Achieving Fair LTE/Wi-Fi Coexistence with Real-Time Scheduling

Yan Huang, *Student Member, IEEE*, Yongce Chen, *Student Member, IEEE*, Y. Thomas Hou, *Fellow, IEEE*, and Wenjing Lou, *Fellow, IEEE*

***Abstract*—Carrier-Sensing Adaptive Transmission (CSAT) is a promising approach to address coexistence between LTE and Wi-Fi in unlicensed bands. Under CSAT, a key problem is the design of a scheduling algorithm to allocate radio resources (across multiple channels and a large number of sub-channels) for LTE and Wi-Fi users. This paper investigates this scheduling problem through an optimization formulation with the objective of minimizing LTE's adverse impact on Wi-Fi users. Special considerations of each LTE user's uplink/downlink rate requirements and channel conditions are given in this optimization formulation. We show that this scheduling problem is NP-hard and propose to develop a near-optimal solution. A major challenge here is to ensure the scheduler can obtain a solution on ∼1 ms time scale — a stringent timing requirement to meet LTE standard. Our main contribution is the development of CURT, a scheduling algorithm that can obtain near-optimal solution in ∼1 ms under standard LTE small cell scenarios. CURT exploits the unique structure of the underlying optimization problem and decomposes it into a large number of independent sub-problems. By taking advantage of GPU's parallel processors, we allow the large number of sub-problems to be run in parallel and independently from each other. By implementing CURT on Nvidia GPU/CUDA platform, we demonstrate that CURT can deliver near-optimal scheduling solution in ∼1 ms for LTE small cells with no more than 20 users following 3GPP's evaluation methodology.**

***Index Terms*—Coexistence, LTE, Wi-Fi, unlicensed spectrum, scheduling, optimization, real-time, GPU.**

I. Introduction

THERE is a strong interest from cellular carriers to use existing unlicensed spectrum (e.g., the 5 GHz UNII bands) to boost cellular services. This approach is appealing for a number of reasons: (i) unlicensed spectrum is free (no need of auction and a license fee), (ii) the underlying bandwidth is substantial (e.g., 775 MHz available bandwidth in 5 GHz UNII bands), (iii) coexisting with other unlicensed wireless technologies (e.g., Wi-Fi) bears significantly fewer operational risk concerns when compared to sharing spectrum on the military bands (e.g., with radar systems). As a result, there have been significant activities on coexistence of cellular (LTE) and Wi-Fi on unlicensed bands from both industry [1], [2], [3], [4], [5] and academia [6], [7], [8], [9], [10], [11], [12], [13].

A key consideration in the design and operation of LTE in unlicensed band is to ensure fairness when they coexist with Wi-Fi. LTE was originally designed to work exclusively in operator-owned licensed bands. Its transmissions are centrally controlled and have no consideration for cross-technology coexistence [4]. In contrast, Wi-Fi employs CSMA/CA and is based on distributed contention. It can only transmit after the operating channel is clear and the lapse of its backoff period. Such incompatibility makes Wi-Fi highly vulnerable to the presence of LTE in the same band.

To address this issue, a number of mechanisms have been proposed for LTE in unlicensed band, such as Listen-Before-Talk (LBT) [3], [5] and Carrier-Sensing Adaptive Transmission (CSAT) [1], [2]. LBT is a random access approach similar to Wi-Fi's CSMA/CA, while CSAT is based on centralized scheduling, which is native to LTE's operation. With proper design, both CSAT and LBT can achieve fair spectrum sharing between LTE and Wi-Fi. Although CSAT may cause collisions to Wi-Fi's on-going packets, such impact can be mitigated by configuring longer duration for each LTE transmission burst [8]. CSAT is fully compatible with 3GPP Release 10/11 and does not require any change of LTE specifications [7]. It can be quickly launched in countries that do not mandate implementing LBT (e.g., the U.S. and China). Due to these benefits, operators such as T-Mobile have started supporting CSAT-based LTE-U in a number of U.S. cities [14].

In this paper, we employ the CSAT mechanism and study a scheduling problem for the coexistence of LTE and Wi-Fi in 5 GHz unlicensed bands. In the 5 GHz spectrum, there are multiple bands that can be used by LTE simultaneously. Under CSAT, the air time of each channel is divided into periodic LTE "on/off" cycles, where the "on" and "off" periods are used by LTE and Wi-Fi for channel access, respectively. Optimal division of "on" and "off" periods is determined by the LTE eNodeB (eNB) based on Wi-Fi's traffic load as measured from carrier sensing. Within LTE's "on" period of a channel, the bandwidth of the channel is expanded into a group of sub-channels and it is at this level that the so-called Resource Blocks (RBs) are allocated to LTE users. Suppose we have a different set of Wi-Fi users on each channel. To support a set of LTE users on these channels, where each user may have its own uplink (UL) and downlink (DL) rate requirements, the

Manuscript received November 13, 2018; revised March 3, 2019 and June 9, 2019; accepted November 21, 2019. Date of publication December 2, 2019; date of current version March 6, 2020. This research was supported in part by NSF under grants 1800650, 1642873, and 1617634. An abridged version of this paper appeared in IEEE DySPAN conference, October 22–25, 2018, Seoul, South Korea. The associate editor coordinating the review of this article and approving it for publication was S. Choi. *(Corresponding author: Y. Thomas Hou.)*

The authors are with Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: huangyan@vt.edu; yc.chen@vt.edu; thou@vt.edu; wjlou@vt.edu).

Digital Object Identifier 10.1109/TCCN.2019.2957076

problem becomes how to perform radio resource allocation to minimize LTE's impact on Wi-Fi while meeting various constraints and requirements. We will show that this scheduling problem for LTE/Wi-Fi coexistence is NP-hard, which means that it is impossible to obtain an optimal solution in real-time for a general network setting.

We formulate the above scheduling problem as an optimization problem. Due to NP-hardness, it cannot be solved efficiently. So it is necessary to pursue a heuristic solution that can achieve near-optimal objective. But the main challenge we need to address is to ensure the scheduling solution can be obtained in real-time — with a computational time of $\sim$1 ms. This timing requirement comes from the fact that channel coherence time in 5 GHz bands is at most tens of ms, meaning that a channel-dependent scheduling solution can remain valid only for tens of ms. If the computation time is beyond this time limit, the solution would not be considered good since channel conditions may have already changed considerably.

The goal of this paper is to develop a scheduler that can find a near-optimal solution in $\sim$1 ms under realistic LTE small cell scenarios so that LTE can use this solution within the coherence time period. We propose CURT, which arises from either the abbreviation of CSAT based Unlicensed LTE Real-Time resource scheduling (from coexistence scheme's perspective) or CUDA-based Real-Time resource scheduling (from implementation's perspective). We summarize the main contributions of CURT as follows:

- For LTE scheduling, we consider a wide range of parameters in our scheduling problem so as to best resemble what one would encounter in the field. These include (i) multiple channels available for LTE/Wi-Fi coexistence; (ii) both UL and DL rate requirements from LTE users; (iii) variation of channel conditions across subchannels. We formulate this scheduling problem into an optimization problem with the objective of minimizing the adverse impact on Wi-Fi while meeting LTE users' rate requirements. Further, we prove that the above scheduling problem for LTE/Wi-Fi coexistence is NP-hard.
- We present CURT, a novel GPU-based scheduler that can achieve near-optimal performance while meeting the stringent real-time constraint. In our design, by exploiting the unique problem structure, we decompose the original scheduling problem into a large number of independent sub-problems encompassing all possible cases of parameter settings. Then by performing a simple and fast evaluation of the feasibility of each sub-problem independently and in parallel with all other sub-problems through a novel use of massive GPU processing cores, we can determine a near-optimal (or optimal) solution among all the feasible solutions.
- To validate the performance of CURT, we implement it on off-the-shelf Nvidia Quadro P6000 GPUs in an integrated PC host-GPU architecture. Our implementation is based on meticulous considerations of GPU/CUDA architecture, mathematical structure of our proposed solution, and most importantly, the $\sim$1 ms constraint for overall scheduling time. Through extensive experimental study, we confirm that CURT can consistently find near-optimal scheduling solutions in $\sim$1 ms for LTE small cells with no more than 20 users (following 3GPP's evaluation methodology [3]) and meet all of our design objectives. This represents the first known CSAT-based scheduler design that can achieve real-time and near-optimal scheduling for coexistence between LTE and Wi-Fi.

The remainder of this paper is organized as follows. In Section II, we review related work on LTE/Wi-Fi coexistence on unlicensed bands. In Section III, we describe in detail the system architecture of CSAT-based LTE and state the underlying scheduling problem. In Section IV, we formulate the scheduling problem for LTE/Wi-Fi coexistence and prove its NP-hardness. In Section V, we present CURT, a real-time scheduler for LTE in unlicensed band. In Section VI, we show how CURT is implemented on off-the-shelf Nvidia Quadro P6000 GPUs. In Section VII, we conduct experiments to validate the performance of CURT. Section VIII concludes this paper.

II. Related Work

In the research community, there have been a number of studies on LTE/Wi-Fi coexistence, such as modeling and analysis of LTE's impact on Wi-Fi [9], [10], optimizations of coexistence mechanisms [6], [12], and radio resource management of LTE in unlicensed band [11], [13].

In [9], Abdelfattah *et al.* developed an analytical model for Wi-Fi's collision probability and throughput when coexisting with CSAT-based LTE. In [10], Voicu *et al.* proposed a general framework to evaluate the performance of multiple technologies operating in the same unlicensed bands. Both [9] and [10] focused on modeling of LTE/Wi-Fi coexistence and did not address allocation of radio resources.

In [12], the authors derived optimal division of LTE "on" and "off" periods under CSAT for a given number of LTE and Wi-Fi users. In [6], the authors considered joint optimization of channel selection and CSAT parameters. A fairness criterion was derived for LTE and Wi-Fi sharing multiple unlicensed channels. The criterion requires LTE not to impact Wi-Fi more than another Wi-Fi network with the same traffic load. The efforts in [12] and [6] addressed optimizations of CSAT parameters, but fell short to address resource management for LTE at the RB level.

In [11], Chen *et al.* addressed optimization of energy efficiency for CSAT-based LTE by studying RB allocation over licensed and unlicensed bands. The analysis in this work, however, did not consider channel fading effect on each individual RB, which is what will happen in practice. In [13], channel selection and per-frame RB scheduling were studied for coexistence between LTE and WLAN on multiple channels from unlicensed spectrum. An optimization problem was formulated with the objective of maximizing LTE's throughput while maintaining fairness between LTE and WLAN. The sequential algorithm proposed in this paper, although being polynomial-time, involves a large amount of iterative computations for scheduling in each frame. For both [11] and [13], it is not

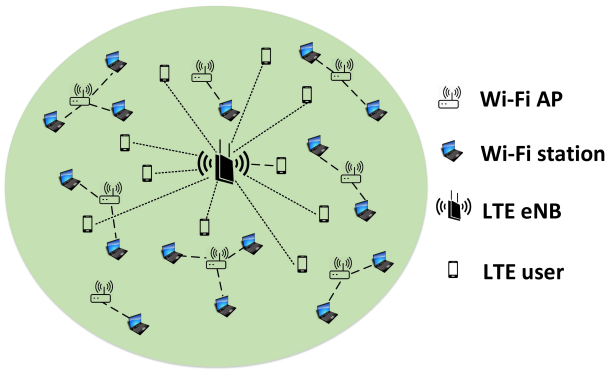


Fig. 1. Coexistence of LTE and Wi-Fi in an area.

clear if the proposed scheduling algorithms can meet real time requirement (i.e., $\sim$1 ms).

Employing GPU platform to address real-time resource scheduling for cellular networks (in licensed spectrum) has been studied in [15]. In that work, the authors considered the problem of joint RB allocation and MCS selection for each user in a licensed channel owned exclusively by a cellular operator, which is very different from the resource scheduling problem for LTE/Wi-Fi coexistence that is considered in this paper. In particular, the search space intensification and sub-problem selection techniques developed in [15] cannot be used to solve our problem in this paper.

