Spectrum Sharing For Delay-Sensitive Applications With Continuing QoS Guarantees

Yuanzhang Xiao, Kartik Ahuja, and Mihaela van der Schaar
Department of Electrical Engineering, UCLA
Emails: yxiao@seas.ucla.edu, ahujak@ucla.edu, mihaela@ee.ucla.edu

***Abstract*—We study a wireless network in which multiple users stream delay-sensitive applications such as video conferencing and video streaming. Existing spectrum sharing policies, which determine when users access the spectrum and at what power levels, are either *constant* (i.e. users transmit simultaneously, at constant power levels) or weighted *round-robin* time-division multiple access (TDMA) (i.e. users access the spectrum in turn, one at a time). Due to multi-user interference, constant policies have low spectrum efficiency. We show that round-robin policies are inefficient for delay-sensitive applications because the various "positions" (i.e. transmission opportunities) in a cycle are not created equal: earlier transmission opportunities are more desirable since they enable users to transmit with lower delays. Specifically, we show that (weighted) round-robin TDMA policies cannot simultaneously achieve high network performance and low transmission delays. This problem is exacerbated when the number of users is large.**

We propose a novel framework for designing optimal TDMA spectrum sharing policies for delay-sensitive applications, which can guarantee their continuing QoS (CQoS), i.e. the desired throughput (and the resulting transmission delay) starting from *every moment in time* is guaranteed for each user. We prove that the fulfillment of CQoS guarantees provides strict upper bounds on the transmission delays incurred by the users. We construct the optimal TDMA policy that maximizes the desired network performance (e.g. max-min fairness or social welfare) subject to the users' CQoS guarantees. The key feature of the proposed policy is that it is *not* cyclic as in (weighted) round-robin policies. Instead, it adaptively determines which user should transmit next, based on the users' remaining amounts of transmission opportunities needed to achieve the desired performance. We also propose a low-complexity algorithm, which is run by each user in a distributed manner, to construct the optimal policy. Simulation results demonstrate that our proposed policy significantly outperforms the optimal constant policy and round-robin policies by up to 6 dB and 4 dB in peak signal-to-noise ratio (PSNR) for video streaming.

I. INTRODUCTION

A variety of bandwidth-intensive and delay-sensitive applications, such as multimedia streaming, gaming, and teleconferencing, are increasingly deployed over wireless networks. Such applications impose huge challenges when deployed over wireless networks, in which the users share the spectrum and cause interference to each other. Hence, it is crucial to design spectrum sharing policies that provide delay-sensitive users with both high rates and low delays.

The spectrum sharing policies studied in earlier works [1]–[4] require the users to transmit at *constant* power levels all the time. We call them *constant* (spectrum sharing) policies. Constant policies are inefficient in many spectrum sharing

low delay, but unfair for user 4:	1	2	3	4	1	2	3	4
fair, but high delay for user 1:	1	2	3	4	4	3	2	1

Fig. 1. Two simple round-robin schedules with cycle length 8 for 4 users. The first one has low delay of 4 for all 4 users, but unfair sharing of transmission opportunities (TXOPs) (i.e. user 4 gets later TXOPs). The second one has a fair sharing of TXOPs, but incurs high maximum delay of 7 for user 1.

scenarios with strong multi-user interference. Under strong multi-user interference, increasing one user's power level significantly degrades the other users' throughput, which results in low spectrum efficiency.

One way to manage interference is to let one user access the spectrum at one time, as in e.g. 802.11e MAC wireless networks [5]. Such policies are commonly known as time-division multiple access (TDMA) policies. This paper focuses on designing optimal TDMA policies for delay-sensitive users.

All the existing TDMA policies are round-robin policies or their variants (e.g. weighted round-robin policies) [5]–[8]. In round-robin policies, time slots are divided into cycles of a fixed predetermined length, and each user transmits in fixed predetermined positions within each cycle. The cyclic nature of round-robin policies simplifies the implementation, but imposes restrictions that render round-robin policies inefficient for delay-sensitive applications. For delay-sensitive application, not all the transmission opportunities (i.e. positions) in a cycle are created equal: the earlier transmission opportunities (TXOPs) are more desirable because they result in higher chances to deliver packets before their delay deadlines [5]–[9]. To ensure that the user's rate and delay constraints are met, round-robin policies need a long cycle, and a careful sharing of TXOPs in a cycle. First, a long cycle is necessary. Suppose that the cycle length is the shortest possible, namely equal to the number of users (as in standard round-robin policies). Then the user allocated to the last TXOP suffers severely from delay. We can compensate this user for its delay by having a longer cycle and allocating some of the extra TXOPs to it. However, a long cycle results in an exponentially increasing (in the cycle length) number of possible policies to choose from. Second, a careful sharing of TXOPs is necessary (see Fig. 1 for an illustration of the following discussion). Suppose that the cycle length is twice the number of users, and that each user gets two positions in a cycle. For fairness, no user should get two advantageous (i.e. earlier) TXOPs. A possible fair sharing may ensure that the user gets both an earlier and a later TXOPs. However, such a schedule is inefficient in worst-case delay: the user who gets the first and the last TXOPs in a cycle will experience high delay between consecutive transmissions. As we will illustrate in our motivating example (Section IV) and by simulations (Section VII), round-robin policies cannot

This research was supported by NSF Grant CCF-1218136 (Xiao, Ahuja, van der Schaar) and fellowship from Guru Krupa Foundation (Ahuja).

TABLE I. COMPARISON WITH RELATED WORKS.

	Spectrum efficiency	CQoS guarantee	Delay guarantee	System or individual performance achieved
Constant [1]–[4]	Low	No	Yes	Neither
Round-robin [5]–[8]	High	No	Yes	Either for small # of users Neither for large # of users
Proposed	High	Yes	Yes	Both

simultaneously achieve high system performance (e.g. max-min fairness) and fulfill the guarantees in terms of transmission delays required by the delay-sensitive users.

In this paper, we propose a framework for designing optimal TDMA spectrum sharing policies for delay-sensitive applications. We define a novel quality-of-service (QoS) metric, called continuing QoS (CQoS) guarantees. CQoS guarantees require a user's average throughput starting from *every point in time* to be higher than a threshold. CQoS guarantees are stricter requirements than conventional QoS guarantees which only guarantee the average throughput starting from the beginning. We will prove that fulfilling CQoS guarantees results in upper bounds on transmission delays. We propose a systematic design methodology, which constructs the optimal TDMA policy that maximizes the system performance (e.g. fairness) subject to the users' CQoS guarantees. The key feature of the proposed policy is that it is not cyclic as in round-robin policies. Instead, it adaptively determines which user should transmit according to the users' remaining amounts of TXOPs needed to achieve the target throughput. We propose a low-complexity distributed algorithm to construct the optimal policy. Simulation results show that our proposed policy significantly outperforms the optimal constant policy [1]–[4] and round-robin policies in peak signal-to-noise ratio (PSNR) for video streaming, by up to 6 dB and 4 dB respectively, .

