC is larger than that of exponential capacity algorithm, which is consistent with Theorems 1 and 2.

B. Scheduling Performance

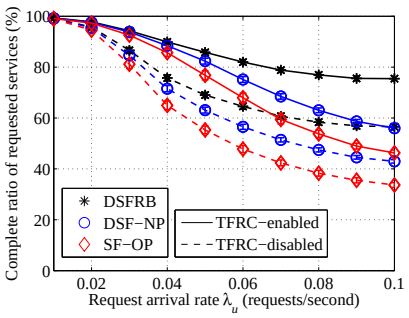
To evaluate the proposed TFRC-enabled scheduling techniques, more extensive simulations are run for both offline and online scheduling. For offline scheduling, we focus on the performance improvement of TFRC-enabled schedule and the benefits of the newly proposed penalty function, while for online scheduling we measure the performance gap between the proposed offline and multiple online algorithms. Specifically, for offline scheduling, two benchmark algorithms are compared with the proposed DSFRB algorithm: the one proposed in [1] and integrated with the new penalty function (denoted as "DSF-NP") and the one with the traditional penalty function (denoted as "SF-OP")[7]. For online scheduling, two other benchmark algorithms are compared with the proposed MSR and MEC algorithms: the algorithm with earliest-deadline-first policy (denoted as "EDF") and the algorithm proposed in [34] (denoted as "L-MaxWeight"). L-MaxWeight is tailored for scheduling flows with deadlines and is shown superior performance in underloaded identical-deadline systems. In the simulations, for either scenario there are 5 users being severed by a cell with 32 subchannels, each channel with a bandwidth of 15KHz. The channel fading is modeled by the Rayleigh distribution. The impacts of different parameters, including traffic load, request size, mean request lifetime, and channel condition, on scheduling performance are studied.

1) Offline Scheduling: Fig. 5 shows the impact of traffic load on both total system reward (operator side) and complete ratio of all requests (user side), with $\lambda_u \in [0.01, 0.1]$ requests/s, $Q_{\min} = 15$ Mbits, $Q_{\max} = 20$ Mbits, $\mu = 30$ s, and mean signal-to-noise ratio (SNR) for each channel equal to 10 dB. Unless otherwise specified, simulation for effects of other parameters are all based on the same setting, and the changed parameters are listed. In Fig. 5, both results for the TFRC-enabled and TFRC-disabled DSFRB, DSF-NP, and SF-OP algorithms are presented. It is observed that the algorithms

[7]Scheduling with the traditional penalty function is to solve a mixed integer linear OP (see Proposition 2). So, we can solve the OP directly with the SF algorithm effectively.

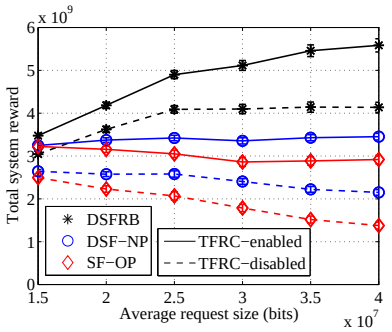


(a) Total reward vs. traffic load.

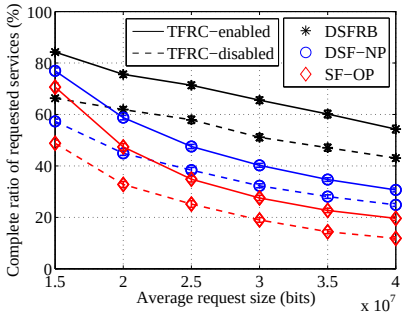


(b) Complete ratio vs. traffic load.

Fig. 5. Impact of traffic load on total system reward and complete ratio with offline schedule.



(a) Total reward vs. average request size.



(b) Complete ratio vs. average request size.