TABLE I
NOTATION

Symbol	Definition
$\mathcal{F}$	A set of channels shared between LTE and Wi-Fi
$\mathcal{S}_i$	A set of sub-channels on channel i
(i,j)	The jth sub-channel in $\mathcal{S}_i$
$\mathcal{K}$	A set of LTE users offloaded to unlicensed bands
T_0	Duration of a TTI
T_{SF}	Duration of a scheduling frame
N_{SF}	The number of TTIs in a scheduling frame
M	The number of radio frames in a scheduling frame
I_i^{UL}	Binary variable indicating whether or not channel i is selected for UL transmission
I_i^{DL}	Binary variable indicating whether or not channel i is selected for DL transmission
$n_{(i,j)}^{k,\text{UL}}$	Integer variable denoting the amount of TRBs allocated to user k for UL transmission on sub-channel (i,j)
$n_{(i,j)}^{k,\text{DL}}$	Integer variable denoting the amount of TRBs allocated to user k for DL transmission on sub-channel (i,j)
$n_{i,\max}^{\text{UL}}$	The amount of TRBs reserved for LTE's UL transmission on channel i
$n_{i,\max}^{\text{DL}}$	The amount of TRBs reserved for LTE's DL transmission on channel i
$C_{(i,j)}^{k,\text{UL}}$	The UL achievable rate of user k on sub-channel (i,j)
$C_{(i,j)}^{k,\text{DL}}$	The DL achievable rate of user k on sub-channel (i,j)
$R^{k,\text{UL}}$	UL rate requirement of user k in unlicensed bands
$R^{k,\text{DL}}$	DL rate requirement of user k in unlicensed bands
Q_i	The maximum number of TTIs that can be used for LTE scheduling on channel i
U_i	The number of Wi-Fi nodes on channel i
w_i	The weight reflecting the Wi-Fi traffic load on channel i
z	Objective value in Problem OPT-R

III. System Architecture

Due to various stringent regional regulations on transmit power in unlicensed bands [3], it is envisioned that unlicensed LTE is only suitable for deployment under small-cell settings. We consider an LTE small cell overlapping with multiple Wi-Fi APs, as shown in Fig. 1. Table I lists notation in this paper. In Fig. 1, a set of LTE users $\mathcal{K}$ is served by a single LTE eNB while each Wi-Fi user is served by a nearby Wi-Fi AP.[1] Note that Fig. 1 only shows Wi-Fi nodes that fall in the LTE eNB's interference range. Potential Wi-Fi nodes outside the LTE eNB's interference range that are "hidden" from the eNB are not shown in this figure. The reason why we do not consider those hidden Wi-Fi nodes (outside the interference range of the eNB) is the following. Although those hidden Wi-Fi nodes may have adverse impact on Wi-Fi nodes that are inside the LTE eNB's interference range, they are not of concern of LTE scheduling and thus is not part of our problem formulation. Further, in practice, the LTE eNB has no mechanism to detect such hidden Wi-Fi nodes outside its interference range. Each LTE user $k \in \mathcal{K}$ has both UL and DL rate requirements in the unlicensed spectrum, which are denoted by $R^{k,\text{UL}}$ and $R^{k,\text{DL}}$, respectively. The rate requirements should be configured dynamically by a traffic management mechanism on unlicensed bands as described in Section V-E.

Suppose there is a number of channels in the unlicensed band that can be used by both LTE and Wi-Fi networks. Due to dense Wi-Fi deployment, there may not be enough clear channels for LTE and thus LTE has to coexist with Wi-Fi on some of these channels. In this paper, we focus on this subset of channels, denoted by $\mathcal{F}$, where both LTE and Wi-Fi are present. For LTE, its transmission scheduling is centrally controlled by the eNB, and it can combine multiple channels for UL and DL transmissions via FDD carrier aggregation (CA).[2] Every channel $i \in \mathcal{F}$ (used for either UL or DL) is further divided into a set of sub-channels $\mathcal{S}_i$. Thus the frequency granularity of LTE is on sub-channel level. In contrast, for Wi-Fi, the frequency granularity is on the channel level, where an AP or station typically occupies the entire bandwidth of a channel (instead of a sub-channel) for DL or UL data transmission. Denote U_i as the number of Wi-Fi nodes on channel $i \in \mathcal{F}$.

CSAT-based LTE Scheduling: We employ CSAT for LTE scheduling [2]. As shown in Fig. 2, CSAT is a time division multiplexing (TDM)-like channel access mechanism where each CSAT cycle (a.k.a scheduling frame) consists of an LTE "off" and "on" period. During the "off" period, LTE transmission is suspended so that co-channel Wi-Fi nodes can access the medium. Once the "off" period is over, LTE network starts transmission regardless of channel status (even if there is a collision). Since Wi-Fi senses the channel before a new transmission, it will cease to transmit during LTE's "on" period. In a CSAT cycle, the division of "off" and "on" periods on a channel is part of the scheduling problem. Through carrier sensing, the eNB measures Wi-Fi traffic load on each channel,

[1]The set $\mathcal{K}$ of LTE users that are offloaded to unlicensed bands is determined by the carrier load balancing function defined in the LTE specification [16]. How set $\mathcal{K}$ is determined is beyond the scope of this paper.

[2]Under CA, only up to 5 carriers are possible [17]. Although there are more than 10 channels in 2.4 and 5 GHz bands, LTE can only select no more than 5 channels for coexistence. Thus we have $|\mathcal{F}| \leq 5$.

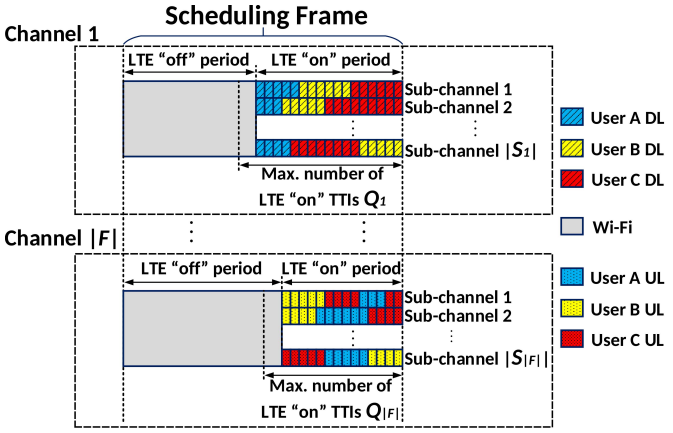


Fig. 2. CSAT-based scheduling.

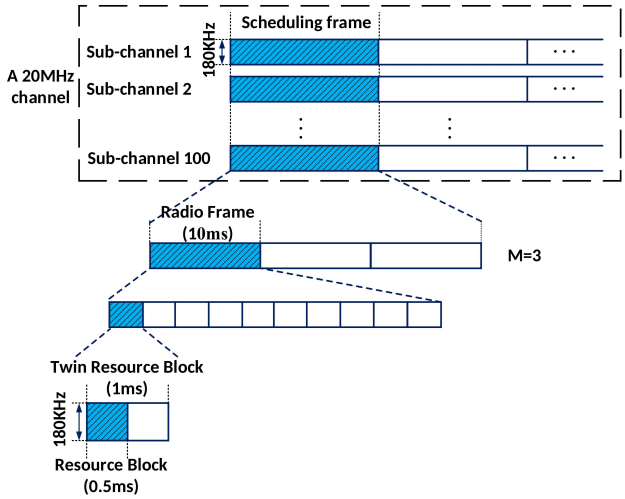


Fig. 3. Radio resource arrangement in LTE and setting of scheduling frame.

and uses it as input to determine "off" and "on" periods on this channel.

Radio Resource Arrangement in LTE: An illustration of LTE's radio resource arrangement is given in Fig. 3. Specifically, on each channel, radio resource is organized as a two-dimensional resource grid [4]. In frequency domain, the channel is divided into a set of sub-channels, each with a bandwidth of 180 kHz. In time domain, we have consecutive *radio frames*, each with a duration of 10 ms. A radio frame consists of 10 sub-frames. The duration of a sub-frame is 1 ms, which is termed a *Transmission Time Interval* (TTI). A TTI is further divided into two time slots, each with a duration of 0.5 ms. A resource block (RB) is defined as a time-frequency resource unit with 180 kHz in frequency (a sub-channel) and 0.5 ms in time (a time slot). The time resolution for LTE scheduling is two consecutive RBs in a sub-frame, which we call a *Twin RBs* (TRB). Since each TRB is of 1 ms, a radio frame consists of 10 TRBs.

Scheduling Frame and Coherence Time: We define a scheduling frame (SF) as a consecutive M radio frames (refer to Fig. 3). Since a radio frame is 10 ms, the duration of a SF, denoted by T_{SF}, is equal to 10 M ms.

The maximum number of radio frames that can be packed into a scheduling frame, M, is upper limited by the coherence time of the underlying channel. That is, M should be small enough so that there is no significant change of LTE users' channel conditions (as well as their achievable data rates) over a period of T_{SF}. As an example, consider the 5 GHz spectrum for an indoor deployment scenario. The channel coherence time T_{C} can be calculated by $T_{\mathrm{C}} = \sqrt{\frac{9}{16\pi f_{\mathrm{M}}^2}}$ [20], where $f_{\mathrm{M}} = v/\lambda$ denotes the maximum Doppler shift, v is the user speed, and λ is the carrier wavelength. In an indoor small cell, assuming a user speed of 3 km/h [3], the coherence time on 5 GHz spectrum is $T_{\mathrm{C}} = 30.58$ ms. Therefore, the maximum value M can take is 3 ($\leq 30.58/10$). That is, $T_{\mathrm{SF}} = 30$ ms.

Denote the number of TTIs in a SF by N_{SF}. By definition, we have $N_{\mathrm{SF}} = \frac{T_{\mathrm{SF}}}{T_0} = 10M$, where $T_0 = 1$ ms denotes the duration of a TTI. Within a CSAT "on/off" cycle, we will have an integral number of TTIs for both "on" and "off" periods. Further, we assume perfect time synchronization so that the boundaries of TTIs and SFs across all channels occupied by LTE are perfectly aligned.

Problem Statement: We are interested in addressing the following problem: Given that a set $\mathcal{K}$ of LTE users are to coexist with Wi-Fi, how do we minimize LTE traffic's adverse impact on Wi-Fi users while meeting each LTE user's UL and DL rate requirements? To answer this question, we must address the following sub-problems:

- (i) For LTE, since each user has both UL and DL data traffic, we must decide how to use each channel in $\mathcal{F}$. That is, should a channel $i \in \mathcal{F}$ be used for UL or DL transmission?
- (ii) For Wi-Fi/LTE coexistence on each channel $i \in \mathcal{F}$, we must decide the durations of "off" and "on" periods within each SF. The "on" period directly translates into adverse impact on Wi-Fi users. Our objective is to divide "off" and "on" periods on each channel optimally to minimize such adverse impact across all channels.
- (iii) To meet each LTE user's UL and DL rate requirements, we need to allocate TRBs on each sub-channel to users. A user's rate requirements can be fulfilled by allocating TRBs from multiple channels. This is not trivial because the achievable data rate of a user varies on different sub-channels, due to frequency-selective channel fading.
- (iv) Last but perhaps most significant is that we are interested in a *real-time* scheduling algorithm. By real-time, we mean that the LTE scheduling solution must be found within the "off" periods of the SFs (more precisely, the smallest "off" period across all channels in $\mathcal{F}$). This will ensure that LTE users can follow the pre-computed, optimized transmission schedule in "on" periods on all channels. Given that T_{SF} is typically several 10 s of ms and optimal "off" periods may be less than 10 ms, the scheduling time must be within a few ms. In this paper, we use 1 ms as our target scheduling time.

IV. Mathematical Modeling

In this section, we develop a mathematical model for the resource scheduling problem for LTE/Wi-Fi coexistence.

