Application-Level Scheduling with Deadline Constraints

Huasen Wu*‡, Xiaojun Lin†, Xin Liu‡§, and Youguang Zhang*

*School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

†School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

‡Microsoft Research Asia, Beijing 100080, China

§Department of Computer Science, University of California, Davis CA 95616, USA

Abstract

Opportunistic scheduling of delay-tolerant traffic has been shown to substantially improve spectrum efficiency. To encourage users to adopt delay-tolerant scheduling for capacity-improvement, it is critical to provide guarantees in terms of completion time. In this paper, we study application-level scheduling with deadline constraints, where the deadline is pre-specified by users/applications and is associated with a deadline violation probability. To address the exponentially-high complexity due to temporally-varying channel conditions and deadline constraints, we develop a novel asymptotic approach that exploits the largeness of the network to our advantage. Specifically, we identify a lower bound on the deadline violation probability, and propose simple policies that achieve the lower bound in the large-system regime. The results in this paper thus provide a rigorous analytical framework to develop and analyze policies for application-level scheduling under very general settings of channel models and deadline requirements. Further, based on the asymptotic approach, we propose the notion of Application-Level Effective Capacity Region, i.e., the throughput region that can be supported subject to deadline constraints, which allows use to quantify the potential gain of application-level scheduling.

I. Introduction

Today's mobile Internet is facing a grand challenge to meet the exponentially increasing demand for mobile broadband services. However, *not all traffic is created equal*. While

some applications require instant access, many other applications may be willing to tolerate delay from minutes to hours [1, 2]. By opportunistically scheduling delay-tolerant transmission when the network condition is more favorable, we can significantly improve network utilization.

In this context, delay is a key performance metric that is directly tied to the users' overall experience. Unless the network can set a clear expectation for the completion time, the user may fear that his/her traffic could be delayed for too long. Therefore, providing predictable completion time is critical for encouraging users to adopt delay-tolerant scheduling for capacity improvement. In this paper, we consider a model where each transmission task is associated with a user- or application-specific deadline, which is the maximum delay that the application can tolerate and ranges from minutes to hours depending on the applications. The goal of the network is then to schedule as many users as possible before their deadlines. We refer to this problem as the *application-level scheduling* problem, and will discuss its differences from classical opportunistic scheduling [3–5] in detail in Section II.

When there is a single base-station, the above problem can be mapped to a single-server job scheduling problem with deadlines. When there is no channel variation, it is well-known that simple policies such as earliest-deadline-first (EDF) are optimal in underloaded systems [6]. Unfortunately, such a deadline-constrained scheduling problem is known to be extremely difficult when there are channel variations because of the difficult trade-off between serving more urgent users and serving users with better channel conditions. In the special case with two-state channels, variants of EDF have been proposed to deal with this trade-off, e.g., the Feasible-Earlier-Due-Date (FEDD) policy in [7] and the Earliest Positive-Debt Deadline First (EPDF) policy in [8]. However, for more general multi-state channel models, we are not aware of a tractable methodology to find optimal scheduling policies subject to strict deadline constraints.

Under such multi-state channel models, although recently-developed optimization-based approaches to wireless control have been very successful for maximizing long-term throughput and stability [9–11], they are of limited capability in maximizing capacity subject to deadline constraints. For instance, the Best-Rate (BR) [10] and Delay-driven MaxWeight [11] policies are shown to be throughput optimal for flow-level scheduling, but may perform poorly in the case with deadline constraints. Similarly, even though the Lyapunov-function based method developed in [12] can produce order-optimal capacity-delay tradeoffs, the attainable capacity at a finite deadline constraint could still be far from optimum [1]. Finally, stochastic

decision theory such as Markov Decision Process (MDP) can be used to solve the optimal decision subject to deadline constraints. However, as the number of users increases, such a stochastic decision problem is known to incur exponentially-high complexity (known as "*the curse of dimensionality*" for MDP).

In this paper, we develop a novel approach to this open problem. Our key idea is that when the system is large, significant simplicity will arise, which will enable us to develop simple policies that are close-to-optimal. In other words, instead of suffering from the curse-of-dimensionality when the problem size is large, we exploit the largeness of the system to our advantage. Specially, we consider the large-system regime where both the arrival rate and the capacity increase proportionally to infinity. We show that when the system size is large, with appropriately-designed scheduling policies, the interactions between users can be decoupled in two aspects: the deadline violation probability of each user is mainly determined by its own connectivity, while the traffic carrying capacity of the overall system is mainly determined by the *average* load aggregated over all users. Based on such insights, we can then design policies that are not only provably optimal in the large-system regime, but also perform very well for medium-sized systems.