Fig. 6. Impact of average request size on total system reward and complete ratio with offline schedule.

enabling TFRC outperform their counterpart disabling TFRC, in terms of the system reward and complete ratio, generating a potential win-win situation for both the operator and the user. The performance improvement results from the fact that the CI utilized in the TFRC-enabled schedule offers more freedom to find a better scheduling solution. Further, the performance gap, especially the one for DSFRB, increases with traffic load, illustrating the potential advantage of TFRC-enabled schedule in addressing a heavy-load network scenario. In Fig. 5, a performance gap between DSFRB and DSF-NP exists, mainly due to the newly proposed deflation strategy. Both DSFRB and DSF-NP outperform SF-OP, illustrating an advantage of the new penalty function over the traditional one in increasing delivered request number. Taking comparison among TFRC-disabled algorithms as an example, the average total system reward improvement (average improvement of complete ratio of requested services) here between DSFRB and DSF-NP and that between DSFRB and SF-OP are 13.4% and 18.9% (13.3% and 30.9%), respectively.

Fig. 6 shows how the total system reward and the complete ratio change with the average request size, with $\lambda_u = 0.075$ requests/s, $Q_{\min} = 10$ Mbits, and $Q_{\max}$ changing with the average request size from 15 Mbits to 40 Mbits. We can see that the complete ratio of request services with any of the three algorithms decreases with an increase of the average request size, as the traffic load increases with the average request size. The TFRC strategy makes each algorithm avoid suffering too much from the increased traffic load, as observed in Fig. 5. Further, it is shown in Fig. 6(a) that, without applying TFRC, the total system reward for DSF-NP or SF-OP decreases with the average request size; yet, the performance of algorithms enabling TFRC remains almost unchanged. On the other hand, the total system reward for DSFRB without TFRC increases slightly for the average request sizes; yet, a much larger increasing rate is observed for the algorithm enabling TFRC.

In Fig. 7, we study the impact of mean request lifetime, with $\lambda_u = 0.075$ requests/s and $\mu \in [10, 80]$ s. Due to space limitation, here only results of total system reward are provided. For results of complete ratio, the interested reader can refer to [35] for details. It is observed that the performance of the three algorithms, with or without TFRC, increases as the mean lifetime μ increases. With an increase of μ, each request has more time for resource allocation, thus more chances to be finished before deadline. Yet, DSFRB is much better than other two algorithms, for the same reasons as aforesaid. Also, by applying TFRC, all algorithms perform better but tend to converge when $\mu \geq 40$ s. This is because the cell capacity has been fully exploited by the TFRC strategy when $\mu \geq 40$ s.

Fig. 8 shows how the total system reward change with the channel quality. Here, we set $\lambda_u = 0.075$ requests/s, vary the mean SNR of each channel from 5 dB to 20 dB, and keep other parameters the same as for Fig. 5. The total system reward improves with the channel quality as the data transmission rate increases with it as well. Yet, it is clear that integrating TFRC with scheduling techniques helps each algorithm harvest much more potential benefits from the improved channel quality.

2) Online Scheduling: The good performance of TFRC-enabled schedule benefits from not only algorithm design but also non-causal information on request demand and channel capacity. Next, we evaluate its online counterpart, thus to understand the effect of TFRC-enabled scheduling techniques in a more practical scenario. In the following, all simulation settings are the same as those for offline scheduling. The results are normalized by TFRC-enabled DSFRB algorithm. Due to space limitation, only results of total system reward

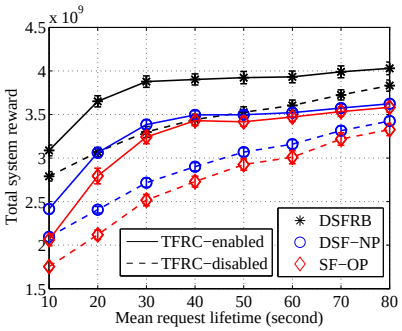


Fig. 7. Impact of mean request lifetime with offline schedule.

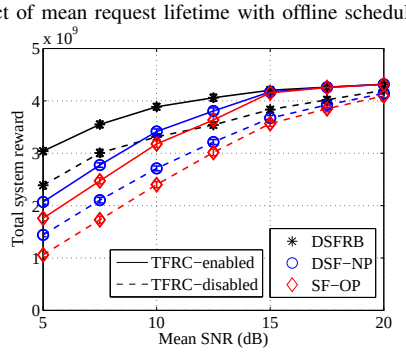


Fig. 8. Impact of mean SNR with offline schedule.