UL/DL Channel Assignment: Referring to Fig. 2, consider the set of channels $\mathcal{F}$ where each channel is shared between LTE and Wi-Fi users. For LTE, denote I_i^{UL} and I_i^{DL} as binary variables to indicate whether channel $i \in \mathcal{F}$ is used for UL and DL transmissions, respectively, i.e.,

$$I_i^{\mathrm{UL}} = \begin{cases} 1, & \text{if channel } i \in \mathcal{F} \text{ is selected for UL;} \\ 0, & \text{otherwise.} \end{cases}$$

$$I_i^{\mathrm{DL}} = \begin{cases} 1, & \text{if channel } i \in \mathcal{F} \text{ is selected for DL;} \\ 0, & \text{otherwise.} \end{cases}$$

Since each channel can only be used by LTE for either UL or DL transmission, but not both, we have:

$$I_i^{\mathrm{UL}} + I_i^{\mathrm{DL}} \leq 1 \quad (i \in \mathcal{F}). \tag{1}$$

Effective Occupancy by LTE on A Channel: Referring to Fig. 2, for each channel $i \in \mathcal{F}$, there is a set of sub-channels $\mathcal{S}_i$. Scheduling for LTE is performed on sub-channel level. On each sub-channel of channel *i*, LTE's transmission time (either UL or DL) may not terminate at the same time. Denote $(i,j) \in \mathcal{S}_i$ as sub-channel *j* on channel *i*. Then as far as Wi-Fi is concerned, channel *i* is available only if LTE ceases transmissions on *all* sub-channels.

To model this effective channel occupancy by LTE, denote $n_{(i,j)}^{k,\mathrm{UL}}$ and $n_{(i,j)}^{k,\mathrm{DL}}$ as the number of TRBs on sub-channel $(i,j) \in \mathcal{S}_i$ within a SF that are allocated to user $k \in \mathcal{K}$ for UL and DL transmissions, respectively. If channel *i* is selected for UL transmission (i.e., $I_i^{\mathrm{UL}} = 1$), then LTE's usage of TTIs on sub-channel (*i,j*) across all users in $\mathcal{K}$ is $\sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{UL}}$. Denote $n_{i,\max}^{\mathrm{UL}}$ as the effective channel occupancy by LTE on channel *i* across all $|\mathcal{S}_i|$ sub-channels. Then,

$$n_{i,\max}^{\mathrm{UL}} = \max_{j\in\mathcal{S}_i} \sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{UL}} \quad (i \in \mathcal{F}). \tag{2}$$

Likewise, if channel *i* is selected for DL transmission (i.e., $I_i^{\mathrm{DL}} = 1$), then LTE's usage of TTIs on sub-channel (*i, j*) across all users in $\mathcal{K}$ is $\sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{DL}}$. Denote $n_{i,\max}^{\mathrm{DL}}$ as the effective channel occupancy by LTE on channel *i*. We have:

$$n_{i,\max}^{\mathrm{DL}} = \max_{j\in\mathcal{S}_i} \sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{DL}} \quad (i \in \mathcal{F}). \tag{3}$$

Within a SF on channel *i*, the usable time duration (in unit of TTIs) for LTE is determined by $n_{i,\max}^{\mathrm{UL}}$ (if $I_i^{\mathrm{UL}} = 1$) or $n_{i,\max}^{\mathrm{DL}}$ (if $I_i^{\mathrm{DL}} = 1$). While the time duration left for Wi-Fi is $N_{\mathrm{SF}} - n_{i,\max}^{\mathrm{UL}}$ (for UL) or $N_{\mathrm{SF}} - n_{i,\max}^{\mathrm{DL}}$ (for DL) TTIs.

Upper Bound on LTE Usage: To ensure that LTE does not monopolize radio resource of each channel $i \in \mathcal{F}$, it is important to set up an upper bound on LTE's transmission time for its "on" period on each channel [3]. Let Q_i ($Q_i < N_{\mathrm{SF}}$) denote the upper bound on the number of TTIs that LTE can use for UL or DL transmission on channel *i* within a SF. Then

$$n_{i,\max}^{\mathrm{UL}} \leq I_i^{\mathrm{UL}} Q_i \quad (i \in \mathcal{F}), \tag{4}$$

$$n_{i,\max}^{\mathrm{DL}} \leq I_i^{\mathrm{DL}} Q_i \quad (i \in \mathcal{F}). \tag{5}$$

The setting of Q_i's depends on the fairness criterion used for LTE/Wi-Fi coexistence. For example, a popular fairness criterion is that on each channel, LTE should not impact Wi-Fi more than another Wi-Fi network with the same traffic load [3], [6]. When we assume persistent (i.e., infinitely buffered) traffic for both LTE and Wi-Fi users, it is reasonable to set $Q_i = \lfloor N_{\mathrm{SF}} \frac{|\mathcal{K}|/|\mathcal{F}|}{|\mathcal{K}|/|\mathcal{F}|+U_i} \rfloor$ following the spirit of this criterion, where $|\mathcal{K}|/|\mathcal{F}|$ is the number of LTE users per channel and U_i is the number of Wi-Fi nodes on channel *i*. Basically, this setting allocates transmission time to LTE and Wi-Fi in proportion to the number of nodes per channel in each network. In [6], the authors proposed to optimize Q_i's based on Wi-Fi's actual perceived interference from LTE. This approach is theoretically interesting but problematic in practice, as it would require information of path loss and channel fading coefficients between LTE and Wi-Fi nodes. But such information is not available in practice because there is no channel training and estimation mechanisms between LTE and Wi-Fi. In this regard, our proposed setting of Q_i's is both simple and feasible.

We stress that one could employ an entirely different setting for Q_i's to achieve her own "fairness" objective. In this regard, Q_i's are just input parameters to the scheduling problem. How Q_i's are set will not affect the solution design and our proposed solution algorithm works under any setting of Q_i's. In fact, one can even tune the setting of Q_i's per SF if it is deemed necessary.

Meeting LTE User Rate Requirement: For each user $k \in \mathcal{K}$, to ensure both of its UL and DL rate requirements are met, we have the following constraints:

$$R^{k,\mathrm{UL}} \leq \frac{\sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{S}_i} n_{(i,j)}^{k,\mathrm{UL}} C_{(i,j)}^{k,\mathrm{UL}} T_0}{T_{\mathrm{SF}}} \quad (k \in \mathcal{K}), \tag{6}$$

$$R^{k,\mathrm{DL}} \leq \frac{\sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{S}_i} n_{(i,j)}^{k,\mathrm{DL}} C_{(i,j)}^{k,\mathrm{DL}} T_0}{T_{\mathrm{SF}}} \quad (k \in \mathcal{K}), \tag{7}$$

where $C_{(i,j)}^{k,\mathrm{UL}}$ and $C_{(i,j)}^{k,\mathrm{DL}}$ are UL and DL achievable data rates for user *k* on sub-channel (*i, j*), respectively. The data rates $C_{(i,j)}^{k,\mathrm{UL}}$'s and $C_{(i,j)}^{k,\mathrm{DL}}$'s are obtained based on users' CSI reports [5]. This is how it is done in real-world FDD LTE systems. During the "on" period when LTE is transmitting, if there is any interference from Wi-Fi, then such interference will be captured in the CSI reports as well as the estimated parameters $C_{(i,j)}^{k,\mathrm{UL}}$'s and $C_{(i,j)}^{k,\mathrm{DL}}$'s. That is, $C_{(i,j)}^{k,\mathrm{UL}}$'s and $C_{(i,j)}^{k,\mathrm{DL}}$'s are obtained via channel estimation and have already considered interference from Wi-Fi, if there is any. By the definition of SF in Section III, $C_{(i,j)}^{k,\mathrm{UL}}$'s and $C_{(i,j)}^{k,\mathrm{DL}}$'s remain constant during each SF.

Objective Function and Problem Formulation: In a SF, the transmission time of LTE on channel $i \in \mathcal{F}$ is determined by $I_i^{\mathrm{UL}} \cdot n_{i,\max}^{\mathrm{UL}} + I_i^{\mathrm{DL}} \cdot n_{i,\max}^{\mathrm{DL}}$. For the same transmission time duration, LTE's impact on Wi-Fi depends on the traffic load of Wi-Fi. The heavier traffic that is being served by Wi-Fi on the same channel, the greater the impact of LTE on Wi-Fi. To take this into consideration, we introduce a weight parameter w_i to reflect Wi-Fi's traffic load on channel $i \in \mathcal{F}$. A simple example is to set w_i to the number of Wi-Fi nodes

on channel i, i.e., $w_i = U_i$. To find U_i on each channel, the eNB can monitor Wi-Fi's channel usage during "off" periods, which are allocated to Wi-Fi transmission. For example, with the methods proposed in [18], [19], U_i can be determined online based on the proportion of observed busy time slots. This is feasible since an LTE eNB using unlicensed band is expected to be able to perform carrier sensing [1]. For a given w_i, the impact of LTE on Wi-Fi on channel i can be quantitatively measured by $w_i(I_i^{\mathrm{UL}} \cdot n_{i,\max}^{\mathrm{UL}} + I_i^{\mathrm{DL}} \cdot n_{i,\max}^{\mathrm{DL}})$. Note that the weight w_i is the same for uplink and downlink because regardless of the direction (uplink or downlink) in which LTE transmits, the worst-case lost transmission time for Wi-Fi is equal to the duration of LTE's "on" period.

The reason why we use the duration of LTE "on" period weighted by the number of Wi-Fi nodes on the channel to model the impact of LTE on Wi-Fi is as follows. In practice, there is no channel training and coordination mechanism between LTE and Wi-Fi. As a result, the centralized LTE scheduler (located at eNB) has no information on channel statistics of radio links to Wi-Fi (e.g., path loss and fast-fading coefficients), the received interference power level at Wi-Fi, or the traffic condition of Wi-Fi. Thus in LTE scheduling optimization, one cannot assume such information is available. To have a reasonable model of the impact of LTE on Wi-Fi, we use the length of LTE's "on" period in each scheduling frame, during which Wi-Fi nodes are not expected to transmit. Although Wi-Fi may transmit opportunistically when fast deep fading occurs on the channel during "on" period, such information is not available at the LTE scheduler. Therefore, the "on" period is actually the worst-case loss of transmission period for Wi-Fi. In addition, during an "on" period, the impact of LTE on Wi-Fi depends on the traffic load of Wi-Fi on the same channel, i.e., the higher the Wi-Fi traffic load, the more severe the impact. As it is impossible for the LTE scheduler to know the actual Wi-Fi traffic load, one can only use the number of active Wi-Fi nodes through sensing the channel as a load indicator, assuming the traffic at Wi-Fi nodes is persistent. Previous work has shown the feasibility of obtaining such information through carrier sensing methods [18], [19].

Since LTE's impact on Wi-Fi varies from channel to channel, a plausible objective for the network operator is to minimize the maximum of LTE's impact across all channels, i.e.,

$$\min \max_{i\in\mathcal{F}} w_i\Big(I_i^{\mathrm{UL}} \cdot n_{i,\max}^{\mathrm{UL}} + I_i^{\mathrm{DL}} \cdot n_{i,\max}^{\mathrm{DL}}\Big). \tag{8}$$

This is the objective we use in this paper.

Our optimization problem can be formally stated as follows:

OPT

$$\begin{aligned}
\text{minimize}\quad & \max_{i\in\mathcal{F}} w_i\Big(I_i^{\mathrm{UL}} \cdot n_{i,\max}^{\mathrm{UL}} + I_i^{\mathrm{DL}} \cdot n_{i,\max}^{\mathrm{DL}}\Big)\\
\text{subject to}\quad & \text{UL/DL channel assignment: (1)},\\
& \text{Effective channel occupancy by LTE: (2), (3)},\\
& \text{Upper bound on LTE usage: (4), (5)},\\
& \text{Per-user rate requirement: (6), (7)},\\
& n_{i,\max}^{\mathrm{UL}}, n_{i,\max}^{\mathrm{DL}}, n_{(i,j)}^{k,\mathrm{UL}}, n_{(i,j)}^{k,\mathrm{DL}} \in \mathbb{N},\\
& I_i^{\mathrm{UL}}, I_i^{\mathrm{DL}} \in \{0,1\} \quad (i\in\mathcal{F}, j\in\mathcal{S}_i, k\in\mathcal{K}).
\end{aligned}$$

The solution to OPT determines the allocation of resources for LTE and Wi-Fi within an entire SF (CSAT cycle). When implementing the solution in an LTE small cell, resource allocated to LTE users can be delivered in the same per-TTI manner as in licensed band operation. Specifically, the solution for an entire SF is stored in the eNB, and in each TTI for LTE's transmission, the eNB informs users of their resource allocation for the current TTI via control channel signaling.