For readers familiar with the large-system asymptotics [13], the intuition that the competition between users becomes less dominant in large-system regime may seem somewhat natural. However, as we will elaborate in the rest of the paper, when there is channel variation, it is non-trivial to design scheduling policies that correctly exploit this intuition. Specifically, if one simply generalizes policies from the case of no channel variations (e.g., EDF), these policies may in fact perform poorly even if the system size is large. In contrast, the results in this paper provide a rigorous analytical framework to develop and analyze the correct scheduling policies in such settings.

In summary, the main contributions of this paper are:

- We first present a lower bound on the deadline violation probability for application-level scheduling with deadline constraints under a given network capacity (Section III-A). Moreover, we show that this lower bound is tight in the large-system regime as it can be achieved by appropriately designed scheduling schemes. We note that this result holds under very general channel models that may have multiple transmission rates and even temporal correlation patterns.
- We then develop new scheduling policies, called Maximum-Total-On-users (MTO) and its work-conserving enhancement (MTO-WCE) (Sections III-B to III-C). They are not

only asymptotically optimal in the large-system regime, but also achieve superior performance for medium-sized systems. We demonstrate that it is non-trivial to design good policies, e.g., the variants of EDF and Delay-driven MaxWeight in fact perform poorly even when the system size is large.

- We generalize these results from single-class systems (SectionIII) to multi-class systems (Section IV), where the performance requirements of different classes can differ significantly. Further, based on the above asymptotic approach, we study the Application-Level Effective Capacity (ALEC), i.e., the maximum throughput that can be supported by the system with given requirements on the deadline violation probability (Section IV). We show that our proposed policies asymptotically achieve the optimal ALEC region. By evaluating the ALEC, we demonstrate the significant potential for capacity improvement thanks to application-level opportunistic control. Specifically, the ALEC varies greatly with the deadline constraint, e.g., by a factor more than 6 as shown in Section V.

II. System Model

A. Network and Traffic Models

Consider a wireless network with a single base-station (BS) operating in a time-slotted fashion, where $t \in \{0, 1, 2, \ldots\}$. We note that the time-slot length considered throughout this paper is typically much larger than that in the conventional opportunistic-scheduling schemes that leverage small-time-scale fading [3, 4]. There, each time-slot is on the order of milliseconds. In contrast, since the deadlines for application-level scheduling usually range from minutes to hours [14], we will use time-slot length of tens of seconds to a few minutes.

We focus on the downlink of the BS in this paper although the uplink can be studied similarly. The BS serves K classes of users. We assume that the arrival processes are stationary and ergodic, and independent across classes. Let $A_k(t)$ ($k = 1, 2, \ldots, K$) represent the number of class-k users that arrive during time-slot t. For ease of exposition, we focus on the case where for each class k, the arrival process $A_k(t)$ is a discrete-time Poisson process with mean $\lambda_k = \mathbb{E}\{A_k(t)\}$. Denote λ as the aggregated arrival rate, i.e, $\lambda = \sum_{k=1}^{K} \lambda_k$, and let α_k be the ratio of the load contributed by class-k users, i.e, $\alpha_k = \lambda_k/\lambda$.

Let $\mathcal{I} = \{1, 2, \ldots\}$ be the index set of all users that enter the system. Each user $i \in \mathcal{I}$ requests to download a file of size F_i. We assume that the file size F_i is known as soon as the task is created. For ease of exposition, in this paper we present the theoretical results assuming unit-size files, i.e., $F_i = 1$. However, we note that the results in this paper can

be easily extended to the scenario where users from the same class request files with i.i.d. random size, provided that the file sizes are independent of the channel processes. Further, all simulation results in Section V are based on random file sizes.

Associated with each class-k user is a (relative) deadline D_k, which is the maximum waiting time that a class-k user can tolerate. For example, for a class-k user arriving in time-slot t, its transmission task should be completed before $t+D_k$ (absolute deadline). Otherwise, user i will give up the task and depart the system.

B. Channel Model

For each $i \in \mathcal{I}$, let the channel state $S_i(t)$ represent the transmission rate (in units of bits/slot per unit of radio resource) available to user i in time-slot t. We model $S_i(t)$ as a Markov chain over a finite set of the possible transmission rates, i.e., $S_i(t) \in \{r_1, r_2, \ldots, r_J\}$, where J is the number of possible rates, and $0 = r_1 < r_2 < \ldots < r_J$.