Finally, we summarize related works in Table I.

II. SYSTEM MODEL

We consider a wireless network with N users. The set of users is denoted by $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$. Each user has a transmitter and a receiver. The channel gain from user i's transmitter to user j's receiver is g_{ij}. Each user i chooses a power level p_i from a compact set $\hat{\mathcal{P}}_i$. We assume that $0 \in \hat{\mathcal{P}}_i$, namely user i can choose not to transmit. We also assume that the users need to comply with some interference temperature constraints (ITCs) measured at K locations in the network. Depending on different scenarios, the ITCs can be imposed by primary users in a cognitive radio network or the base station in a femtocell network. The channel gain from user i's transmitter to the kth location is g_{i0_k}. Each user i knows the channel gain $\{g_{i0_k}\}_{k=1}^K$ to each measurement location and the interference temperature limit $\{I_k\}_{k=1}^K$ at each location. Hence, each user i's set of *admissible* power levels is

$$\mathcal{P}_i = \{p_i \in \hat{\mathcal{P}}_i : g_{i0_k} \cdot p_i \leq I_k, \forall k = 1, \ldots, K\}. \quad (1)$$

For convenience, we also define user i's maximum admissible power level as $P_i^{max} \triangleq \max_{p_i \in \mathcal{P}_i} p_i$.

Remark 1: Our system model is general enough to model many wireless communication networks. It can model wireless ad hoc networks where N users transmit in the unlicensed spectrum without ITCs ($K = 0$). It can also model the uplink (the receivers are co-located) and the downlink (the transmitters are co-located) of a cellular network with possible ITCs imposed by base stations in nearby cells. It can also model cognitive radio networks with N secondary users sharing the spectrum with K primary users imposing ITCs at their receivers.

Denote the joint power profile of all the users by $\boldsymbol{p} = (p_1, \ldots, p_N)$. Since the users cannot jointly decode their messages and can only treat other users' interference as noise, each user i's instantaneous throughput under $\boldsymbol{p}$ is [1]–[4]

$$r_i(\boldsymbol{p}) = \log_2\left(1 + \frac{p_i g_{ii}}{\sum_{j \in \mathcal{N}, j \neq i} p_j g_{ji} + \sigma_i}\right), \quad (2)$$

where σ_i is the noise power at user i's receiver. We write each user i's maximum throughput as $r_i^{max} \triangleq \log_2(1 + p_i g_{ii}/\sigma_i)$, which is achieved when user i transmits at the maximum power level and the other users do not transmit.

The system is time slotted at $t = 0, 1, \ldots$. We write each user i's transmission policy as $\pi_i : \mathbb{N}_+ \to \mathcal{P}_i$, where $\pi_i(t)$ is user i's transmit power level at time t. The spectrum sharing policy is then the collection of all the users' transmission policies, denoted by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$. In a constant policy, we have $\boldsymbol{\pi}(t) = \boldsymbol{p}^{const}$ for all $t \in \mathbb{N}_+$. In a TDMA policy, we have $\|\boldsymbol{\pi}(t)\|_0 = 1$, where $\|\cdot\|_0$ is the ℓ-0 norm, namely the number of nonzero elements in a vector.

Each user i's (discounted) average throughput is defined as

$$R_i(\boldsymbol{\pi}) = (1 - \delta) \sum_{t=0}^{\infty} \delta^t \cdot r_i(\boldsymbol{\pi}(t)), \quad (3)$$

where $\delta \in [0, 1)$ is the discount factor that models the delay-sensitivity of a user [4][9]. A more delay-sensitive user discounts the future throughput more (i.e. has a smaller discount factor), because it has more urgency to transmit.

III. CONTINUING QOS GUARANTEES

The widely-used average QoS (AQoS) guarantee [1]–[9] is that the average throughput is above some fraction γ_i^{avg} of the maximum possible throughput r_i^{max}, namely

$$\textbf{AQoS:}\ R_i(\boldsymbol{\pi}) \geq \gamma_i^{avg} \cdot r_i^{max}. \quad (4)$$

The AQoS guarantee does not provide sufficient guarantees for TDMA policies: even if a user's average throughput (starting from the beginning) $R_i(\boldsymbol{\pi})$ is high, it may get a extremely low throughput starting from certain point in time, because it may not get sufficient TXOPs after certain point.[1]

In this paper, we propose *continuing QoS (CQoS) guarantees*, which ensure that at every point in time, a user's future throughput is guaranteed to be above some minimum requirement. Such continuing guarantees are important for delay-sensitive users. Next, we formally define CQoS guarantees.

First, we define the continuation throughput at time t as

$$R_i^t(\boldsymbol{\pi}) = (1 - \delta) \sum_{\tau=t}^{\infty} \delta^{\tau - t} \cdot r_i(\boldsymbol{\pi}(\tau)), \quad (5)$$

[1]Constant policies do not have such a problem. If they fulfill the average throughput requirements, the throughput will be high enough starting from every point in time. However, it is difficult for them to fulfill the average throughput requirement in the first place, due to multi-user interference.

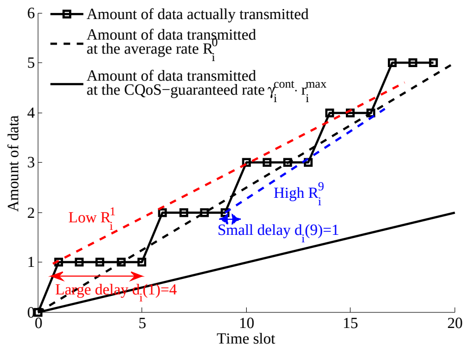

Fig. 2. Relationship of delay and CQoS guarantees of user i. The solid curve with square data points is the amount of data transmitted; each jump in the curve corresponds to a transmission. The two straight lines through the origin are the amount of data transmitted as if the throughput was R_i^0 and $\gamma_i^{cont} \cdot r_i^{max}$, respectively. At each time t, if the continuation throughput R_i^t is higher, the user needs to transmit more after time t. Hence, the corresponding delay $d_i(t)$ is lower.

which is the discounted average throughput starting from time t. Note that $R_i^0(\boldsymbol{\pi}) = R_i(\boldsymbol{\pi})$. Then, the continuing QoS guarantees can be written as

$$\textbf{CQoS:}\ R_i^t(\boldsymbol{\pi}) \geq \gamma_i^{cont} \cdot r_i^{max},\ \forall t = 0, 1, \ldots. \tag{6}$$

To avoid triviality, we assume that $\gamma_i^{cont} < \gamma_i^{avg}$.