A Reformulation: In Problem OPT, since the objective function involves integer variables and two levels of max functions (due to (2) and (3)), a reformulation would be needed. In particular, in the presence of constraints (4) and (5), the objective function can be simplified to $\max_{i\in\mathcal{F}} w_i(n_{i,\max}^{\mathrm{UL}} + n_{i,\max}^{\mathrm{DL}})$. To remove the two levels of max functions, we define $z = \max_{i\in\mathcal{F}} w_i(n_{i,\max}^{\mathrm{UL}} + n_{i,\max}^{\mathrm{DL}})$ as the new objective function. Then we have the following constraint:

$$z \geq w_i\Big(n_{i,\max}^{\mathrm{UL}} + n_{i,\max}^{\mathrm{DL}}\Big) \quad (i\in\mathcal{F}). \tag{9}$$

By constraint (1), at most one of the two terms, $n_{i,\max}^{\mathrm{UL}}$ and $n_{i,\max}^{\mathrm{DL}}$, can be nonzero. Then (9) can be reformulated to $z \geq w_i \cdot n_{i,\max}^{\mathrm{UL}}$ and $z \geq w_i \cdot n_{i,\max}^{\mathrm{DL}}$ for $i\in\mathcal{F}$. By definitions of $n_{i,\max}^{\mathrm{UL}}$ and $n_{i,\max}^{\mathrm{DL}}$ in (2) and (3), we have:

$$n_{i,\max}^{\mathrm{UL}} \geq \sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{UL}} \quad (i\in\mathcal{F}, j\in\mathcal{S}_i),$$

$$n_{i,\max}^{\mathrm{DL}} \geq \sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{DL}} \quad (i\in\mathcal{F}, j\in\mathcal{S}_i).$$

Therefore, we have the following constraints on z:

$$z \geq w_i \sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{UL}} \quad (i\in\mathcal{F}, j\in\mathcal{S}_i), \tag{10}$$

$$z \geq w_i \sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{DL}} \quad (i\in\mathcal{F}, j\in\mathcal{S}_i). \tag{11}$$

The constraints in (2), (3), (4) and (5) can be simplified by eliminating $n_{i,\max}^{\mathrm{UL}}$ and $n_{i,\max}^{\mathrm{DL}}$ and removing the max functions. We have:

$$\sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{UL}} \leq I_i^{\mathrm{UL}} Q_i \quad (i\in\mathcal{F}, j\in\mathcal{S}_i), \tag{12}$$

$$\sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{DL}} \leq I_i^{\mathrm{DL}} Q_i \quad (i\in\mathcal{F}, j\in\mathcal{S}_i). \tag{13}$$

Finally we have the reformulated optimization problem:

OPT-R

$$\begin{aligned}
\text{minimize}\quad & z\\
\text{subject to}\quad & \text{Adverse impact of LTE on Wi-Fi: (10), (11)},\\
& \text{UL/DL channel assignment: (1)},\\
& \text{Upper bound on LTE usage: (12), (13)},\\
& \text{Per-user rate requirement: (6), (7)},
\end{aligned}$$

$$z \geq 0,\ n_{(i,j)}^{k,\mathrm{UL}}, n_{(i,j)}^{k,\mathrm{DL}} \in \mathbb{N},$$
$$I_i^{\mathrm{UL}}, I_i^{\mathrm{DL}} \in \{0,1\} \quad (i \in \mathcal{F}, j \in \mathcal{S}_i, k \in \mathcal{K}).$$

Problem OPT-R is a mixed integer linear program (MILP), one of the most common types of problems for wireless network optimization. Although commercial solvers such as the IBM CPLEX [28] can be employed to compute its optimal solution off-line (useful for benchmark purpose), they cannot meet the stringent real-time scheduling requirement ($\sim$1 ms).

Problem Complexity: We have the following result for the computational complexity of our scheduling problem for LTE/Wi-Fi coexistence.

Lemma 1: The scheduling problem for LTE/Wi-Fi coexistence (i.e., minimization of objective (8) *under the constraints of LTE users' rate requirements and fair LTE/Wi-Fi coexistence on each channel) is NP-hard.*

A proof of Lemma 1 is given in the Appendix.

The NP-hardness result suggests that there is no efficient polynomial-time algorithm to solve the problem exactly. In OPT-R, the numbers of variables and constraints are both on the order of $O(|\mathcal{F}||\mathcal{S}_i||\mathcal{K}|)$. For instance, suppose $|\mathcal{F}| = 5$, $|\mathcal{S}_i| = 100$ (20 MHz channel bandwidth), $|\mathcal{K}| = 20$ and $T_{\mathrm{SF}} = 30$ ms. The searching space in OPT-R involves 20011 variables and 22056 constraints. The state-of-the-art solver CPLEX (based on BB-CP) would take 10s of seconds to hours (refer to Section VII) to get an optimal or lower-bound solution.

V. CURT – A Novel Real-Time Scheduler

Before presenting the design of CURT, we first explain why conventional methods fail to meet our target of providing near-optimal solution to OPT in real-time. We consider the following approaches: (i) reusing LTE schedulers designed for licensed bands, (ii) solving linear programming (LP) relaxation of OPT-R and rounding up the solution to integers, and (iii) using an exact algorithm such as BB to solve OPT-R directly.

Existing LTE schedulers designed for licensed bands are typically metric-based algorithms that allocate TRBs on a per-TTI basis [21]. Specifically, in every TTI, each TRB (or group of TRBs) is allocated to the user that has the highest metric associated with it (e.g., achievable rate, rate requirement, delay, or fairness index). These schedulers, although meeting their real-time requirements, cannot be readily extended to solve problem OPT. This is because OPT has an objective of minimizing LTE's impact on Wi-Fi across multiple channels while meeting LTE users' traffic requirements and constraints. This problem formulation is completely different from those modeled for licensed bands with the objective of maximizing spectral efficiency. Those schedulers have no consideration of the impact of LTE transmission on Wi-Fi and do not address how to optimally divide each CSAT cycle into "on/off" periods across multiple channels. Thus they cannot be used to solve the scheduling problem in this paper.

Standard optimization techniques such as LP relaxation and BB cannot meet the real-time requirement of $\sim$1 ms. Although an LP relaxation of OPT-R can be solved efficiently by Simplex or interior-point methods, the computation time is much larger than 1 ms. BB can be used to solve an MILP such as OPT-R. The basic idea of a BB-based approach is to find upper and lower bounds of each sub-problem as well as the global upper and lower bounds across all sub-problems at each iteration. The gap between upper and lower bounds is expected to shrink after each iteration until it is within the desired optimality gap. The main problem with such an approach is that the overall running time to close the gap is too long to meet our real-time requirement.

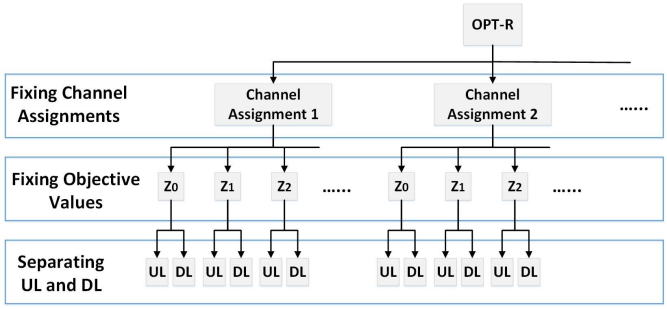


Fig. 4. Decomposition of OPT-R.

A. An Overview of CURT

Different from existing LTE schedulers used on licensed bands, CURT jointly addresses time division between LTE and Wi-Fi across multiple channels and TRB allocation within LTE "on" periods, with the target of getting optimal or near-optimal solution to OPT-R in real-time ($\sim$1 ms). The design of CURT is based on problem decomposition and parallel execution of sub-problems on a massive number of GPU cores. To pursue near-optimality, CURT first decomposes OPT-R into a large number of sub-problems, each with a fixed assignment of UL/DL channels and "off/on" time division pattern across all channels, and then solves all sub-problems in parallel by GPU cores. In particular, our proposed decomposition ensures that there is no inter-dependency among sub-problems, so that all sub-problems can be executed independetly and in parallel. Thus the time complexity of solving OPT-R is reduced to that of solving one sub-problem. Further, since all sub-problems are of the same structure, they can be solved within a close-to-same amount of time. The parallel design of CURT ensures that all possible cases of LTE's impact on Wi-Fi can be evaluated (for feasibility) via a simple and fast algorithm in parallel. By comparing among the objectives of all feasible solutions, CURT has a high probability to find a near-optimal solution.

B. Problem Decomposition

Figure 4 shows how we decompose OPT-R into small independent sub-problems. There are three levels of decomposition: 1) fixing UL and DL channel assignments, 2) fixing achievable objective values, and 3) separating UL and DL sub-problems. Details of these steps are as follows.

Fixing UL/DL Channel Assignments: Given a set of channels $\mathcal{F}$ for LTE/Wi-Fi coexistence, there is a total of $\binom{|\mathcal{F}|}{1} +$

$\binom{|\mathcal{F}|}{2}+\cdots+\binom{|\mathcal{F}|}{|\mathcal{F}|-1}=2^{|\mathcal{F}|}-2$ different ways for assigning the channels for UL and DL transmissions. In practice, $|\mathcal{F}|$ is not a large number due to limitations on signal processing capability at LTE eNBs (small cell access points) and users. For example, when $|\mathcal{F}|=5$, there is a total of 30 different channel assignments. Problem OPT-R can thus be decomposed into a set of $(2^{|\mathcal{F}|}-2)$ sub-problems, each with a given UL/DL channel assignment. For each sub-problem, we need to find the smallest objective value that is achievable under the given channel assignment.

Fixing Achievable Objective Values: From constraints (10) and (11), it is clear that the range of objective value of OPT-R under a given UL/DL channel assignment contains only a finite number of values. Specifically, since both $w_i\sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{UL}}$ and $w_i\sum_{k\in\mathcal{K}} n_{(i,j)}^{k,\mathrm{DL}}$ (for all $j\in\mathcal{S}_i$ on channel $i\in\mathcal{F}$) can only take values from $\mathcal{Z}_i=\{0,w_i,2w_i,\ldots,Q_iw_i\}$, the optimal objective value z^* must be within the set

$$\mathcal{Z}=\bigcup_{i\in\mathcal{F}}\mathcal{Z}_i. \tag{14}$$

Since each set $\mathcal{Z}_i$ $(i\in\mathcal{F})$ consists of 0 and Q_i nonzero elements (if $Q_i>0$), we have

$$|\mathcal{Z}|\le 1+\sum_{i\in\mathcal{F}}Q_i\le|\mathcal{F}|\cdot N_{\mathrm{SF}}, \tag{15}$$

where the second inequality follows from the definition $Q_i<N_{\mathrm{SF}}$ for all $i\in\mathcal{F}$.

By fixing objective value, we further decompose each sub-problem under a specific channel assignment into $|\mathcal{Z}|$ sub-problems. For each resulting sub-problem, we need to determine whether or not a given objective value in $\mathcal{Z}$ is achievable (i.e., feasibility) under the channel assignment. After this decomposition, the $|\mathcal{Z}|$ sub-problems under a given channel assignment include all possible "off/on" time division patterns across channels in $\mathcal{F}$.

Separating UL/DL Sub-Problems: For each sub-problem under a given channel assignment and objective value, we need to check whether or not it is feasible to meet all users' UL and DL rate requirements. This feasibility check can again be decomposed into two parallel problems, one for UL and the other for DL.

Now the original problem OPT-R is decomposed into a total of $2(2^{|\mathcal{F}|}-2)\cdot|\mathcal{Z}|$ UL/DL sub-problems for feasibility check (each with a given channel assignment and objective value). We propose to employ low-cost off-the-shelf GPU (each consisting of a massive number of cores) to solve them in parallel. Once feasibility checks for all UL/DL sub-problems are completed, we pick the smallest feasible objective value under all UL/DL channel assignments that has both its UL and DL sub-problems pass feasibility checks (both are feasible). The scheduling solution corresponding to this objective value and channel assignment is our output solution. Details about how these operations are implemented on GPU are given in Section VI.