We assume that users from the same class have the same transition probability matrix, which is given by

$$\boldsymbol{P}^{(k)} = \left[p^{(k)}_{j_1 j_2}\right]_{J\times J}, k = 1, 2, \ldots, K,$$

where $p^{(k)}_{j_1 j_2} \in [0, 1]$, $1 \leq j_1, j_2 \leq J$, is the transition probability from state j_1 to state j_2 for class-k users. In addition, we assume that channel processes are independent across users. Denote the stationary distribution for the Markov chain of class-k as $\boldsymbol{\pi}^{(k)} = [\pi^{(k)}_1, \pi^{(k)}_2, \ldots, \pi^{(k)}_J]$,where $\pi^{(k)}_j$ $(1 \leq j \leq J)$ is the stationary probability of state j. We assume that the channel processes have reached the steady state, i.e., with the stationary distribution $\boldsymbol{\pi}^{(k)}$, when transmission tasks are created.

C. Scheduling Model and Performance Objectives

At the beginning of each time-slot t, the BS makes scheduling decisions based on the network status. We define the system capacity C as the amount of available radio resource, which is the product of bandwidth and slot-length. We assume that when a user i is selected to transmit in time-slot t, its download task can be completed *within the given time-slot* using $F_i/S_i(t)$ units of resource. This assumption is reasonable since the time-slot length is much larger than that in packet-level scheduling. For example, if we take a time-slot of 30 seconds, as many as 3 Gbits (when the bandwidth is 20 MHz and the spectrum efficiency is 5 bps/Hz

[15]) can be transferred in a time-slot. Hence, for a medium file size of a few MBytes, these files can be easily completed in one time-slot provided that the channel condition is good.

Let $Q_k(t)$ represent the number of class-k users waiting for transmission. Note that in time-slot t, the users departing the system include the users being scheduled and the users violating their deadlines. Then, for each $k \in \{1, 2, \ldots, K\}$, the queue length $Q_k(t)$ evolves as follows

$$Q_k(t+1) = Q_k(t) - Z_k(t) - V_k(t) + A_k(t),$$

where $Z_k(t)$ and $V_k(t)$ represent the number of completed users and expired users in time-slot t, respectively. Let Γ be the set of all possible policies. Then for each policy $\gamma \in \Gamma$, the deadline violation probability of class k is defined as

$$v_{k,\gamma}(\boldsymbol{\lambda}, C) = \limsup_{T\to\infty} \frac{1}{\lambda_k T} \sum_{t=0}^{T-1} \mathbb{E}[V_k(t)],$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_K]$ is the arrival rate vector.

In a single-class system, we omit the class index for simplicity and denote the deadline violation probability by $v_\gamma(\lambda, C)$. The objective of the BS is to minimize the deadline violation probability subject to a given load level, i.e.,

$$\min_{\gamma\in\Gamma} \quad v_\gamma(\lambda, C).$$

In a multi-class system, the deadline violation probabilities across different classes are coupled and the BS needs to trade-off the performance of different classes. In this case, we are interested in the optimal region of the deadline violation probability, which is defined as follows.

Definition 1 (Optimal DVP region) *Given $\boldsymbol{\lambda}$ and C, the optimal region for the Deadline Violation Probability (optimal DVP region) is defined as the set of probability vectors that can be achieved under certain scheduling policy, i.e.,*

$$\mathcal{V}(\boldsymbol{\lambda}, C) = \big\{\boldsymbol{v} \in [0,1]^K : \exists \gamma \in \Gamma,$$
$$\textit{such that } v_{k,\gamma}(\boldsymbol{\lambda}, C) \le v_k \textit{ for all classes } k\big\}. \tag{1}$$

We are then interested in identifying the optimal DVP region and designing policies that can achieve any point in this region.