A byproduct of the CQoS guarantees is that once they are satisfied, we can also provide upper bounds on the transmission delays of each user. First, we define user i's transmission delay at any time t as

$$\textbf{Transmission Delay:}\ d_i^t(\boldsymbol{\pi}) \triangleq \min_{\tau > t} \{\tau - t : \pi_i(\tau) > 0\}.$$

In words, the transmission delay $d_i^t(\boldsymbol{\pi})$ is the minimum wait time until the next transmission. An upper bound on the transmission delays are critical for delay-sensitive applications. According to Theorem 3, each user's CQoS guarantee leads to an upper bound on its maximum delay $\sup_t d_i^t(\boldsymbol{\pi})$. Fig. 2 illustrate the relationship of delay and CQoS guarantees.

IV. A Motivating Example

We provide a motivating example to illustrate the importance and impact of the CQoS guarantees, and to show the advantage of the proposed optimal TDMA policy over round-robin TDMA policies, in terms of both the performance and the computational complexity. Consider a simple network with four symmetric users. They have the same maximum throughput normalized to 1 bits/s/Hz (i.e. $r_i^{max} = 1,\ \forall i$), and the same discount factor of $\delta = 0.83$. The system performance metric is the max-min fairness (i.e. the minimum of all the users' throughput).

A. CQoS Guarantees and System Performance

We first illustrate the tradeoff between CQoS guarantees and the system performance. Intuitively, CQoS guarantees require that a user has sufficiently many transmission opportunities every once in a while. In other words, the transmission delay at any point in time should be small. For example, for round-robin TDMA policies with cycle length $L = 8$, the one that maximizes CQoS guarantees (or minimizes transmission delay) has a cycle of $1234\ 1234$. It is not difficult to see that, any other policies (we consider the policy with a cycle of $4321\ 4321$ as the same since the users are symmetric) with $L = 8$ will have a maximum transmission delay higher than 4, and will have a worse CQoS. However, the policy with cycle $1234\ 1234$ is not fair: user 4 always transmits at later positions in the cycle, and hence will experience a very low average rate. The policy that achieves the best max-min fairness (i.e. the worst user's rate is maximized) has a cycle of $1234\ 4321$, because user 4 will get two positions in the middle of the cycle. However, such a policy has a worst-case transmission delay of 7 (for user 1). In other words, user 1 has a low CQoS (e.g. its throughput starting from time slot 2 is very low, because it needs to wait until time slot 8 to transmit).

TABLE II. Round-robin TDMA policies cannot achieve both good performance and good CQoS.

Cycle length	$L = 4$	$L = 5$	$L = 6$	$L = 7$
Rates (bits/s/Hz)	0.18	0.19	0.20	0.23
CQoS (bits/s/Hz)	0.18	0.13	0.10	0.07

TABLE III. Performance loss and complexity of round-robin TDMA policies, under CQoS guarantees $\gamma_i^{cont} = 0.1$.

Cycle length	$L = 4$	$L = 5$	$L = 6$	$L = 7$	Proposed
Worst user's rate	0.18	0.19	0.20	0.23	0.25
CQoS guarantee	fulfilled	fulfilled	fulfilled	*violated*	fulfilled
Performance loss (compared to proposed)	40%	32%	25%	N/A	–
# of policies	24	240	1560	8400	–

We illustrate the tradeoff between the CQoS guarantees and the system performance (i.e. max-min fairness) for round-robin TDMA policies in Table II. We can see that with the increase of the cycle length, round-robin TDMA policies achieve better performance, but will have worse CQoS guarantees.

In Table III, we illustrate the performance loss of round-robin TDMA policies compared to the proposed optimal TDMA policy. We find the optimal round-robin policies of different cycle lengths subject to a CQoS guarantee of 0.1 bits/s/Hz. The proposed policy achieves the optimal fairness (i.e. 0.25 bits/s/Hz for all 4 users), and outperforms round-robin policies by at least 20%.

B. Computational Complexity

Our proposed optimal policy is much easier to compute than round-robin policies. Note that in a round-robin policy, the user's performance is determined not only by the number of slots in a cycle but also by the positions of the slots since users are discounting their future throughput (due to delay sensitivity). For a given number of users N and a given cycle length L, the number of non-trivial round-robin schedules (i.e., the ones in which each user gets at least one time slot in a cycle) is greater than N^{L-N}. So searching among these schedules will be totally impractical even if L is moderately larger than N. For instance, for the 4-user case above, achieving energy efficiency within 10% of the optimal policy requires that the cycle length be at least 7, and requires searching among the thousands (8400) of different nontrivial schedules of cycle length 7. Even this small problem is computationally intensive. For a moderate number of users - say 10 - and a cycle length of 20 - we need to search more than ten billion (i.e. 10^{10}) schedules, which is completely intractable. However, we will propose a simple algorithm to compute the optimal policy whose complexity grows *linearly* with the number of users.

Algorithm 1 The Optimal Operating Point Selection (OOPS) algorithm run by user i.

Require: AQoS γ_i^{avg}, precision ε

Set $\underline{\lambda} = 0$, $\bar{\lambda} = 1$, $\lambda = \bar{\lambda}$.
Solve $\frac{\partial W}{\partial r_i} = -\frac{\lambda}{r_i^{max}}$ for r_i^*
Set $r_i^* \leftarrow \max\{r_i^*, \gamma_i^{avg} \cdot r_i^{max}\}$
Broadcast r_i^*/r_i^{max}, and receive r_j^*/r_j^{max} from users $j \neq i$
while $\sum_{j\in\mathcal{N}} r_j^*/r_j^{max} > 1$ **do**
  $\bar{\lambda} \leftarrow 2 \cdot \bar{\lambda}$, $\lambda \leftarrow \bar{\lambda}$
  Solve $\frac{\partial W}{\partial r_i} = -\frac{\lambda}{r_i^{max}}$ for r_i^*, $r_i^* \leftarrow \max\{r_i^*, \gamma_i^{avg} \cdot r_i^{max}\}$
  Broadcast r_i^*/r_i^{max}, and receive $r_j^*/r_j^{max}, j \neq i$
end while
while $\left|\sum_{j\in\mathcal{N}} r_j^*/r_j^{max} - 1\right| > \varepsilon$ **do**
  $\lambda \leftarrow \frac{\underline{\lambda}+\bar{\lambda}}{2}$
  Solve $\frac{\partial W}{\partial r_i} = -\frac{\lambda}{r_i^{max}}$ for r_i^*, $r_i^* \leftarrow \max\{r_i^*, \gamma_i^{avg} \cdot r_i^{max}\}$
  Broadcast r_i^*/r_i^{max}, and receive $r_j^*/r_j^{max}, j \neq i$
  if $\sum_{j\in\mathcal{N}} r_j^*/r_j^{max} < 1$ **then**
    $\bar{\lambda} \leftarrow \lambda$
  else
    $\underline{\lambda} \leftarrow \lambda$
  end if
end while
Normalize $r_i^* \leftarrow r_i^* / \left(\sum_{j\in\mathcal{N}} r_j^*/r_j^{max}\right)$