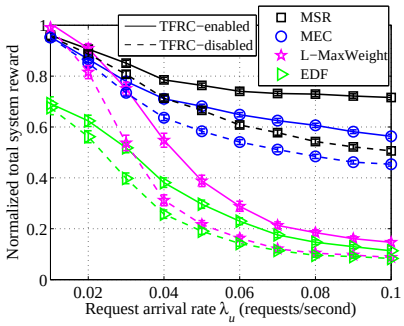


Fig. 9. Impact of traffic load with online schedule.

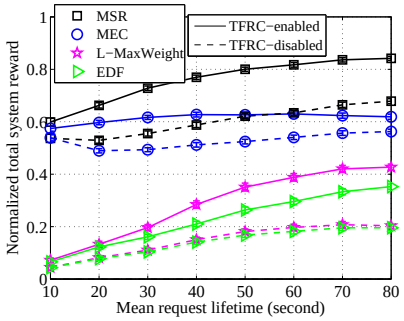


Fig. 10. Impact of mean lifetime with online schedule.

with respect to traffic load and mean request lifetime are provided. For results of complete ratio or with respect to other parameters (i.e., request size and channel condition), the interested reader can refer to [35] for details.

Fig. 9 shows the impact of traffic load. It is observed that each algorithm performs better if enabling TFRC. As the traffic load increases, the normalized total system reward with any of the compared algorithms decreases, and the performance gap between these online algorithms and the DSFRB algorithm increases. Yet, from the simulations we find that the absolute value of the total system reward with MSR or MEC always increases with the traffic load. Further, the two algorithms perform much better than L-MaxWeight and EDF, especially when traffic load is heavy. When enabling TFRC and $\lambda_\mu = 0.1$ requests/s, as compared with L-MaxWeight the improvement of MSR (MEC) can be 387.9% (283.8%). Yet, TFRC-enabled L-MaxWeight performs slightly better than MSR and MEC when request arrival rate is less than 0.02 requests/s, showing the advantage of L-MaxWeight in a low traffic load region.

Fig. 10 studies the impact of mean request lifetime. It can be seen that the performance gap between the online algorithms and the offline DSFRB algorithm reduces with mean request lifetime. The performance loss for TFRC-disabled MSR is reduced by 30.4% when the mean request lifetime increases from 10 s to 80 s, implying the benefits of extending request scheduling time. Besides, this improvement is more obvious if TFRC is enabled, e.g., for TFRC-enabled MSR the same performance loss can be further reduced by 60% when mean request life time changes within the same range.

IX. Conclusions

Time-frequency resource conversion is a recently proposed network resource allocation strategy. Integrating the strategy with call admission control increases the cell capacity and reduces network congestion. In this work, we focus on designing TFRC-based scheduling algorithms for on-demand data service with hard deadline constraints. We study the service provisioning problem for maximizing the total reward of completed requests, which however is NP-hard. Solving this problem needs information on future request demand and channel capacity, but offers a benchmark for the upper bound performance of TFRC-based scheduling techniques. To this end, a novel yet polynomial-time algorithm based on deflation and sequential fixing techniques has been proposed. For practical online implementation, we have proposed two TFRC-enabled low complexity algorithms, exploiting prediction-based leaky bucket-like traffic shaping and modified Smith ratio or exponential capacity-based utility function. Moreover, we have proposed an analytical model to study the benefits of utilizing user behavior information in designing scheduling algorithms. Simulation results show the effectiveness of the proposed algorithms and corroborate the advantages of the proposed TFRC-based schedule techniques in terms of QoS provisioning for each user and revenue improvement for a service operator. Further work to refine the proposed technique will be carried out to design more efficient online scheduler.

Appendix A: Proof of Proposition 2

Take scheduling without TFRC as an example. Let $f_1(\mathcal{X}) = J_1 - \sum_{k=1}^{K} p_1(k)$. To compare the performance of scheduling between maximizing $f_1(\mathcal{X})$ and maximizing the usage-based system reward J_1, we define a new objective function $f_2(\mathcal{X}) = J_1 - \sum_{k=1}^{K} \alpha A_k (Q_k - \sum_{i=1}^{I} \sum_{n=1}^{N} C_{i,n,k} x_{i,n,k})$, which can be further simplified as $f_2(\mathcal{X}) = \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{n=1}^{N} (1+\alpha) A_k C_{i,n,k} x_{i,n,k} - R_{\mathrm{T}}$, where $R_{\mathrm{T}} = \sum_{k=1}^{K} \alpha A_k Q_k$ is a constant. So, maximizing $f_2(\mathcal{X})$ is equivalent to maximizing J_1.