Remark: We note that *application-level scheduling* studied in this paper differs from typical packet-level and flow-level scheduling problems in literature. Our model differs from packet-level scheduling [3, 4] due to two reasons. First, the user population is dynamic (rather than fixed in [3, 4]) due to user arrivals and departures/expirations. Second, there is a difference in the time-scale that we are interested in. Specifically, packet-level scheduling focuses on the small-time-scale channel variations typically due to multi-path fading. In contrast, our application-level scheduling focuses on exploiting larger time-scale variations, which are typically due to shadowing and/or users moving further/closer to the BS. This difference can be seen by our choices of using larger time-slots and of completing a task in one time-slot. We emphasize that our model does not preclude the BS from using packet-level opportunistic scheduling schemes [3, 4] when serving the users *within* one time-slot, and we assume that the data rate $S_i(t)$ already captures such fast-time-scale opportunistic gains. Our model also differs from flow-level scheduling. In typical flow-level scheduling studies [10, 11, 16], flow-level dynamics and packet-level dynamics are mixed together, i.e., packet-level scheduling decisions must take into account flow-level statistics (e.g., delay or residual file size [11]). In contrast, our model can be viewed as a simplification that decouples flow-level scheduling from packet-level scheduling. The benefit of such simplification is that we can provide rigorous delay guarantees (in comparison, existing flow-level studies focus only on stability and throughput optimality [10, 11, 16]).

III. Scheduling in Single-Class Systems

In this section, we study the single-class case, i.e., $K = 1$, and omit the class index for simplicity. Recall that the BS aims to minimize the deadline violation probability for a given system capacity C and arrival rate λ. We first identify a lower bound on the deadline violation probability by studying an individual decision problem. Then, we propose asymptotically optimal policies, called MTO and MTO-WCE, which achieve the lower bound in the large-system regime, i.e., when C and λ proportionally grow to infinity.

A. A Lower Bound on the Deadline Violation Probability

To obtain a lower bound on the deadline violation probability, we first focus on the decision problem for an individual user: the user decides whether or not to request transmission in each time-slot based on its *waiting time* and *channel condition*. We will show that the optimal

performance obtained in such an individual decision problem provides a lower bound on the performance of network-scale scheduling.

Let $w \in \{0, 1, \ldots, D-1\}$ be the waiting time of the user, i.e., the number of time-slots that the user has waited in the system. Then, a request decision policy for the user can be represented by an *individual decision matrix* $\boldsymbol{x} = [x_{w,j}]_{D \times J}$, where $x_{w,j} \in [0, 1]$ ($w = 0, 1, \ldots, D-1$, $j = 1, 2, \ldots, J$) is the probability that the user requests transmission when its waiting time is w and its channel state is j. Let $\mathcal{X}$ be the set of all possible decision matrices. Corresponding to each $\boldsymbol{x} \in \mathcal{X}$, we define the following two metrics:

- *Silent probability* $p_0(\boldsymbol{x})$*:* the probability that the user does not request transmission within D slots.
- *Expected consumed resource* $c(\boldsymbol{x})$*:* the expected amount of resource consumed by the user if it ever requests transmission in some time-slot.

Let p_0^* be the optimal value of the following resource-constrained individual decision problem:

$$\begin{aligned} p_0^* = \quad & \min_{\boldsymbol{x} \in \mathcal{X}} \quad p_0(\boldsymbol{x}) \\ & \text{subject to} \quad c(\boldsymbol{x}) \leq \frac{C}{\lambda}. \end{aligned} \tag{2}$$

We note that the above problem (2) can be viewed as a constrained MDP and solved by a Lagrangian relaxation approach as in [17]. In particular, when the channel process is independent across time, the optimal solution can be shown to follow a threshold structure, i.e., for each given w, there exists a j_0 such that $x_{w,j} = 0$ for $j < j_0$, $x_{w,j} = 1$ for $j > j_0$, and $x_{w,j_0} \in [0, 1]$. In other words, in each time-slot t, the user only requests transmission when its data rate exceeds a certain threshold j_0. If $x_{w,j_0} \neq 0$ or 1, it corresponds to some randomization at the state j_0, which may be necessary to guarantee the equality of resource constraint in (2). This threshold j_0 may depend on the waiting time w. In the case when the channel is i.i.d. across time, the user will use larger threshold j_0 when w is small, and use a smaller threshold j_0 when w is large, i.e., when it is close to expiration. We refer the readers to [17] for the details of solving this constrained MDP problem.

Next, the following proposition states that p_0^* is a lower bound of the deadline violation probability.

Proposition 1 *Given system capacity C and arrival rate λ, the deadline violation probability under any scheduling policy γ satisfies $v_\gamma(\lambda, C) \geq p_0^*$.*

Remark: Note that in general, a multi-user system is complicated to analyze due to the coupling across the users. In other words, when one user requests transmission, the system may not have the capacity to accommodate it if there are many users requesting transmissions at the same time. However, a key insight from Proposition 1 is that, the deadline violation probability is bounded by each user's own random connectivity pattern, while the coupling across users is captured only through the *average* resource consumption $c(\boldsymbol{x})$. Intuitively, there are on average λ users that should be served in each time-slot, and hence the expected resource consumption of each user should not be larger than C/λ. Proposition 1 then shows that (2) indeed gives a lower bound on the minimum deadline violation probability.