V. Formulation of The Policy Design Problem

We aim to design a TDMA spectrum sharing policy $\boldsymbol{\pi}$ that maximizes the system performance, defined as a function of the users' throughput, $W(R_1(\boldsymbol{\pi}), \ldots, R_1(\boldsymbol{\pi}))$. We assume that $W(\cdot)$ is increasing, separable, and strictly concave in each argument R_i. Such a definition of system performance is general enough to include the objective functions adopted in most existing works [1]–[9] as special cases. One example of system performance, which will be used in our simulations, is the (normalized) max-min fairness defined as $W(R_1(\boldsymbol{\pi}), \ldots, R_1(\boldsymbol{\pi})) = \min_i \frac{R_i(\boldsymbol{\pi})}{r_i^{max}}$. In addition, we will impose the AQoS and CQoS guarantees. To sum up, we can formally define the policy design problem:

$$\textbf{Design Problem:} \quad \max_{\boldsymbol{\pi}} \; W(R_1(\boldsymbol{\pi}), \ldots, R_1(\boldsymbol{\pi})) \tag{7}$$
$$s.t. \quad \textbf{AQoS: } R_i(\boldsymbol{\pi}) \geq \gamma_i^{avg} \cdot r_i^{max}, \;\forall i,$$
$$\textbf{CQoS: } R_i^t(\boldsymbol{\pi}) \geq \gamma_i^{cont} \cdot r_i^{max}, \;\forall i, t.$$

For the policy design problem (7) to be feasible, we require that $\sum_{i\in\mathcal{N}} \gamma_i^{avg} < 1$.

VI. Solving The Policy Design Problem

Now we solve the policy design problem (7). Our proposed solution (illustrated in Fig. 3) has two phases: an offline phase implemented before run-time, which determines the optimal operating point (i.e. each user's target average throughput), and a low-complexity online phase implemented at run-time, which determines the transmission schedule that achieves the optimal operating point while fulfilling AQoS and CQoS guarantees.

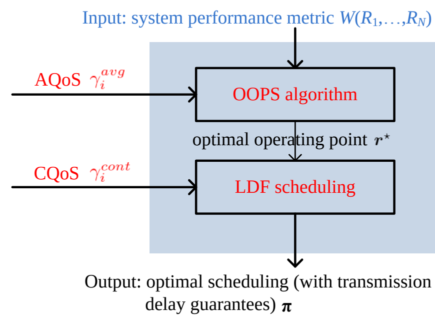

Fig. 3. Illustration of our proposed design framework. The operations in blue and in red are done by the policy designer and the decentralized users, respectively.

A. *Offline Phase – The Optimal Operating Point*

Before run-time, the users solve the following problem in a distributed manner to determine the optimal operating point:

$$\begin{aligned} \boldsymbol{r}^\star = \quad & \arg\max_{\boldsymbol{r}\geq \boldsymbol{0}} && W(r_1, \ldots, r_N) \\ & s.t. && \textstyle\sum_{i\in\mathcal{N}} r_i/r_i^{max} = 1, \\ & && r_i \geq \gamma_i^{avg} \cdot r_i^{max}, \;\forall i. \end{aligned} \tag{8}$$

In (8), the linear equality $\sum_{i\in\mathcal{N}} r_i/r_i^{max} = 1$ comes from the requirement that the policy is TDMA. Intuitively, it ensures that the total fraction of all the users' transmission opportunities sum up to 1. Note that the CQoS guarantees are not present in (8). They will be taken care of in our scheduling policy described in the next subsection.

We propose a distributed optimal operating point selection (OOPS) algorithm (described in Algorithm 1) to solve (8).

Theorem 1: The optimal operating point $\boldsymbol{r}^\star$ that solves (8) can be found by each user running the distributed OOPS algorithm (Algorithm 1), which converges linearly[2] at rate $\frac{1}{2}$.

Proof: See Appendix A. ■

B. *Online Phase – The Optimal Transmission Schedule*

After finding the optimal operating point $\boldsymbol{r}^\star$, we need to determine the transmission schedule that achieves it. Importantly, the transmission schedule should fulfill the CQoS guarantees, which is the major challenge of our solution. We propose a distributed online longest-distance-first (LDF) scheduling algorithm (described in Algorithm 2).

Algorithm 2 has a nice interpretation of longest distance first scheduling. At each time slot t, the user with the largest "distance to target" (i.e. $\alpha_j(t)$ in Algorithm 2) transmits in this time slot[3]. The algorithm updates the distances in the "correct" way, such that the optimal operating points are achieved. Theorem 2 proves the desirable properties of the proposed LDF scheduling algorithm.

---

[2] Suppose that the sequence $\{x_k\}$ converges to x. We say that this sequence converges linearly at rate c, if $\lim_{k\to\infty} \frac{|x_{k+1}-x|}{|x_k-x|} = c$ [10, Sec. 9.3.1].

[3] Ties can be broken arbitrarily. In Algorithm 2, we choose the user with the smallest index. Specifically, when $\arg\max_{j\in\mathcal{N}} \alpha_j(t)$ returns a set of indices, we choose the minimum one.

Theorem 2: For any $\delta \geq \frac{N-1}{N-\sum_{j\in\mathcal{N}} \gamma_j^{cont}}$, if each user i runs the distributed LDF scheduling algorithm, then

- each user i's average throughput up to time t converges to its optimal operating point linearly at rate δ, namely $|(1-\delta)\sum_{\tau=0}^{t} \delta^{\tau} \cdot r_i^{\tau} - r_i^{\star}| \leq r_i^{max} \cdot \delta^{t+1}$;
- each user i fulfills its CQoS guarantee.

Proof: See Appendix B. ■

Algorithm 2 The Longest-Distance-First (LDF) scheduling algorithm run by user i.

Require: normalized operating points $\{r_j^{\star}/r_j^{max}\}_{j\in\mathcal{N}}$, discount factor δ
Initialization: $t = 0$, "distances" $\alpha_j(0) = r_j^{\star}/r_j^{max}$, $\forall j \in \mathcal{N}$
repeat
  Find the user $i^* \triangleq \min\{\arg\max_{j\in\mathcal{N}} \alpha_j(t)\}$
  if $i = i^*$ **then**
    Transmit at power level P_i^{max}
  end if
  Updates distances $\alpha_j(t+1)$ for all $j \in \mathcal{N}$ as follows:
  $\alpha_{i^*}(t+1) = \frac{\alpha_{i^*}(t)}{\delta} - (\frac{1}{\delta} - 1)$, $r'_j(t+1) = \frac{\alpha_j(t)}{\delta}$, $\forall j \neq i^*$
  $t \leftarrow t+1$
until $\varnothing$

Remark 2: Although the CQoS guarantees do not directly appear in Algorithm 2, they impose a constraint on the discount factor δ used in Algorithm 2. Theorem 2 proves that the algorithm, given a proper input of discount factor (namely $\delta \geq \frac{N-1}{N-\sum_{j\in\mathcal{N}} \gamma_j^{cont}}$), will construct a policy that fulfill the CQoS guarantees.