Obviously, $\forall \mathcal{X}$, $f_1(\mathcal{X}) \le f_2(\mathcal{X})$ holds, as $\alpha A_k \ge 0$ and $[x]^+ \ge x$. Let $\mathcal{X}_1$ ($\mathcal{X}_2$) denote the optimal solution of maximizing $f_1(\mathcal{X})$ ($f_2(\mathcal{X})$). Then, the following relationship should hold $f_1(\mathcal{X}_2) \le f_1(\mathcal{X}_1) \le f_2(\mathcal{X}_2)$.

Checking the gap between $f_1(\mathcal{X}_1)$ and $f_1(\mathcal{X}_2)$, we have $0 \leq f_1(\mathcal{X}_1) - f_1(\mathcal{X}_2) \leq f_2(\mathcal{X}_2) - f_1(\mathcal{X}_2) = \sum_{k=1}^{K} \alpha \cdot A_k(\Delta_k + [-\Delta_k]^+)$, where $\Delta_k = \sum_{i=1}^{I}\sum_{n=1}^{N} C_{i,n,k} x_{i,n,k}^{(2)} - Q_k$, with $x_{i,n,k}^{(2)}$ denoting the slot-channel-request allocation indicator decided in $\mathcal{X}_2$. Obviously, for the requests without finishing data transmission or allocated a total link capacity equal to data size (i.e., $\sum_{i=1}^{I}\sum_{n=1}^{N} C_{i,n,k} x_{i,n,k}^{(2)} \leq Q_k$), $\Delta_k + [-\Delta_k]^+ = 0$. Yet, due to slot-based and orthogonal resource allocation, there could be requests that finish data transmission may not fully utilize the link capacity of the last allocated subchannel (i.e., $\sum_{i=1}^{I}\sum_{n=1}^{N} C_{i,n,k} x_{i,n,k}^{(2)} > Q_k$). Let Ω denote the set consisting of these requests. Then the performance gap can be rewritten as $0 \leq f_1(\mathcal{X}_1) - f_1(\mathcal{X}_2) \leq \sum_{k\in\Omega} \alpha A_k \Delta_k \leq \sum_{k\in\Omega} \alpha\omega_k (C_{i(k),n(k),k} - 1)/Q_k$. Here $C_{i(k),n(k),k}$ represents the capacity of the link, with slot index $i(k)$ and subchannel index $n(k)$, which any request s_k ($\in \Omega$) finally occupies to finish its data transmission. The last inequality holds because $A_k = \omega_k/Q_k$ and $\max\{\Delta_k\} = C_{i(k),n(k),k} - 1$. Based on the last inequality we can see that the performance gap is small if as compared with the per-slot link capacity the data size of any request is large. Besides, the mean of the performance gap reduces to zero if all requests have the same average link capacity and time-frequency resources can be divided into the granularity of unit bit per channel allocation, i.e., $\mathrm{E}\left[C_{i(k),n(k),k}\right] = 1$, because $0 \leq \mathrm{E}\left[f_1(\mathcal{X}_1) - f_1(\mathcal{X}_2)\right] \leq \mathrm{E}\left[\sum_{k\in\Omega} \alpha\omega_k \frac{C_{i(k),n(k),k}-1}{Q_k}\right] = \sum_{k\in\Omega} \alpha \cdot \mathrm{E}\left[\frac{\omega_k}{Q_k}\right] \cdot \mathrm{E}\left[C_{i(k),n(k),k} - 1\right] = 0$, which ends the proof.

Appendix B: Proof of Lemma 1

As the maximal number of type 1 charges, which request s_{k_0} completed by TFRC-disabled schedule can receive, equals its unit number p_{k_0}, for p_{k_0} charges each with ω_j/p_j payment, the total type 1 charge it can pay is at most $(\omega_{k_0}/p_{k_0}) \cdot p_{k_0} = \omega_{k_0}$.