Algorithm 1 Feasibility Check of DL Sub-Problem

Input the set of DL channels $\mathcal{F}^{\mathrm{DL}}$ and the objective value z;
Initialize:
  1) $V_{\mathrm{res}}^{k,\mathrm{DL}}:=R^{k,\mathrm{DL}}T_{\mathrm{SF}}$ for each $k\in\mathcal{K}$;
  2) Q_i' as in (16) for each DL channel $i\in\mathcal{F}^{\mathrm{DL}}$;
  3) Feasibility: = False;
while ($\mathcal{F}^{\mathrm{DL}}\neq\emptyset$ **and** Feasibility = False) **do**
  Choose any remaining channel in $\mathcal{F}^{\mathrm{DL}}$ and denote it as channel *i*;
  while ($\mathcal{S}_i\neq\emptyset$ **and** Feasibility = False) **do**
    Choose any remaining sub-channel in $\mathcal{S}_i$ and denote it as (*i, j*);
    $Q'_{(i,j)}:=Q_i'$;
    while ($Q'_{(i,j)}>0$ **and** Feasibility = False) **do**
      Find the user $k':=\arg\max_{k\in\mathcal{K}} C_{(i,j)}^{k,\mathrm{DL}}\cdot V_{\mathrm{res}}^{k,\mathrm{DL}}$;
      Set $n_{(i,j)}^{k',\mathrm{DL}}:=\min\left\{Q_i',\left\lceil\frac{V_{\mathrm{res}}^{k',\mathrm{DL}}}{C_{(i,j)}^{k',\mathrm{DL}}T_0}\right\rceil\right\}$;
      Update $V_{\mathrm{res}}^{k',\mathrm{DL}}:=V_{\mathrm{res}}^{k',\mathrm{DL}}-n_{(i,j)}^{k',\mathrm{DL}}C_{(i,j)}^{k',\mathrm{DL}}T_0$
      and $Q'_{(i,j)}:=Q'_{(i,j)}-n_{(i,j)}^{k',\mathrm{DL}}$;
      if ($V_{\mathrm{res}}^{k',\mathrm{DL}}\le 0$) **then**
        $\mathcal{K}:=\mathcal{K}\setminus\{k'\}$;
        if ($\mathcal{K}=\emptyset$) **then**
          Feasibility: = True;
    end while
    $\mathcal{S}_i:=\mathcal{S}_i\setminus\{(i,j)\}$;
  end while
  $\mathcal{F}^{\mathrm{DL}}:=\mathcal{F}^{\mathrm{DL}}\setminus\{i\}$;
end while
return Feasibility;

C. Feasibility Check of Sub-Problems

In the feasibility check of a UL/DL sub-problem, we aim to determine whether or not each user's UL/DL rate requirement can be met under the given UL/DL channel assignment. This problem can be formulated as an integer linear program (ILP), which is NP-hard and cannot be solved exactly under our tight time constraint ($\sim$1 ms). So a fast and efficient heuristic algorithm is needed. In this section, we present the design for feasibility check of DL sub-problems. The case for UL sub-problems is similar and is omitted to conserve space.

For each DL channel *i*, there are $|\mathcal{S}_i|$ sub-channels on it and we consider one sub-channel at a time to fill users' rate requirements. The order for selecting channels and sub-channels is arbitrary. For a given sub-channel, we use it to fill one or more user's rate requirement. The question is: Which user (among the users whose rate requirements have not been met) should we consider? This is a user selection problem. Note that the sub-channel capacity of each user differs. That is, one user may find the sub-channel to be of good condition while another user may find it otherwise. So first we need to find each user's sub-channel capacity.

The next question is: should sub-channel capacity be the *only* criterion in user selection? The answer is *No.* This is because a user experiencing low capacity on this sub-channel may experience a even lower capacity on other sub-channels. If we do not consider this user on the given sub-channel, it will consume even more TRBs on other sub-channels.

Taking the above two considerations together, we propose a user selection criterion that selects one user (among the remaining users whose rate requirements have not been met)

who has the largest sub-channel capacity weighted by its remaining work (rate to be filled).

Our proposed feasibility check algorithm for the DL sub-problem is given in Algorithm 1. The input is a given set $\mathcal{F}^{\mathrm{DL}}$ of DL channels and an objective value z. For convenience, we introduce a new notation $V_{\mathrm{res}}^{k,\mathrm{DL}}$, $k \in \mathcal{K}$, which represents the remaining DL data (in bits) that should be scheduled for user k within a SF to meet its DL rate requirement. Initially, we have $V_{\mathrm{res}}^{k,\mathrm{DL}} = R^{k,\mathrm{DL}} T_{\mathrm{SF}}$.

Denote Q_i' as the number of TRBs that are available to LTE on each sub-channel $(i,j) \in \mathcal{S}_i$ under the objective value z. Then Q_i' is upper bounded by Q_i. Further, for given z, by (11), Q_i' is also upper bounded by $\lfloor z/w_i \rfloor$. We have:

$$Q_i' = \begin{cases} \min\left\{ Q_i, \lfloor \frac{z}{w_i} \rfloor \right\}, & \text{for } w_i > 0, \\ Q_i, & \text{for } w_i = 0. \end{cases} \tag{16}$$

The main body of Algorithm 1 consists of three "while" loops, with the two outer while loops enumerating all remaining DL channels and sub-channels and the most inner while loop filling in users' rate requirements. Specifically, on sub-channel (*i, j*), we select user $k' \in \mathcal{K}$ (where $\mathcal{K}$ is the set of remaining users) based on the criterion that we discussed earlier, i.e., with the largest $C_{(i,j)}^{k',\mathrm{DL}} \cdot V_{\mathrm{res}}^{k',\mathrm{DL}}$. The number of TRBs allocated to user k' is

$$n_{(i,j)}^{k',\mathrm{DL}} = \min\left\{ Q_i', \left\lceil \frac{V_{\mathrm{res}}^{k',\mathrm{DL}}}{C_{(i,j)}^{k',\mathrm{DL}} T_0} \right\rceil \right\}. \tag{17}$$

Then we update the remaining bit volume for this user:

$$V_{\mathrm{res}}^{k',\mathrm{DL}} = V_{\mathrm{res}}^{k',\mathrm{DL}} - n_{(i,j)}^{k',\mathrm{DL}} C_{(i,j)}^{k',\mathrm{DL}} T_0. \tag{18}$$

When $V_{\mathrm{res}}^{k',\mathrm{DL}} \leq 0$, i.e., the user's rate requirement is met, we remove this user from $\mathcal{K}$. If there are still remaining TRBs on this sub-channel (*i, j*), we continue to select another user from $\mathcal{K}$ (with the same criterion) and follow the same TRB allocation process. Once all TRBs on this sub-channel are allocated, we move on to the next sub-channel and eventually the next channel.

Algorithm 1 terminates when either all users' DL rate requirements are met (i.e., $\mathcal{K} = \emptyset$) or TRBs on all DL channels (and sub-channels) are already allocated (i.e., $\mathcal{F}^{\mathrm{DL}} = \emptyset$). The DL sub-problem is infeasible if there remains some user with $V_{\mathrm{res}}^{k,\mathrm{DL}} > 0$ after all TRBs are allocated. Otherwise, we conclude that it is feasible.

D. Computational Complexity

The time complexity of CURT is determined by the feasibility check of a DL/UL sub-problem, while its space complexity is determined by the number of DL/UL sub-problems.

For each DL/UL feasibility check, we need to go through at most $|\mathcal{F}||\mathcal{S}_i|$ sub-channels. On each sub-channel, user selection is on the order of $O(|K|)$. So the time complexity of a feasibility check is $O(|\mathcal{F}||\mathcal{S}_i||\mathcal{K}|)$.

For space complexity, we need to determine how many processors are needed for parallel feasibility checks. From our analysis in Section V-B, we know that the total number of parallel UL/DL sub-problems is $2(2^{|\mathcal{F}|}-2)|\mathcal{Z}|$. Based on (15), it is upper bounded by $2(2^{|\mathcal{F}|}-2)|\mathcal{F}|N_{\mathrm{SF}}$, which is independent of the number of LTE and Wi-Fi users. That is, the number of GPU cores needed by CURT does not increase with the number of LTE or Wi-Fi users.

E. Guaranteeing Feasibility via Traffic Management

Problem OPT (and OPT-R) may have no feasible solution to meet the constraints on rate requirements $R^{k,\mathrm{UL}}$'s and $R^{k,\mathrm{DL}}$'s for all $k \in \mathcal{K}$ and LTE's channel usage Q_i's for all $i \in \mathcal{F}$. On the other hand, CURT may not be able to find a feasible solution when OPT-R is actually feasible since CURT does not traverse the entire search space of OPT-R. In fact, it is impossible for any algorithm to go through all possible solutions of an NP-hard problem (such as OPT-R) while meeting the real-time requirement of $\sim$1 ms. To guarantee feasibility, CURT should work in concert with a traffic management mechanism. Specifically, if a user's UL/DL rate requirements cannot be fully met after scheduling a SF, the traffic management module should negotiate with this user to decrease its rate requirements or switch it to a band on licensed spectrum. Detailed design of such a traffic management module is beyond the scope of this paper.

VI. Implementation

As a proof of concept, we implement CURT on off-the-shelf Nvidia Quadro P6000 GPUs [27] based on the CUDA programming model [24]. Our implementation is done on a Dell desktop computer with an Intel Xeon E5-2687W v4 CPU (3.0 GHz) and dual Nvidia Quadro P6000 GPUs. Each Quadro P6000 GPU consists of an array of 30 streaming multiprocessors (SMs) with 3840 CUDA cores (128 cores per SM). In each SM, there is 96 KB shared memory.[3] CUDA is a programming model for general-purpose parallel computing on Nvidia GPUs. Logically, CUDA executes a multi-threaded function (termed a kernel) on GPU through a hierarchy of threads. This hierarchy has a two-layer structure, where the upper layer is a grid of thread blocks, and at the lower layer each block consists of a number of threads. Each block is executed by a single SM and an SM addresses one block at a time. All blocks are queued and scheduled among the available SMs on the GPU. A thread is the smallest computing granularity under CUDA. For current GPUs, the maximum number of threads allowed per thread block is 1024. Threads within a block are handled by CUDA cores on the assigned SM and can communicate among each other via shared memory.

Fig. 5 illustrates our implementation, which consists of four stages: (i) transferring input data from host (CPU) memory to GPU global memory; (ii) performing parallel UL/DL feasibility checks in GPU (refer to Section V-C); (iii) transferring results of feasibility checks from GPU back to host; and

[3]Shared memory is on-chip and locates at each SM, which can only be accessed by cores within an SM. In contrast, GPU's global memory is off-chip and accessible to cores from all SMs. Access to shared memory is much faster than access to GPU's global memory [25].

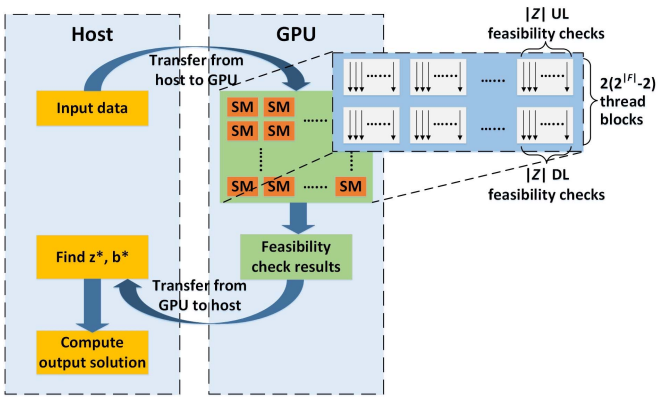


Fig. 5. An illustration of our implementation of CURT.

TABLE II
CLASSIFICATION OF INPUT TO CURT

Data	Time Variation	Transferring to GPU per SF
T_{SF} (also N_{SF})	Slow	No
$\mathcal{F}$		
$\mathcal{S}_i$ for $i \in \mathcal{F}$		
$\mathcal{K}$		
w_i for $i \in \mathcal{F}$		
Q_i for $i \in \mathcal{F}$		
$R^{k,\mathrm{UL}}, R^{k,\mathrm{DL}}$ for $k \in \mathcal{K}$		
The $(2^{\lvert\mathcal{F}\rvert} - 2)$ possible UL/DL channel assignment patterns		
The set $\mathcal{Z}$ of possible objective values of OPT-R		
$C^{k,\mathrm{UL}}_{(i,j)}, C^{k,\mathrm{DL}}_{(i,j)}$ for $i \in \mathcal{F}, j \in \mathcal{S}_i, k \in \mathcal{K}$	Fast	Yes

(iv) determining the output scheduling solution on the host. Next we present details of these four stages.