Theorems 1 and 2 establish the convergence results of our proposed scheme. Theorem 1 proves that the process of finding the optimal operating points converges in logarithmic time, and Theorem 2 proves that the LDF scheduling achieves the optimal operating points in logarithmic time. Hence, the overall convergence speed is fast. Moreover, Theorem 2 ensures that the CQoS guarantees are fulfilled.

As we have discussed before, a byproduct of the CQoS guarantees is the upper bounds on the transmission delays, which are provided in Theorem 3

Theorem 3: For any discount factor $\delta \geq \frac{N-1}{N-\sum_{j\in\mathcal{N}} \gamma_j^{cont}}$, if each user $i \in \mathcal{N}$ runs the distributed LDF scheduling algorithm, we have

- each user i's maximum transmission delay is upper bounded, namely $\sup_{t\geq 0} d_i^t(\boldsymbol{\pi}) \leq \frac{\log \gamma_i^{cont}}{\log \delta}$;
- at each time t, each user i's transmission delay is upper bounded, namely $d_i^t(\boldsymbol{\pi}) \leq \frac{\log \alpha_i(t)}{\log \delta}$, where $\alpha_i(t)$ is user i's distance from target at time t calculated in Algorithm 2.

Proof: See Appendix C. ■

Theorem 3 gives us the upper bound of the maximum transmission delay, as well as finer upper bounds of transmission delays at each time t based on the user's distances from target $\alpha_i(t)$ (calculated in Algorithm 2). Note that the upper bound on the maximum delay, namely $\log \gamma_i^{cont} / \log \delta$, is decreasing in the CQoS, because we have $\gamma_i^{cont} < 1$ and $\delta < 1$.

TABLE IV. COMPARISON OF COMPUTATIONAL COMPLEXITY.

Policy	Computational complexity
Constant policies [1]–[4]	NP-hard to find the optimal $\boldsymbol{p}^{const}$
Round-robin TDMA (cycle length L)	Offline: $\geq N^{L-N}$ policies to search Online: 0
Proposed	Offline: $N \cdot O(\log_2 1/\varepsilon)$ Online: $O(N)$

C. Computational Complexity and Message Exchange

We compare the computational complexity of the existing solutions and our proposed solution, and discuss the amount of message exchange in our solution.

1) Computational Complexity: For constant policies, finding the optimal power profile $\boldsymbol{p}^{const}$ is NP-hard in general [1]. This is due to the nonconvexity of the problem: the throughput function is not jointly concave in the power profile because of the interference. For round-robin TDMA policies, the number of policies to search is lower bounded by N^{L-N}. To ensure a good performance, the cycle length needs to be large, which means that the number of policies grows exponentially with the number of users N. Hence, it may take a long time to find the optimal round-robin TDMA policy before run-time, although they are easy to implement at run-time. In contrast, in our proposed solution, the OOPS algorithm converges in logarithmic time before run-time, and the complexity of the online LDF scheduling is low (i.e. each user only needs to update the distances based on simple analytical formula).

2) Message Exchange: In our proposed solution, the message exchange happens only before run-time. The total amount of message exchange (i.e. the broadcast of r_i^*/r_i^{max}) is $N \cdot O(\log_2 1/\varepsilon)$. There is no message exchange at run-time.

VII. SIMULATION RESULTS

We demonstrate the performance gain of our proposed TDMA policy over existing policies. Throughout this section, we use the following system parameters. The noise powers at all the users' receivers are normalized as 0 dB. The maximum transmit powers of all the users are 20 dB. We normalize the direct channel gains to 1, namely $g_{ii} = 1, \forall i$, and generate the cross channel gains randomly according to the distribution $g_{ij} \sim \mathcal{CN}(0, 0.5), \forall i \neq j$. The system performance is measured by the (normalized) max-min fairness $\min_i R_i/r_i^{max}$, namely we aim to maximize the worst user's (normalized) throughput. At the optimal max-min fairness, each user's normalized average throughput $R_i(\boldsymbol{\pi})/r_i^{max}$ cannot exceed $\frac{1}{N}$. Hence, we let each user's AQoS guarantee to be within 10% of its maximum normalized throughput, namely $\gamma_i^{avg} = \frac{0.9}{N}, \forall i$. In most simulations, we will vary the CQoS guarantees (which are equal cross users). Given each CQoS guarantee, we choose the minimum discount factor specified by Theorem 2, namely $\delta = \frac{N-1}{N(1-\gamma_i^{cont})}$. In other words, we evaluate the performance of the most delay-sensitive applications.

We first fix the number of users to be $N = 4$, and increase the CQoS guarantees from 0.1 to 0.22. Note that the AQoS guarantee under $N = 4$ is 0.225. Hence, a CQoS guarantee of 0.22 is close to the AQoS guarantee. In Fig. 4, we show

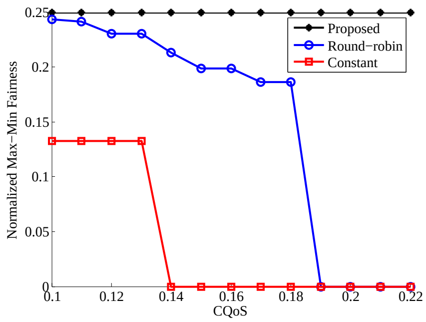

Fig. 4. Comparison of the max-min fairness achieved by different policies under different CQoS guarantees.

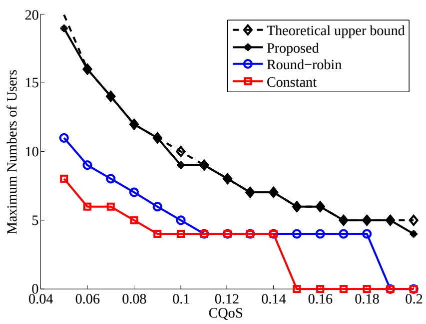

Fig. 5. Comparison of the maximum number of users that can be supported by different policies under different CQoS guarantees.

the optimal max-min fairness (i.e. $\min_i R_i/r_i^{max}$) achieved by different policies. We can see that under all CQoS guarantees, our proposed policy can achieve the optimal max-min fairness of 0.25. In contrast, the optimal constant policy is at least 50% away from the optimal max-min fairness when the CQoS guarantee is small, and becomes infeasible when the CQoS guarantee exceeds 0.13. For round-robin policies, we search all the 186480 non-trivial policies up to cycle length 9 and choose the optimal one under each CQoS guarantee. We can see that the performance of round-robin policies decreases to 20% away from the optimal performance before it becomes infeasible at CQoS of 0.19.