Appendix C: Proof of Lemma 2

As shown in Fig. 2, let t_0 denote the finish time of s_{k_0} by the TFRC-disabled schedule, and $\delta_{\min}$ be the smallest time such that during time interval $[\delta_{\min}, t_0)$ neither idle slot nor other request completion exists. Then, according to ρ-monotonicity property, the unit scheduled by TFRC-disabled schedule at time $t_0 - i$, for $i \in [1, t_0 - \delta_{\min} + 1]$, has utility no more than $\pi(k_0, 1)\rho^{i-1}$. Notice that type 2 charges to s_{k_0} in TFRC-disabled schedule can only originate from units (say (j, b)) that satisfy the following two constraints: 1) the units are scheduled by TFRC-enabled schedule at or after time $\delta_{\min} - 1$ (say time t); 2) if checking the timeline of TFRC-disabled schedule at $t - c_j$ ($\leq D_j$), which is just in $\Gamma = [\delta_{\min} - 1, t_0 - 1]$, the related request (i.e., s_j) is not completed by the TFRC-disabled schedule. According to the timeline of TFRC-enabled schedule, we analyze the type 2 charge by classifying the following two cases.

Firstly, consider the case that in TFRC-enabled schedule the unit (j, b) is scheduled before time t_0. According to validity property, for the charge we have $\omega_j/p_j \leq \pi(k, a)$, where (k, a) is the unit scheduled by TFRC-disabled schedule at time $t - c_j$. The upper bound of this charge is given at $c_j = 0$ due to ρ-monotonicity property. Thus, the total type 2 charge to s_{k_0} for the first case can be bounded by

$$\sum_{\substack{(k,a) \text{ scheduled} \\ \text{at time in } \Gamma}} \pi(k, a) \leq \pi(k_0, 1)(1 + \rho + \ldots + \rho^{t_0 - \delta_{\min}}) < \pi(k_0, 1)/(1 - \rho).$$

Secondly, consider the case that the unit (j, b) is scheduled at or later than time t_0. It can charge to request s_{k_0} completed by TFRC-disabled schedule if $t - c_j$ is in Γ. For the charge we have $\omega_j/p_j \leq \pi(k, a) \leq \pi(k_0, 1)$, where (k, a) is the unit scheduled by TFRC-disabled schedule at time $t - c_j$. So the next work is to find the largest t such that $t - c_j$ is still in Γ. As $t - c_j \in \Gamma$, we know $t \leq c_j + t_0 - 1 \leq c_{\max} + t_0 - 1$, where $c_{\max}$ is the maximum deadline extension of the requests over the scheduling duration of interest. Thus, the latest time $t_{\max}$ at which a unit scheduled by TFRC-enabled schedule can charge to s_{k_0} is $c_{\max} + t_0 - 1$, and there are at most $t_{\max} - t_0 + 1 = c_{\max}$ of such units scheduled at or later than t_0. Thus, the total type 2 charge to s_{k_0} for the second case can be bounded by $\pi(k_0, 1)c_{\max}$. Adding the results of the two cases, we complete the proof.

Appendix D: Proof of Lemma 3

Let π_{β_j} denote the utility of the unit scheduled by TFRC-disabled schedule at time β_j, where as shown in Fig. 2 β_j is the critical time of request s_j when disabling TFRC. According to validity property, we have $\omega_j/p_j \leq \pi_{\beta_j}$. During time interval $[\beta_j, t_0)$, neither idle slot nor other request completion (expect for s_{k_0}) exists. Thus, because of ρ-monotonicity property, we further have $\omega_j/p_j \leq \pi_{\beta_j} \leq \pi(k_0, 1)$, where $\pi(k_0, 1)$ is the utility of $(k_0, 1)$ scheduled by TFRC-disabled schedule at time $t_0 - 1$. That is, each unit (j, b) in TFRC-enabled schedule can get type 3 charge at most $\pi(k_0, 1)$.

The next step is to find the maximum number of such units that can charge to s_{k_0} completed by TFRC-disabled schedule. As unit (k, a) scheduled by TFRC-disabled schedule at time $t - c_j$ has utility $\pi(k, a)$ less than ω_j/p_j thus also less than π_{β_j}, we must have $t - c_j \geq t_0$, by utilizing ρ-monotonicity property over time interval $[\beta_j, t_0)$. Hence, the maximum number of such units is $\max\left\{t - t_0 + 1 | t < D_j^{\mathrm{TF}}(t), \beta_j \leq t_0 - 1\right\}$, which can be further derived as follows $t - t_0 + 1 \leq t - \beta_j \leq (D_j^{\mathrm{TF}}(t) - 1) - \beta_j = (D_j^{\mathrm{TF}}(t) - D_j) + (D_j - \beta_j) - 1 \leq c_j + p_j - 1 \leq c_{\max} + p_{\max} - 1$. Thus, the total type 3 charge to s_{k_0} can be bounded by $(c_{\max} + p_{\max} - 1) \cdot \pi(k_0, 1)$.