Transferring Input Data to GPU: Input data to CURT are classified into *fast-varying* and *slow-varying* data, depending on how fast they vary in time. A classification of input data is given in Table II. Fast-varying data refer to those that vary from SF to SF and thus must be uploaded to GPU for each SF. In each SF, we transfer fast-varying data from host to GPU before executing the kernel for feasibility checks. On the other hand, slow-varying data does not vary from SF to SF and only need to be updated over a time period much longer than a SF (hundreds of ms or more). As these data do not change per SF, we define a separate kernel to transfer them from host to GPU only when they vary.

When measuring the scheduling time of CURT, we include the transferring time for fast-varying data since they must be updated to GPU per SF. The time cost for updating slow-varying data is not incorporated in the scheduling time as such transfer only occurs on a much larger time scale.

Performing Feasibility Checks on GPU: When all input data is available in GPU's global memory, the second stage is to perform feasibility check on a total of $2(2^{|\mathcal{F}|} - 2) \cdot |\mathcal{Z}|$ UL/DL sub-problems (refer to Section V-B) in parallel by a kernel with a grid of thread blocks. We need to take the following factors into consideration when designing the kernel: 1) the number of available SMs; 2) the capacity of shared memory on each SM; and 3) the design of feasibility checks. First, since an SM can only execute one thread block at a time and works sequentially if it is assigned with multiple blocks, the total number of thread blocks in the grid should match the number of SMs to maximize occupancy while minimizing sequential operations on SMs. Next, to meet the time requirement of $\sim$1 ms, we should make the best use of shared memory on each SM and reduce access to global memory. However, the size of all input data of CURT (see Table II) may exceed the capacity of shared memory per SM (96 KB for Nvidia Quadro P6000 GPU). We need to ensure that the shared memory used by each thread block is within the per-SM capacity limit. Last, our proposed feasibility check algorithm (Algorithm 1) is of sequential design and would be used on a large number of independent UL/DL sub-problems. To maximize parallelism, we should properly distribute all feasibility checks among the defined thread blocks and utilize the large number of threads per block for concurrent processing.

With the above considerations, we use a kernel with a one-dimensional grid of $2(2^{|\mathcal{F}|} - 2)$ thread blocks for this stage. Each thread block addresses the $|\mathcal{Z}|$ feasibility checks for all UL (or all DL) sub-problems under one of the $(2^{|\mathcal{F}|} - 2)$ UL/DL channel assignments. Detailed operations are:

- *Step 1:* At the beginning of the kernel, we use all 1024 threads in each block to load input data (for either UL or DL) from global memory into shared memory successively in a round-by-round manner (1024 elements per round).
- *Step 2:* After loading input data, we proceed to run the $|\mathcal{Z}|$ feasibility checks in each block. Each feasibility check is executed by a single thread. We only keep result of the check, i.e., feasible or infeasible, while the scheduling solution is discarded (not stored in either shared memory or global memory). More explanation on this will be given later in the next stage.
- *Step 3:* When feasibility checks on all thread blocks are completed, we store the check results into GPU's global memory.

The above kernel structure meets all of our design considerations. Since $|\mathcal{F}|$ is not a large number and the number of thread blocks would be comparable to that of available SMs. For example, for $|\mathcal{F}| = 5$ [17], we have $2(2^{|\mathcal{F}|} - 2) = 60$, which is equal to the number of SMs from dual Quadro P6000 GPUs that we use for implementation. In addition, since each thread block only addresses sub-problems for one transmission direction (UL or DL), we only need to load input data of the specific direction into the shared memory of the assigned SM. Thus the required shared memory for input data per thread block (SM) is reduced by half. Further, feasibility checks on all thread blocks run concurrently and in parallel, with no need for communication among blocks and threads. From (15), the number of parallel feasibility checks that each thread block needs to execute is upper bounded by $|\mathcal{Z}| \leq |\mathcal{F}| \cdot N_{\mathrm{SF}}$, which does not exceed the maximum number of threads per block,

i.e., 1024. For example, for $|\mathcal{F}| = 5$ and $N_{\rm SF} = 30$ (refer to Section III), we have $|\mathcal{Z}| \leq 150$.

Transferring Feasibility Results to Host: When the GPU kernel completes all feasibility checks and stores these results in GPU's global memory, we transfer these results back to the host memory. This transferring time overhead is included in the total scheduling time.

Determining Output Scheduling Solution on Host: The feasibility check results obtained from GPU indicate whether or not each UL/DL sub-problem under a specific channel assignment and objective value is feasible. We now determine the final output scheduling solution on the host through the following operations:

- *Step 1:* First, we find the smallest objective value $z^* \in \mathcal{Z}$ under the best channel assignment b^* (among the $(2^{|\mathcal{F}|} - 2)$ assignments) that has both the corresponding UL and DL sub-problems pass feasibility checks (both are feasible). This is done on the host by traversing the feasibility check results. Specifically, under each channel assignment b, we have one UL sub-problem and one DL sub-problem for each objective value $z \in \mathcal{Z}$. Denote z_b^* as the smallest objective value (if exists) under channel assignment b with its both UL and DL sub-problems being feasible. Then we have $z^* = \min_b z_b^*$ and $b^* = \arg\min_b z_b^*$.
- *Step 2:* We then run Algorithm 1 on the host with the input of objective value z^* and channel assignment b^*. The obtained scheduling solution is the final output solution of CURT.

The computational time of this stage is included in the total scheduling time of CURT.

The reason why we do not store candidate solutions on GPU and instead re-compute the output solution on the host is as follows. First, storing candidate solutions on GPU during feasibility checks would result in a large amount of access to GPU's global memory (as shared memory is insufficient). But this is unacceptable under our tight time constraint of $\sim$1 ms. Second, transferring solutions from GPU to host is actually more time-consuming than doing a sequential check of results and re-computing the scheduling solution (using Algorithm 1) one more time on the host.

VII. Experimental Validation

In this section, we validate and evaluate the performance of CURT through experiments.

A. Experimental Platform and Network Parameters

Our experiments are done on a Dell desktop computer with an Intel Xeon E5-2687W v4 CPU (3.0 GHz) and dual Nvidia Quadro P6000 GPUs (each with 30 SMs and 3840 CUDA cores). The communication bus between CPU and GPU is a PCIe 3.0 X16 slot with default configuration. Implementation on GPU is based on the Nvidia CUDA version 9.2 programming model [24]. We employ IBM CPLEX Optimizer version 12.7.1 [28] on the same computer to compute optimal or lower-bound solution to OPT-R.[4] During our experiments, one of the two GPUs (GPU 1) also performs graphics processing and display functions for the desktop computer, in addition to the computational task associated with CURT, while the other GPU (GPU 2) is solely dedicated to the computational task associated with CURT. Since CURT would be implemented on a small cell eNB and does not need to perform graphics processing and display as in a desktop computer, it is more reasonable to use the timing results from GPU 2 to demonstrate CURT's performance.

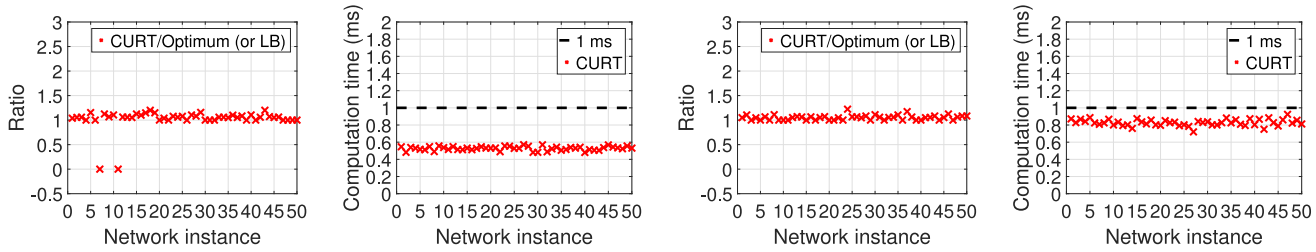
We assume that all LTE and Wi-Fi users in the small cell are within each other's transmission and interference ranges and there is no hidden-node. Suppose that $|\mathcal{F}| = 5$ channels in the 5 GHz unlicensed spectrum are chosen for LTE/Wi-Fi coexistence, with carrier frequencies being 5.20, 5.22, 5.24, 5.26, and 5.28 GHz, respectively. Each channel has 20 MHz bandwidth and is divided into $|\mathcal{S}_i| = 100$ sub-channels (each with 180 kHz bandwidth). The time duration of a SF $T_{\rm SF} = 30$ ms. To facilitate reproducibility of the results, we use Shannon's formula to calculate data rates $C_{(i,j)}^{k,\rm UL}$'s and $C_{(i,j)}^{k,\rm DL}$'s, i.e., $C_{(i,j)}^{k,\rm UL} = B\log_2(1 + \frac{\rho^{\rm UL} l_i^k |h_{(i,j)}^k|^2}{\sigma_0^2})$ and $C_{(i,j)}^{k,\rm DL} = B\log_2(1 + \frac{\rho^{\rm DL} l_i^k |h_{(i,j)}^k|^2}{\sigma_0^2})$, where B is the bandwidth of a sub-channel, $\rho^{\rm UL}$ and $\rho^{\rm DL}$ are an LTE user's UL transmit power and eNB's DL transmit power on each sub-channel, respectively, l_i^k is the pathloss between the eNB and user $k \in \mathcal{K}$ on channel $i \in \mathcal{F}$, $h_{(i,j)}^k$ is the Rayleigh fading coefficient between the eNB and user $k \in \mathcal{K}$ on sub-channel (i, j) with mean 0 and variance 1, and σ_0^2 denotes the noise power. The pathloss is modeled by the Friis transmission equation $l_i^k = G_t G_r (\frac{\lambda_i}{4\pi D_k})^2$, where G_t and G_r are respectively the transmit and receive antenna gains, λ_i is the wavelength on channel i, and D_k denotes the distance between the eNB and user k.

Antenna gains are set to $G_t = G_r = 1$. Since transmit power of eNB is typically higher than that of user terminals, we set $\rho^{\rm DL}/\sigma_0^2 = 120$ dB and $\rho^{\rm UL}/\sigma_0^2 = 115$ dB. For Q_i, we let $Q_i = \lfloor N_{\rm SF} \frac{|\mathcal{K}|/|\mathcal{F}|}{|\mathcal{K}|/|\mathcal{F}| + U_i} \rfloor$ as described in Section IV.

B. Results

Optimality and Timing Performance: We now evaluate the performance of CURT under realistic network settings. Following 3GPP's evaluation methodology [3], we consider a maximum of 20 users in a LTE small cell. In Fig. 6, we show results for both $|\mathcal{K}| = 10$ and $|\mathcal{K}| = 20$ users. In both cases, distances between eNB and LTE users are randomly and uniformly generated from [1, 30] m. As we discussed in Section V-D, the computational complexity of CURT is independent of the number of Wi-Fi users. Without loss of generality, we assume that U_i on each channel $i \in \mathcal{F}$ is randomly chosen from {1, 2, 3} so that on average there are

[4]To address potentially prohibitively long computation time by CPLEX, we set a time limit of 1 hour. The lower-bound solution is taken as benchmark when CPLEX cannot find optimal solution by 1 hour.



(a) Ratios of objectives. $|\mathcal{K}| = 10$. (b) Computation time. $|\mathcal{K}| = 10$. (c) Ratios of objectives. $|\mathcal{K}| = 20$. (d) Computation time. $|\mathcal{K}| = 20$.

Fig. 6. Achieved objective and timing performance of CURT.

$\sim$10 Wi-Fi users sharing the spectrum with LTE, which is similar to the evaluation scenario in [3].