Next, we investigate the maximum number of users that can be supported by each policy under different CQoS guarantees. We increase the CQoS guarantees from 0.05 to 0.20. Note that theoretically, the maximum number of users that can possibly be supported is $\lfloor \frac{1}{\gamma_i^{cont}} \rfloor$ (because we need $N \cdot \gamma_i^{cont} \leq 1$). In Fig. 5, we can see that the maximum numbers of users supported by our proposed policy are the same as the theoretical upper bounds most of the time. In contrast, the other two policies can support much fewer users. At a CQoS guarantee of 0.05, we roughly double the number of users accommodated by the other two policies. Hence, we can utilize the spectrum much more efficiently while fulfilling CQoS guarantees.

Finally, we evaluate the performance of different polices for wireless video transmission. In the performance evaluation, we use the PSNR, which is commonly-used as performance metric for video quality. In the experiment, We consider a network with 4 users, and use the classic "Coastguard" video sequence. In Table V, we show the worst-case PSNR achieved by different policies under different CQoS guarantees. We can see that our proposed policy improves the PSNR of the constant policy and the round-robin policy by up to 6 dB and 4 dB, respectively. Moreover, when the CQoS guarantees increase, the other two policies become infeasible.

TABLE V. IMPROVEMENT OF PSNR OVER CONSTANT AND ROUND-ROBIN POLICIES UNDER DIFFERENT CQOS GUARANTEES.

CQoS guarantee	0.12	0.15	0.18	0.20
Constant	29 dB	infeasible	infeasible	infeasible
Round-robin	34 dB	32 dB	32 dB	infeasible
Proposed	36 dB	36 dB	36 dB	36 dB
Improvement over Constant	6 dB	–	–	–
Improvement over Round-robin	1 dB	3 dB	4 dB	–

VIII. CONCLUSION

In this paper, we studied spectrum sharing among users with delay-sensitive applications. We proposed a novel performance metric, namely continuing QoS guarantees, to ensure the performance of delay-sensitive applications. We designed the optimal TDMA policy that maximizes the system performance subject to the CQoS guarantees, and proposed low-complexity distributed algorithms for the users to construct the optimal policy. Our proposed policy significantly outperforms existing constant policies and round-robin policies, in terms of the system performance (e.g. max-min fairness), the number of users accommodated while fulfilling their QoS guarantees, as well as the computational complexity of designing the optimal policies. When applied to video streaming, our proposed policy can achieve performance improvement of up to 6 dB and 4 dB, compared to constant policies and round-robin policies.

APPENDIX A
PROOF OF THEOREM 1

Since W is strictly concave in $(r_1, \ldots, r_N)$, and since the constraints are linear in $(r_1, \ldots, r_N)$, the optimization problem (8) is a convex optimization problem. Since $\sum_{i\in\mathcal{N}} \gamma_i^{avg} < 1$, the feasible set has a non-empty interior. Hence, Slater's condition and the strong duality holds for (8).

We solve (8) by looking at the KKT conditions. Write λ as the Lagrangian multiplier associated with the constraint $\sum_{i\in\mathcal{N}} r_i/r_i^{max} = 1$, and $\mu_i \geq 0$ as the Lagrangian multiplier associated with the constraint $r_i \geq \gamma_i^{avg} \cdot r_i^{max}$. The optimal $(r_1^*, \ldots, r_N^*)$ and the optimal λ^* and μ_i^* should satisfy the KKT conditions:

$$\left.\frac{\partial W}{\partial r_i}\right|_{r_i=r_i^*} + \mu_i^* = -\lambda^*/r_i^{max}; \quad (9)$$

$$\textstyle\sum_{i\in\mathcal{N}} r_i^*/r_i^{max} = 1; \quad (10)$$

$$\mu_i^* \cdot (r_i^* - \gamma_i^{avg} \cdot r_i^{max}) = 0; \quad (11)$$

$$\mu_i^* \geq 0; \quad (12)$$

$$r_i^* \geq \gamma_i^{avg} \cdot r_i^{max}. \quad (13)$$

We solve for $(r_1^*, \ldots, r_N^*)$, λ^*, and μ_i^* that satisfy the above KKT conditions as follows. For any given λ, we define $\hat{r}_i(\lambda)$ as the unique solution to the following equation:

$$\left.\frac{\partial W}{\partial r_i}\right|_{r_i=\hat{r}_i(\lambda)} = -\lambda^*/r_i^{max}.$$

The solution $\hat{r}_i(\lambda)$ is unique, because W is strictly concave and hence $\frac{\partial W}{\partial r_i}$ is strictly decreasing in r_i. We then let

$$r_i(\lambda) = \max\left\{\hat{r}_i(\lambda),\ \gamma_i^{avg} \cdot r_i^{max}\right\},$$

such that (13) in the KKT conditions is satisfied.

Again, due to the strict concavity of W, the solution $\hat{r}_i(\lambda)$ is strictly increasing in λ. Hence, $r_i(\lambda)$ is non-decreasing in λ. As a result, we can find λ^* such that

$$\textstyle\sum_{i\in\mathcal{N}} r_i(\lambda^*)/r_i^{max} = 1.$$

We can see that the above λ^* and the associated $r_i^* = r_i(\lambda^*)$ satisfy (10) and (13) in the KKT conditions.

We determine the optimal μ_i^* as follows. If $r_i^* = \hat{r}_i(\lambda)$, namely $\left.\frac{\partial W}{\partial r_i}\right|_{r_i=r_i^*} = -\lambda^*/r_i^{max}$, we let $\mu_i^* = 0$. If $r_i^* = \gamma_i^{avg} \cdot r_i^{max} > \hat{r}_i(\lambda)$, we let

$$\mu_i^* = -\lambda^*/r_i^{max} - \left.\frac{\partial W}{\partial r_i}\right|_{r_i=r_i^*}.$$

Since $r_i^* > \hat{r}_i(\lambda)$, due to strict concavity of W, we have

$$\left.\frac{\partial W}{\partial r_i}\right|_{r_i=r_i^*} < \left.\frac{\partial W}{\partial r_i}\right|_{r_i=\hat{r}_i(\lambda)} = -\lambda^*/r_i^{max}.$$

Hence, we have $\mu_i^* > 0$ when $r_i^* = \gamma_i^{avg} \cdot r_i^{max} > \hat{r}_i(\lambda)$. As a result, we ensure that (9), (11), and (12) in the KKT conditions are satisfied.

In summary, based on the above procedure, we can find $(r_1^*, \ldots, r_N^*)$, λ^*, and μ_i^* that satisfy the KKT conditions. Hence, the thoughput $(r_1^*, \ldots, r_N^*)$ are the optimal solution to (8).

Algorithm 1 exactly implements the above procedure. Note that since $r_i(\lambda)$ is non-decreasing in λ, we can use the bisection method to find the optimal λ^*. Hence, Algorithm 1 converges linearly with rate $\frac{1}{2}$.

APPENDIX B
PROOF OF THEOREM 2

The proof consists of three parts. We first prove that the optimal operating point $(r_1^*, \ldots, r_N^*)$ determined by Algorithm 1 is achievable when $\delta \geq \frac{N-1}{N-\sum_{i=1}^N \gamma_i^{cont}}$. Then we prove that the scheduling computed by Algorithm 2 achieves $(r_1^*, \ldots, r_N^*)$ and satisfies the CQoS guarantees. Finally, we prove the rate of convergence.