References

[1] Y. Zhang *et al.*, "Time-frequency resource conversion based scheduling for on-demand data services," in *Proc. IEEE Globecom*, 2015.

[2] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019," USA, Feb. 2015.

[3] B. Bangerter *et al.*, "Networks and devices for the 5G era," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 90-96, Feb. 2014.

[4] C.-X. Wang *et al.*, "Cellular architecture and key technologies for 5G wireless communication networks," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 122-130, Feb. 2014.

[5] J. Wu *et al.*, "[Guest Editorial] Context-aware networking and communications: Part 1," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 14-15, June 2014.

[6] M. Proebster *et al.*, "Context-aware resource allocation to improve the quality of service of heterogeneous traffic," in *Proc. IEEE ICC*, 2011.

[7] M. Proebster *et al.*, "Context-aware resource allocation for cellular wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, p. 216, Jul. 2012.

[8] H. Abou-zeid and H. S. Hassanein, "Toward green media delivery: Location-aware opportunities and approaches," *IEEE Wireless Commun.*, vol. 21, no. 4, pp. 38-46, Aug. 2014.

[9] H. Abou-zeid and H. S. Hassanein, "Predictive green wireless access: Exploiting mobility and application information," *IEEE Wireless Commun.*, vol. 20, no. 5, pp. 92-99, Oct. 2013.

[10] H. Abou-zeid *et al.*, "Energy-efficient adaptive video transmission: Exploiting rate predictions in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2013-2026, June 2014.

[11] R. Schoenen and H. Yanikomeroglu, "User-in-the-loop: Spatial and temporal demand shaping for sustainable wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 196-203, Feb. 2014.

[12] O. Semiari *et al.*, "Context-aware small cell networks how social metrics improve wireless resource allocation," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 5927-5940, Nov. 2015.

[13] H. Shan *et al.*, "Virtual spectrum hole: Exploiting user behavior-aware time-frequency resource conversion," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 6809-6823, Dec. 2014.

[14] Y. Lu, M. Motani, and W.-C. Wong, "A QoE-aware resource distribution framework incentivizing context sharing and moderate competition," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1364-1377, June 2016.

[15] F. Akhtar, M. H. Rehmani, and M. Reisslein, "White space: Definitional perspectives and their role in exploiting spectrum opportunities," *Telecommunications Policy*, vol. 40, no. 4, pp. 319-331, April 2016.

[16] C. S. In, R. Jain, and A. K. Tamimi, "Scheduling in IEEE 802.16e mobile WiMAX networks: Key issues and a survey," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 156-171, Feb. 2009.

[17] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in OFDMA wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 322-337, Second Quarter, 2012.

[18] G. Maier *et al.*, "A first look at mobile handheld device traffic," in *Proc. ACM 11th Int. Conf. Passive Active Netw. Meas.*, pp. 161-170, 2010.

[19] R. Kuhne, G. Huitema, and G. Carle, "Charging and billing in modern communications networks - A comprehensive survey of the state of the art and future requirements," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 170-192, First Quarter 2012.

[20] Z. Ezziane, "Charging and pricing challenges for 3G systems," *IEEE Commun. Surveys Tuts.*, vol. 7, no. 4, pp. 58-68, Fourth Quarter 2005.

[21] A. Gausby, "Attention spans," *Microsoft research report*, Spring 2015.

[22] L. Leiva *et al.*, "Back to the app: The costs of mobile application interruptions," in *Proc. ACM MobileHCI*, 2012.

[23] H. Weinreich *et al.*, "Not quite the average: An empirical study of Web use," *ACM Trans. the Web*, vol. 2, no. 1, pp. 1-31, Feb. 2008.

[24] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Dordrecht, Heidelberg, London, New York: Springer, 2012.