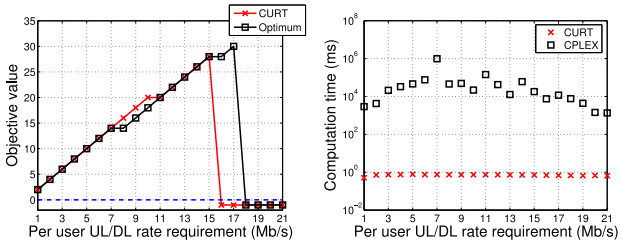
In Fig. 6(a) and (b), UL and DL rate requirements of the 10 users are randomly generated from [10, 40] Mb/s. Fig. 6(a) shows ratios between objective values achieved by CURT and optimal (or lower-bound) solutions found by CPLEX over 50 network instances. For each instance, if CURT finds the optimum, the ratio of objective values equals to one. When CURT fails to find a feasible solution, we set the ratio to zero. Among 50 network instances, CURT finds optimal solutions for 14 instances. We observe that in 2 instances CURT cannot find feasible solution.[5] That is, the percentage that CURT can find a feasible solution is 96%. This is reasonable since CURT does not traverse the entire search space of OPT-R. Among the instances where CURT can find feasible solutions, the average of CURT's ratios is 1.04, with a variance of 0.0021.

Fig. 6(b) shows computation time of CURT. Mean computation time of CURT is 0.53 ms, with a maximum of 0.57 ms and a variance of 0.0006. In contrast, the mean computation time of CPLEX for finding the optimal (or lower-bound) solution is 1246.35 s.

In Fig. 6(c) and 6(d), the rate requirements of the 20 users are randomly generated from [5, 20] Mb/s. Fig. 6(c) shows that for 18 out of 50 instances, CURT achieves optimal objectives. Also, for all 50 instances, CURT is able to find feasible solutions. The average of ratios by CURT (over optimum) is 1.04, with a variance of 0.0014. Mean computation time of CURT is 0.83 ms, with a maximum of 0.92 ms and a variance of 0.0014. In contrast, the mean of CPLEX's computation time for finding optimal or lower-bound solution is 987.86 s. Numerical results in Fig. 6 demonstrate that CURT can indeed find near-optimal solution while meeting the real-time requirement of $\sim$1 ms.

Note that the computational time of CPLEX with 10 users is higher than that with 20 users. This is because the computational time of CPLEX depends on a number of factors, with the number of users in the cell being just one factor. CPLEX is based on branch-and-bound algorithm with cutting plane method, which closes the optimality gap by iteratively comparing the difference between the objective of the best feasible solution and the lower-bound (or upper-bound). In general, its computational time increases with the number of decision variables. But the amount of time in each iteration depends heavily on the local search algorithm to find a feasible solution. For problem OPT, although there are fewer variables in the case of 10 users, it is harder to find a feasible solution as the rate requirement for each user ranges from 10 to 40 MB/s. On the other hand, when there are 20 users, the rate requirement for each user ranges from 5 to 20 MB/s (a narrower range). In this case, the time to find a feasible solution is actually smaller (than 10 users). In contrast, CURT does not follow branch-and-bound framework. Its computational time does not involve a local search to find a feasible solution. Its time complexity (for executing parallel feasibility check in each sub-problem) grows linearly with the number of users (refer to Section V-D).

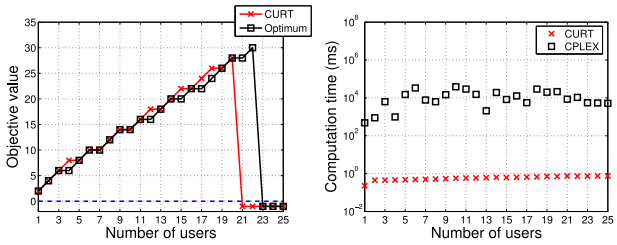
[5]In this case, the traffic management module may be invoked to ensure feasibility.



(a) (b)

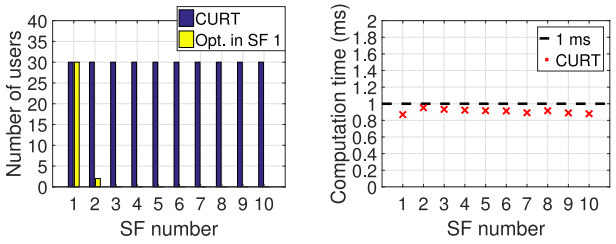
Fig. 7. Performance of CURT under increasing per-user rate requirement. (a) Objective values, where -1 indicates infeasibility. (b) Computation time.

Varying LTE Traffic Load: We now evaluate the behavior of CURT under varying LTE traffic load. Suppose the cell has 30 users. To identify each user distinctly, we name them as user 1 to 30. A user's channel capacity is a function of its distance to the base station, in addition to time-varying channel fading. The number of TRBs required in scheduling depends on each user's channel capacity and its rate requirement. With a given set of user rate requirements, the higher the users' channel capacity, the fewer TRBs need to be allocated and more users can be accommodated. Thus, for evaluating the number of users that can be supported by CURT, it is necessary to specify users' distances to the eNB. For reproducibility, we hereby disclose users' distances (all randomly generated) to the eNB as follows (in meter): 24.58, 1.29, 5.03, 6.88, 6.76, 18.51, 8.89, 6.77, 1.44, 22.66, 13.91, 28.02, 14.51, 13.14, 25.54, 16.23, 6.88, 20.49, 25.31, 1.57, 20.76, 12.01, 25.12, 15.58, 21.57, 13.44, 9.83, 6.50, 6.61, 20.78. Note that the LTE cell may not be able to meet rate requirements of all these users. It may only serve a subset of users and transfer the



(a) (b)

Fig. 8. Performance of CURT under increasing number of users. (a) Objective values, where -1 indicates infeasibility. (b) Computation time.



(a) (b)

Fig. 9. Performance of CURT under temporal channel variations. 30 users. User mobility $v = 1.5$ km/h. (a) Number of users whose rate requirements are met. (b) Computation time of CURT.

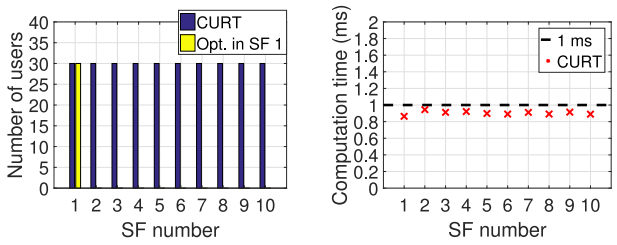
remaining users to licensed band (via the traffic management module) as described in Section V-E. We set $Q_i = N_{\text{SF}}/2$ and $U_i = 2$ for all channels in $\mathcal{F}$.

In Fig. 7(a), we choose the first 20 users and increase their UL/DL rate requirements simultaneously from 1 Mb/s. We see that CURT can support a maximum per-user UL/DL rate requirement of 15 Mb/s, while the optimal solution can support up to 17 Mb/s. On the other hand, in Fig. 7(b), we see that CURT's computation time is consistently less than 1 ms while CPLEXs computation time varies from 1.36 s to 984.33 s.

In Fig. 8(a), we fix the per-user UL/DL rate requirements to 15 Mb/s and increase the number of LTE users (starting from user 1). It shows that CURT can satisfy the first 20 users, while the optimal solution can support the first 22 users. In Fig. 8(b) we see that computation time of CURT is no more than 1 ms while CPLEXs computation time varies from 484 ms to 38.47 s.

Temporal Channel Variations: In an LTE small cell, a user's mobility is typically low and channel conditions only experience small change across consecutive SFs. Even so, we argue that it is still necessary to re-compute scheduling solution for each SF, as we show below.

To account for temporal channel correlations, let's consider a quasi-static block fading channel model [26]. Assume that fast fading coefficients $h^k_{(i,j)}$'s remain constant during an SF and only vary for the next SF. Denote $h^k_{(i,j)}(t)$ as the fast fading coefficient of user k on sub-channel (i, j) in SF t. Then the fast fading coefficient in SF $(t+1)$ is determined by $h^k_{(i,j)}(t+1) = \alpha h^k_{(i,j)}(t) + \tilde{h}^k_{(i,j)}$, where α represents the temporal autocorrelation between consecutive SFs, and $\tilde{h}^k_{(i,j)}$ is the uncorrelated Rayleigh channel variation with mean 0 and



(a) (b)

Fig. 10. Performance of CURT under temporal channel variations. 30 users. User mobility $v = 3.0$ km/h. (a) Number of users whose rate requirements are met. (b) Computation time of CURT.

variance $(1-\alpha^2)$. α is calculated as $\alpha = J_0(2\pi f_{\text{M}} T_{\text{SF}})$ [26], where $J_0(\cdot)$ is zeroth-order Bessel function of the first kind. We now compare the performance of CURT and a fixed scheduling solution over a period of 300 ms, which consists of 10 SFs, numbered as SF 1, 2, ..., 10. We use the same network scenario with 30 users as described earlier. Each users' UL and DL rate requirements are both set to 10 Mb/s. For the fixed solution, we employ the optimal solution that is computed off-line for SF 1 for all the 10 SFs. Results under user mobility $v = 1.5$ Km/h and $v = 3.0$ Km/h are presented in Fig. 9 and Fig. 10, respectively.[6] In Fig. 9(a) and Fig. 10(a), blue and yellow bars represent the numbers of LTE users whose UL and DL rate requirements are both satisfied by CURT and the fixed solution, respectively. We can see that CURT always finds solutions in 1 ms and meets all 30 users' rate requirements while the fixed solution (optimal only for SF 1) expires quickly starting from SF 2 due to channel variations and can no longer satisfy users' rate requirements. These results indicate that even with low user mobility, it is still necessary to have an LTE scheduler capable of re-computing scheduling solution for each SF in real-time (on $\sim$1 ms time scale).

VIII. Conclusion

In this paper, we investigated a resource scheduling problem for LTE in unlicensed bands with CSAT-based coexistence paradigm for ensuring fairness to Wi-Fi. We formulated the scheduling problem as an optimization problem of selecting channels for LTE UL and DL transmissions, dividing transmission time on each channel for LTE and Wi-Fi, and allocating RBs on all channels based on LTE users' channel conditions and UL/DL rate requirements. The objective is to minimize LTE's adverse impact on Wi-Fi on each channel. We proved that this scheduling problem is NP-hard. Then we presented CURT – a novel design of an LTE scheduler for CSAT-based coexistence with Wi-Fi that is able to obtain near-optimal scheduling solution on $\sim$1 ms time scale. CURT exploits problem decomposition techniques and massive number of cores on low-cost off-the-shelf GPUs to achieve parallel real-time computing. To validate the performance of CURT, we implemented it on Nvidia GPU/CUDA platform and

[6]User mobility no greater than 3.0 Km/h (walking speed) is a common assumption for LTE small cells in 3GPP [3]. High-speed users are usually supported by macro cells in licensed spectrum to avoid frequent handover.

conducted extensive experiments. Our experimental results demonstrated that CURT can deliver near-optimal scheduling solution on ∼1 ms time scale and meet all of our design expectations.

APPENDIX
PROOF OF LEMMA 1

Proof: Our proof is based on the set partitioning problem (SPP), which is known to be NP-complete [29]. We consider a decision version of our LTE scheduling problem and construct a polynomial-time reduction from SPP to our problem. We will show that an SPP instance is feasible if and only if the corresponding instance of our constructed problem is feasible, which completes the proof.

SPP is defined as follows. Given a set of positive integers $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$, determine whether or not $\mathcal{A}$ can be partitioned into two subsets $\mathcal{A}_1$ and $\mathcal{A}_2$, where $\mathcal{A}_1 \subset \mathcal{A}$ and $\mathcal{A}_2 = \mathcal{A} \setminus \mathcal{A}_1$, such that the sums of elements in $\mathcal{A}_1$ and $\mathcal{A}_2$ are identical, i.e., $\sum_{a_j \in \mathcal{A}_1} a_j = \sum_{a_j \in \mathcal{A}_2} a_j = \frac{1}{2} \cdot \sum_{a_j \in \mathcal{A}} a_j$.

A decision version of the LTE scheduling problem is to determine whether or not under a given collection of parameters ($R^{k,\mathrm{UL}}$, $R^{k,\mathrm{DL}}$, $C^{k,\mathrm{UL}}_{(i,j)}$, $C^{k,\mathrm{DL}}_{(i,j)}$, Q_i, w_i for all $k \in \mathcal{K}, i \in \mathcal{F}, j \in \mathcal{S}_i$), there exists a feasible TRB-to-user scheduling solution satisfying all constraints with an objective value no more than Z (Z>0).