1) Part 1: As in dynamic programming, we can decompose each user i's discounted average throughput into the current throughput and the *continuation throughput* as follows:

$$\begin{aligned} R_i(\boldsymbol{\pi}) &= (1-\delta)\textstyle\sum_{t=0}^{\infty} \delta^t \cdot (\mathbf{1}_{\{\pi_i(t)>0\}} \cdot r_i^{max}) \\ &= (1-\delta) \cdot \underbrace{(\mathbf{1}_{\{\pi_i(0)>0\}} \cdot r_i^{max})}_{\text{the current throughput at } t=0} \\ &\quad + \delta \cdot \underbrace{\left[(1-\delta)\textstyle\sum_{t=1}^{\infty} \delta^{t-1} \cdot (\mathbf{1}_{\{\pi_i(t)>0\}} \cdot r_i^{max})\right]}_{\text{the continuation throughput starting from } t=1}. \end{aligned}$$

We can see that the continuation throughput starting from $t=1$ is the discounted average throughput as if the system starts from $t=1$. In general, we define user i's continuation throughput starting from t as

$$R_i^t(\boldsymbol{\pi}) \triangleq (1-\delta) \sum_{\tau=t}^{\infty} \delta^{\tau-t} \cdot (\mathbf{1}_{\{\pi_i(\tau)>0\}} \cdot r_i^{max}).$$

Then the decomposition at time t can be written as

$$R_i^t(\boldsymbol{\pi}) = (1-\delta) \cdot (\mathbf{1}_{\{\pi_i(t)>0\}} \cdot r_i^{max}) + \delta \cdot R_i^{t+1}(\boldsymbol{\pi}).$$

Write the continuation throughput vector as

$$\boldsymbol{R}^t = (R_1^t, \ldots, R_N^t).$$

Definition 1 (Self-generating set): A set of throughput vectors $\mathcal{R}$ is a self-generating set, if for any throughput vector $\boldsymbol{R} \in \mathcal{R}$, there exists a $i^* \in \mathcal{N}$ and a continuation throughput vector $\boldsymbol{R}' \in \mathcal{R}$ such that for all $i \in \mathcal{N}$, we have

$$R_i = (1-\delta) \cdot (\mathbf{1}_{\{i=i^*\}} \cdot r_i^{max}) + \delta \cdot R_i'.$$

An important property of the self-generating set, proved in [12], is that any throughput vector in $\mathcal{R}$ can be achieved by a TDMA protocol. This is because for any throughput vector $\boldsymbol{R} \in \mathcal{R}$, we can schedule a user i^* to transmit in the current time slot, and the resulting continuation throughput vector $\boldsymbol{R}'$ starting from the next time slot can be decomposed (by a user to transmit and the following continuation throughput vector) again. We can do the above decomposition iteratively to determine the transmission schedule.

Consider the following set of throughput vectors

$$\mathcal{R} = \left\{\boldsymbol{R} : \sum_{i\in\mathcal{N}} \frac{R_i}{r_i^{max}} = 1, R_i \geq \gamma_i^{cont} \cdot r_i^{max}, \forall i\right\}.$$

We derive the condition on the discount factor δ such that $\mathcal{R}$ is self-generating. For a given vector $\boldsymbol{R} \in \mathcal{R}$, if we let user i to transmit, the continuation throughput vector $\boldsymbol{R}'$ is

$$R_i' = \frac{R_i}{\delta} - \frac{1-\delta}{\delta} \cdot r_i^{max}, \text{ and } R_j' = \frac{R_j}{\delta},\ \forall j \neq i. \tag{14}$$

To ensure $\boldsymbol{R}' \in \mathcal{R}$, the discount factor must satisfy

$$\delta \geq \frac{r_i^{max} - R_i}{r_i^{max} - \gamma_i^{cont} \cdot r_i^{max}} = \frac{1 - R_i/r_i^{max}}{1-\gamma_i^{cont}}.$$

Hence, to ensure that any $\boldsymbol{R} \in \mathcal{R}$ can be decomposed, the discount factor must satisfy

$$\delta \geq \max_{\boldsymbol{R}\in\mathcal{R}} \min_{i\in\mathcal{N}} \frac{1 - R_i/r_i^{max}}{1-\gamma_i^{cont}}. \tag{15}$$

The solution $\boldsymbol{R}^*$ to $\max_{\boldsymbol{R}\in\mathcal{R}} \min_{i\in\mathcal{N}} \frac{1-R_i/r_i^{max}}{1-\gamma_i^{cont}}$ should satisfy

$$\frac{1 - R_i^*/r_i^{max}}{1-\gamma_i^{cont}} = C,$$

where C is some constant. From $\sum_{i\in\mathcal{N}} R_i^*/r_i^{max} = 1$, we can solve for the constant C, and the optimal R_i^* as

$$R_i^* = r_i^{max} \cdot \left(1 - \frac{N-1}{N - \sum_{i\in\mathcal{N}} \gamma_i^{cont}} \cdot (1-\gamma_i^{cont})\right).$$

Then we have

$$\max_{\boldsymbol{R}\in\mathcal{R}} \min_{i\in\mathcal{N}} \frac{1 - R_i/r_i^{max}}{1-\gamma_i^{cont}} = \frac{N-1}{N - \sum_{i\in\mathcal{N}} \gamma_i^{cont}}.$$

In summary, when $\delta \geq \frac{N-1}{N-\sum_{i\in\mathcal{N}}\gamma_i^{cont}}$, any $\boldsymbol{R} \in \mathcal{R}$ is achievable. Since the optimal solution $(r_1^*,\ldots,r_N^*)$ obtained by Algorithm 1 satisfies

$$\sum_{i\in\mathcal{N}} \frac{r_i^*}{r_i^{max}} = 1,$$

and

$$r_i^* \geq \gamma_i^{avg} \cdot r_i^{max} > \gamma_i^{cont} \cdot r_i^{max}, \forall i.$$

We have that $(r_1^*,\ldots,r_N^*) \in \mathcal{R}$. Hence, we can achieve the optimal operating point $(r_1^*,\ldots,r_N^*)$ when $\delta \geq \frac{N-1}{N-\sum_{i\in\mathcal{N}}\gamma_i^{cont}}$.

2) Part 2: Now we show that the LDF scheduling in Algorithm 2 produces a schedule that achieves $(r_1^*,\ldots,r_N^*)$ and satisfies the CQoS guarantees. This proof is closely related to the proof of Part 1.