[25] E. L. Lawler, "A dynamic programming algorithm for preemptive scheduling of a single machine to minimize the number of late jobs," *Annals of Operations Research*, vol. 26, no. 1, pp. 125-133, Dec. 1990.

[26] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Operational Research Society*, vol. 49, no. 3, pp. 237-252, Mar. 1998.

[27] S. Li and J. Huang, "Price differentiation for communication networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 703-714, June 2014.

[28] L. Zhuang *et al.*, "Time dependent pricing in wireless data networks: Flat-rate vs. usage-based schemes," in *Proc. IEEE INFOCOM*, 2014.

[29] E. Matskani *et al.*, "Convex approximation techniques for joint multiuser downlink beamforming and admission control," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2682-2693, Jul. 2008.

[30] L. A. Wolsey, *Integer Programming*. John Wiley and Sons Press, 1998.

[31] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373-395, 1984.

[32] C. Durr *et al.*, "Online scheduling of bounded length jobs to maximize throughput," *J. Sched.*, vol. 15, no. 5, pp. 653-664, Oct. 2012.

[33] N. K. Thang, *Pure Equilibria: Existence and Inefficiency & Online Auction*. PhD thesis, Ecole Polytechnique, France, 2009.

[34] H. Wu, X. Liu, and Y. Zhang, "Laxity-based opportunistic scheduling with flow-level dynamics and deadlines," in *Proc. IEEE WCNC*, 2013.

[35] H. Shan *et al.*, "Simulation results of user behavior-aware scheduling based on time-frequency resource conversion," Available at: arXiv:1602.04900.

Hangguan Shan (M'10) has been with the College of Information Science and Electronic Engineering, Zhejiang University, since 2011, where he is currently an Associate Professor. His current research focuses on cross-layer protocol design, resource allocation, and the quality-of-service provisioning in wireless networks. He is a co-recipient of the Best Industry Paper Award from the 2011 IEEE WCNC. Dr. Shan is currently an Editor of IEEE Transactions on Green Communications and Networking.

Yani Zhang received the B.S. degree from Chongqing University, China, in 2013, and the M.S. degree from Zhejiang University, China, in 2016, both in information and communication engineering. She is currently working in the Meteorological Information and Network Center of Zhejiang Province, China. Her current research interests are in the areas of wireless networking, cloud computing, and the application of big data in meteorological service.

Weihua Zhuang (M'93-SM'01-F'08) has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, since 1993, where she is a Professor and a Tier I Canada Research Chair in Wireless Communication Networks. Her current research focuses on resource allocation and QoS provisioning in wireless networks, and on smart grid. She is a co-recipient of several best paper awards from IEEE conferences. Dr. Zhuang was the Editor-in-Chief of IEEE Transactions on Vehicular Technology (2007-2013), and the TPC Co-Chair of IEEE VTC Fall 2016. She is a Fellow of the IEEE, a Fellow of the Canadian Academy of Engineering, a Fellow of the Engineering Institute of Canada, and an elected member in the Board of Governors and VP Publications of the IEEE Vehicular Technology Society.

Aiping Huang (SM'08) has been with the College of Information Science and Electronic Engineering, Zhejiang University, since 1998, where she is a full Professor. She has authored a book and more than 190 papers in refereed journals and conferences on communications, networking, and signal processing. Her current research interests include heterogeneous networks, performance analysis, cross-layer design, and planning and optimization of cellular mobile communication networks. Dr. Huang serves as a Vice-Chair of the IEEE ComSoc Nanjing Chapter.

Zhaoyang Zhang (M'00) received his Ph.D. degree in communication and information systems from Zhejiang University, Hangzhou, China, in 1998. Since then he has been with the College of Information Science and Electronic Engineering, Zhejiang University, where he became a full professor in 2005. He has a wide variety of research interests including information theory and coding theory, signal processing for communications and in networks, computation-and-communication theoretic analysis, etc. He is a co-recipient of four international conference Best Paper/Best Student Paper Awards. He is currently serving as Editor for IEEE Transactions on Communications, IET Communications and some other international journals. He served as General Chair, TPC Co-Chair or Symposium Co-Chair for many international conferences or workshops like ChinaCOM 2008, ICUFN 2011/2012/2013, WCSP 2013, Globecom 2014 Wireless Communications Symposium, and HMWC 2017, etc.