Consider an arbitrary instance of SPP involving a set $\mathcal{A}$ of N positive integers $a_1, \ldots, a_N$. Solving such an SPP instance is to find a strategy that assigns each element $a_j \in \mathcal{A}$ to either $\mathcal{A}_1$ or $\mathcal{A}_2$ such that $\sum_{a_j \in \mathcal{A}_1} a_j = \sum_{a_j \in \mathcal{A}_2} a_j$. We now construct a corresponding instance of our problem. Our constructed instance consists of two users named user 1 and 2 ($\mathcal{K} = \{1, 2\}$), one channel named channel 1 ($\mathcal{F} = \{1\}$), and N sub-channels on this channel named sub-channel $1, \ldots, N$ ($\mathcal{S}_1 = \{1, \ldots, N\}$). Given the N positive integers $a_1, \ldots, a_N$, input parameters to our problem instance are set as: $N_{\mathrm{SF}} = 1$, $Q_1 = 1$, $w_1 = 1$, $C^{1,\mathrm{UL}}_{(1,j)} = C^{2,\mathrm{UL}}_{(1,j)} = a_j$ and $C^{1,\mathrm{DL}}_{(1,j)} = C^{2,\mathrm{DL}}_{(1,j)} = 0$ for $j \in \{1, \ldots, N\}$, $R^{1,\mathrm{UL}} = R^{2,\mathrm{UL}} = \frac{1}{2} \cdot \sum_{a_j \in \mathcal{A}} a_j$, $R^{1,\mathrm{DL}} = R^{2,\mathrm{DL}} = 0$, and $Z = 1$. We aim to determine whether there exists a feasible scheduling solution for allocating the N TRBs on channel 1 (since $Q_1 = 1$) to user 1 and 2, such that their rate requirements are met and the objective value is no more than $Z = 1$ (the objective value is $+\infty$ if there is no feasible solution). Since $R^{1,\mathrm{DL}} = R^{2,\mathrm{DL}} = 0$, we can readily fix channel assignment variables $I^{\mathrm{UL}}_1 = 1$ and $I^{\mathrm{DL}}_1 = 0$. Then we have $n^{k,\mathrm{DL}}_{(1,j)} = 0$ for all $k \in \{1, 2\}$ and $j \in \{1, \ldots, N\}$. What remains to be determined is how to allocate TRBs for UL transmission, i.e., fixing variables $n^{k,\mathrm{UL}}_{(1,j)}$ for $k \in \{1, 2\}$ and $j \in \{1, \ldots, N\}$.

The reduction from the SPP instance to our constructed problem instance is as follows. Assigning $a_1, \ldots, a_N$ to $\mathcal{A}_1$ and $\mathcal{A}_2$ corresponds to allocating the N TRBs to user 1 and 2 for UL transmission. Each element a_i is assigned to at most one of the two subsets, which corresponds to our constraint that each TRB can be allocated to at most one of the two users. Specifically, if an element a_j ($j \in \{1, \ldots, N\}$) is assigned to $\mathcal{A}_1$ (or $\mathcal{A}_2$), correspondingly, for our problem we fix $n^{1,\mathrm{UL}}_{(1,j)} = 1$, $n^{2,\mathrm{UL}}_{(1,j)} = 0$ (or $n^{1,\mathrm{UL}}_{(1,j)} = 0$, $n^{2,\mathrm{UL}}_{(1,j)} = 1$). Such reduction also applies reversely from our problem instance to the SPP instance. Clearly, the reduction is on the order of $O(N)$ and is polynomial in time.

We now verify that an SPP instance is feasible (i.e., the set $\mathcal{A}$ can be partitioned into two subsets with equal sum of elements) if and only if our problem instance is feasible (i.e., rate requirements $R^{1,\mathrm{UL}}$ and $R^{2,\mathrm{UL}}$ can be met with the objective equal to $Z = 1$). Indeed, if SPP has a feasible partition strategy satisfying $\sum_{a_j \in \mathcal{A}_1} a_j = \sum_{a_j \in \mathcal{A}_2} a_j = \frac{1}{2} \cdot \sum_{a_j \in \mathcal{A}} a_j$, then based on the reduction we have $\sum_{j=1}^{N} n^{1,\mathrm{UL}}_{(1,j)} C^{1,\mathrm{UL}}_{(1,j)} = \sum_{a_j \in \mathcal{A}_1} a_j = \frac{1}{2} \cdot \sum_{a_j \in \mathcal{A}} a_j = R^{1,\mathrm{UL}}$ and $\sum_{j=1}^{N} n^{2,\mathrm{UL}}_{(1,j)} C^{2,\mathrm{UL}}_{(1,j)} = \sum_{a_j \in \mathcal{A}_2} a_j = \frac{1}{2} \cdot \sum_{a_j \in \mathcal{A}} a_j = R^{2,\mathrm{UL}}$, i.e., our problem instance is feasible. On the other hand, if our problem has a feasible scheduling solution that meets rate requirements $R^{1,\mathrm{UL}}$ and $R^{2,\mathrm{UL}}$, then following the reduction we can determine the partition strategy for the SPP instance that satisfies $\sum_{a_j \in \mathcal{A}_1} a_j = \sum_{a_j \in \mathcal{A}_2} a_j$. This completes the proof. ■

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their feedback. They thank Nvidia AI Lab (NVAIL) in Santa Clara, CA, USA for its unrestricted gift and equipment donation to our research. All opinions expressed in this paper are the authors' and do not necessarily reflect the views and opinions of NSF or Nvidia.

REFERENCES

[1] "LTE in unlicensed spectrum: Harmonious coexistence with Wi-Fi," Qualcomm, San Diego, CA, USA, White Paper. [Online]. Available: https://www.qualcomm.com/media/documents/files/lte-unlicensed-coexistence-whitepaper.pdf

[2] *LTE-U CSAT Procedure TS Version 1.0*, LTE-U Forum, Stockholm, Sweden. [Online]. Available: http://www.lteuforum.org/uploads/3/5/6/8/3568127/lte-u_forum_lte-u_sdl_csat_procedure_ts_v1.0.pdf

[3] *Study on Licensed-Assisted Access to Unlicensed Spectrum, Version 13.0.0*, 3GPP, Sophia Antipolis, France, Rep. 3GPP TR 36.889. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2579

[4] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation, Version 15.2.0*, Standard 3GPP TS 36.211. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2425

[5] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures, Version 15.2.0*, Standard 3GPP TS 36.213. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2427

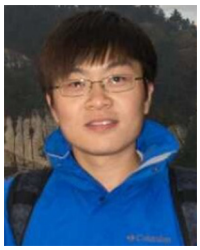
[6] Z. Guan and T. Melodia, "CU-LTE: Spectrally-efficient and fair coexistence between LTE and Wi-Fi in unlicensed bands," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[7] Y. Huang, Y. Chen, Y. T. Hou, W. Lou, and J. H. Reed, "Recent advances of LTE/WiFi coexistence in unlicensed spectrum," *IEEE Netw.*, vol. 32, no. 2, pp. 107–113, Mar./Apr. 2018.

[8] C. Cano, D. J. Leith, A. Garcia-Saavedra, and P. Serrano, "Fair coexistence of scheduled and random access wireless networks: Unlicensed LTE/WiFi," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3267–3281, Dec. 2017.

[9] A. Abdelfattah and N. Malouch, "Modeling and performance analysis of Wi-Fi networks coexisting with LTE-U," in *Proc. 36th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, May. 2017, pp. 1–9.

[10] A. M. Voicu, L. Simić, and M. Petrova, "Inter-technology coexistence in a spectrum commons: A case study of Wi-Fi and LTE in the 5-GHz unlicensed band," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 3062–3077, Nov. 2016.
[11] Q. Chen, G. Yu, R. Yin, A. Maaref, G. Y. Li, and A. Huang, "Energy efficiency optimization in licensed-assisted access," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 723–734, Apr. 2016.
[12] Q. Chen, G. Yu, and Z. Ding, "Optimizing unlicensed spectrum sharing for LTE-U and WiFi network coexistence," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2562–2574, Oct. 2016.
[13] H. Ko, J. Lee, and S. Pack, "Joint Optimization of Channel Selection and Frame Scheduling for Coexistence of LTE and WLAN," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6481–6491, Jul. 2018.
[14] *T-Mobile Completes Nation's First Live Commercial Network Test of License Assisted Access (LAA)*. Accessed: Jun. 2017. [Online]. Available: https://www.t-mobile.com/news/lte-u
[15] Y. Huang, S. Li, Y. T. Hou, and W. Lou, "GPF: A GPU-based design to achieve ∼100 μs scheduling for 5G NR," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, New Delhi, India, 2018, pp. 207–222.
[16] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN); Overall Description; Stage 2, Version 15.2.0*, Standard 3GPP TS 36.300. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2430
[17] *Evolved universal terrestrial radio access (E-UTRA); Carrier aggregation; Base station (BS) radio transmission and reception, version 10.1.0*, 3GPP, Sophia Antipolis, France, Rep. 3GPP TR 36.808. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2487
[18] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, San Francisco, CA, USA, Mar. 2003, pp. 844–852.
[19] A. L. Toledo, T. Vercauteren, and X. Wang, "Adaptive optimization of IEEE 802.11 DCF based on Bayesian estimation of the number of competing terminals," *IEEE Trans. Mobile Comput.*, vol. 5, no. 9, pp. 1283–1296, Sep. 2006.
[20] T. S. Rappaport, *Wireless Communications: Princlples and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996, Ch. 5.
[21] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 678–700, 2nd Quart., 2013.
[22] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley, 1999.
[23] Y. T. Hou, Y. Shi, and H. D. Sherali, *Applied Optimization Methods for Wireless Networks*. Cambridge, U.K.: Cambridge Univ., Press, 2014, Ch. 5.
[24] *CUDA Toolkit Documentation v9.2.148*, Nvidia, Santa Clara, CA, USA. [Online]. Available: https://docs.nvidia.com/cuda/index.html
[25] *CUDA C Best Practices Guide*, Nvidia, Santa Clara, CA, USA. [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html
[26] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650–1662, Jul. 2005.
[27] *Data sheet: Quadro P6000*, Nvidia, Santa Clara, CA, USA. [Online]. Available: https://images.nvidia.com/content/pdf/quadro/data-sheets/192152-NV-DS-Quadro-P6000-US-12Sept-NV-FNL-WEB.pdf
[28] *IBM ILOG CPLEX Optimizer*. Accessed: Dec. 4, 2019. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer
[29] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: Freeman, 1990.



Yan Huang (S'15) received the B.S. and M.S. degrees in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with Virginia Tech, Blacksburg, VA, USA. His research interests are GPU-based real-time optimizations for wireless networks and machine learning for communications.

Yongce Chen (S'16) received the B.S. and M.S. degrees in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with Virginia Tech, Blacksburg, VA, USA. His current research interests include wireless network optimization, MIMO techniques, and real-time implementation of wireless systems.

Y. Thomas Hou (F'14) received the Ph.D. degree from the NYU Tandon School of Engineering in 1998. He is Bradley Distinguished Professor of electrical and computer engineering, Virginia Tech, Blacksburg, VA, USA, where he joined in 2002. During 1997 to 2002, he was a member of Research Staff with Fujitsu Laboratories of America, Sunnyvale, CA, USA. He has over 250 papers published in IEEE/ACM journals and conferences. He has authored/coauthored two graduate textbooks. He holds five U.S. patents. His current research focuses on developing innovative solutions to complex science and engineering problems arising from wireless and mobile networks. His papers were recognized by six Best Paper Awards from the IEEE and two Paper Awards from the ACM. He was the Steering Committee Chair of the IEEE INFOCOM conference and a member of the IEEE Communications Society Board of Governors. He was/is on the editorial boards of a number of IEEE and ACM transactions and journals.

Wenjing Lou (F'15) received the Ph.D. degree in electrical and computer engineering from the University of Florida in 2003. From 2003 to 2011, she was a faculty member with Worcester Polytechnic Institute. She has been with Virginia Tech since 2011, where she is currently a W.C. English Professor of computer science. During 2014 to 2017, she served as a Program Director for the U.S. National Science Foundation, where she was involved in the Networking Technology and Systems Program and the Secure and Trustworthy Cyberspace Program. Her current research interests focus on privacy protection techniques in networked information systems and cross-layer security enhancement in wireless networks, by exploiting intrinsic wireless networking, and communication properties.