Recall that for each continuation throughput vector $\boldsymbol{R}^t$ at time t, if we choose user i to transmit, we can calculate the resulting continuation throughput vector $\boldsymbol{R}^{t+1}$ at time $t+1$ as in (14). The proof of Part 1 ensures that as long as we choose the user to transmit at time t based on

$$i = \arg\min_{j\in\mathcal{N}} \{1 - R_j(t)/r_i^{max}\} = \arg\max_{j\in\mathcal{N}} R_j^t/r_i^{max}$$

(see (15)), the continuation throughput vector $\boldsymbol{R}^{t+1}$ at time $t+1$ will also be achievable. This is why we define the distance as

$$\alpha_j(t) = R_j^t/r_i^{max}.$$

The LDF scheduling schedules the transmission exactly in this way in each time slot. By setting the continuation throughput at time 0 as $R_i^0 = r_i^*$, namely $\alpha_i(0) = r_i^*/r_i^{max}$, each user i can achieve the optimal throughput r_i^*.

In addition, when $\delta \geq \frac{N-1}{N-\sum_{i\in\mathcal{N}}\gamma_i^{cont}}$, from the proof of Part 1, we know that every continuation payoff $\boldsymbol{R}^t \in \mathcal{R}$. Hence, we have

$$R_i^t \geq \gamma_i^{cont} \cdot r_i^{max},\ \forall i.$$

In other words, the CQoS guarantees are satisfied.

3) Part 3: Now we prove the rate of convergence.

We have

$$\begin{aligned} r_i^* &= (1-\delta)\sum_{\tau=0}^{\infty}\delta^\tau \cdot (\mathbf{1}_{\pi_i(\tau)>0} \cdot r_i^{max}) \\ &= (1-\delta)\sum_{\tau=0}^{t}\delta^\tau \cdot (\mathbf{1}_{\pi_i(\tau)>0} \cdot r_i^{max}) \\ &+ (1-\delta)\sum_{\tau=t+1}^{\infty}\delta^\tau \cdot (\mathbf{1}_{\pi_i(\tau)>0} \cdot r_i^{max}). \end{aligned}$$

Since

$$\begin{aligned} 0 &\leq (1-\delta)\sum_{\tau=t+1}^{\infty}\delta^\tau \cdot (\mathbf{1}_{\pi_i(\tau)>0} \cdot r_i^{max}) \\ &\leq (1-\delta)\sum_{\tau=t+1}^{\infty}\delta^\tau \cdot r_i^{max} = \delta^{t+1} \cdot r_i^{max}, \end{aligned}$$

we have

$$\begin{aligned} &\left|(1-\delta)\sum_{\tau=0}^{t}\delta^\tau \cdot r_i^\tau - r_i^*\right| \\ = &\left|(1-\delta)\sum_{\tau=0}^{t}\delta^\tau \cdot (\mathbf{1}_{\pi_i(\tau)>0} \cdot r_i^{max}) - r_i^*\right| \\ \leq\ & r_i^{max} \cdot \delta^{t+1}. \end{aligned}$$

Appendix C
Proof of Theorem 3

At any time t, the continuation throughput is

$$R_i^t = (1-\delta)\sum_{\tau=t}^{\infty}\delta^{\tau-t} \cdot (\mathbf{1}_{\pi_i(\tau)>0} \cdot r_i^{max}).$$

Since $\alpha_i(t) = R_i^t/r_i^{max}$, we have

$$\alpha_i(t) = (1-\delta)\sum_{\tau=t}^{\infty}\delta^{\tau-t} \cdot \mathbf{1}_{\pi_i(\tau)>0}.$$

Suppose that the delay at time t is d_i^t. Then we have

$$\begin{aligned} \alpha_i(t) &= (1-\delta)\sum_{\tau=t}^{t+d_i^t}\delta^{\tau-t} \cdot 0 \\ &+ (1-\delta)\sum_{\tau=t+d_i^t}^{\infty}\delta^{\tau-t} \cdot \mathbf{1}_{\pi_i(\tau)>0} \\ &= (1-\delta)\sum_{\tau=t+d_i^t}^{\infty}\delta^{\tau-t} \cdot \mathbf{1}_{\pi_i(\tau)>0} \\ &\leq (1-\delta)\sum_{\tau=t+d_i^t}^{\infty}\delta^{\tau-t} \cdot 1 \\ &= \delta^{d_i^t}. \end{aligned}$$

Note that $\alpha_i(t) < 1$. Hence, we have

$$d_i^t \leq \log_\delta \alpha_i(t) = \frac{\log \alpha_i(t)}{\log \delta}.$$

Since every $R_i^t \in \mathcal{R}$, we have $R_i^t \geq \gamma_i^{cont} \cdot r_i^{max}$ for all t. Hence, we have $\alpha_i(t) \geq \gamma_i^{cont}$ for all t. As a result, we have

$$d_i^t \leq \frac{\log \gamma_i^{cont}}{\log \delta},\ \forall t.$$

In other words, we have

$$\sup_{t\geq 0} d_i^t \leq \frac{\log \gamma_i^{cont}}{\log \delta},\ \forall i.$$

References

[1] C. W. Tan and S. H. Low, "Spectrum management in multiuser cognitive wireless networks: Optimality and algorithm," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 421-430, 2011.

[2] J. Huang, R. A. Berry, and M. L. Honig, "Distributed interference compensation for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074-1084, May 2006.

[3] S. Sorooshyari, C. W. Tan, M. Chiang, “Power control for cognitive radio networks: Axioms, algorithms, and analysis,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 878–891, 2012.

[4] R. Etkin, A. Parekh, and D. Tse, “Spectrum sharing for unlicensed bands,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 517–528, 2007.

[5] M. van der Schaar, Y. Andreopoulos, and Z. Hu, “Optimized scalable video streaming over IEEE 802.11 a/e HCCA wireless networks under delay constraints,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 755–768, June 2006.

[6] D. Pradas, M. A. Vazquez-Castro, “NUM-based fair rate-delay balancing for layered video multicasting over adaptive satellite networks,” *IEEE J. Sel. Areas in Commun.*, vol. 29, no. 5, May 2011.

[7] P. Dutta, A. Seetharam, V. Arya, M. Chetlur, S. Kalyanaraman, J. Kurose, “On managing quality of experience of multiple video streams in wireless networks,” in *Proc. IEEE Infocom*, 2012.

[8] Y. Xiao and M. van der Schaar, “Spectrum sharing policies for heterogeneous delay-sensitive users: A novel design framework,” in *Proc. Allerton*, 2013.

[9] M. van der Schaar and F. Fu, “Spectrum access games and strategic learning in cognitive radio networks for delay-critical applications,” *Proc. of IEEE, Special issue on Cognitive Radio*, vol. 97, no. 4, pp. 720–740, Apr. 2009.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univ. Press, 2004.

[11] Y. Xiao and M. van der Schaar, “Appendix,” Available at: http://www.seas.ucla.edu/~yxiao/AppendixGlobecom14a.pdf

[12] D. Abreu, D. Pearce, and E. Stacchetti, “Toward a theory of discounted repeated games with imperfect monitoring,” *Econometrica*, vol. 58, no. 5, pp. 1041–1063, 1990.