Sketch of Proof: The scheduling problem of the whole system can be viewed as a MDP. Solving this network-scale MDP is extremely challenging as we discussed before, but we know that there exists an optimal stationary policy for this problem. Then, we bound its performance by showing that any network-scale stationary policy can be mapped to an individual decision policy subject to the constraint of (2). Details are available in Appendix A.

B. Achieving the Lower Bound in the Many-source Regime

In this section, we study scheduling policies that are asymptotically optimal in the large-system regime as the system capacity and arrival rate grow proportionally to infinity.

We consider the following semi-distributed framework. At the mobile-side, each user makes its own decision on whether or not to request transmission. As discussed in Section III-A, an individual decision policy is represented by a decision matrix $\boldsymbol{x} = [x_{w,j}]_{D\times J}$. Namely, for a user with waiting time w and channel condition j, it sends the transmission request with probability $x_{w,j}$. A user is referred to as an "ON" user when it sends the transmission request, and an "OFF" user, otherwise (Note that the notion of ON-OFF users is different from the notion of ON-OFF channels in [7]: the channel in this paper may still have multiple rate levels). Again, note that not all ON users can be served if there are too many of them requesting transmission at the same time. Hence, at the network-side, the scheduler needs to make decision for serving the "ON" users. Next, we show that the following *Maximum-Total-On-users (MTO)* policy performs very well when the system size is large and all users use appropriate $\boldsymbol{x}$.

Definition 2 (MTO policy) *In each time-slot, every user is considered for scheduling only when it is ON. Further, the BS serves the users such that the number of served ON users are*

maximized.

We represent the MTO policy as MTO($\boldsymbol{x}$), since it depends on the individual decision matrix $\boldsymbol{x}$ for each user. We note that the MTO($\boldsymbol{x}$) policy exhibits a number of highly desirable features for ease of implementation. First, each user determines its own individual decision matrix $\boldsymbol{x}$, possibly based on its future connectivity patterns. The BS does not need to know the connectivity patterns of each individual users. Second, to schedule which users should be served, the BS only needs to know the current channel conditions of those users who request transmissions. The BS does not need to track the state of all other users. Both features significantly reduce the amount of signalling overhead between the users and the BS.

Let $\boldsymbol{x}^*$ be the optimal solution to problem (2), we next show that MTO($\boldsymbol{x}^*$) (i.e., using $\boldsymbol{x}^*$ as the individual decision matrix) is asymptotically optimal in the large-system regime.

Proposition 2 *Fix $\bar{c} = C/\lambda$ and let $\boldsymbol{x}^*$ be the optimal solution of problem* (2)*. Then, MTO($\boldsymbol{x}^*$) is asymptotically optimal in the large-system regime, i.e.,*

$$\lim_{C\to\infty} v_{\mathrm{MTO}(\boldsymbol{x}^*)}(C/\bar{c}, C) = p_0^*, \tag{3}$$

and the convergence speed is at least $1/\sqrt{C}$.

The proposition indicates that, as the system capacity and the arrival rate grow proportionally to infinity, the deadline violation probability under MTO($\boldsymbol{x}^*$) approaches the lower bound. We note that this result is non-trivial because the lower bound in Proposition 1 implicitly assumes that all users requesting transmissions can be served immediately. However, due to randomness, not all ON users can be served even when the *average* total consumed resource is no greater than C. Fortunately, when the system size is large, this "fluctuation" effect becomes less critical. The proof is divided into two parts. First, we consider an even simpler policy, called FOO, that also has the same asymptotic properties. Then, we show that the MTO policy dominates the FOO policy with the same individual decision matrix, and thus has better performance.

1) A Baseline Policy: FOO

We first consider a policy that only serves those users requesting transmission for the first time after they arrive. Such a user is referred to as a "First-ON" user, and the corresponding policy is referred to as *First-On-Only (FOO)* policy.

Definition 3 (FOO policy) *Every user is considered for scheduling only once and only when the user requests transmission for the first time before they expire in D slots. In each time-slot, the BS serves as many "First-ON" users as possible.*

Similar to MTO($\boldsymbol{x}$), we represent the FOO policy as FOO($\boldsymbol{x}$), since it also depends on the individual decision matrix $\boldsymbol{x}$ for each user. We first consider a general individual decision matrix $\boldsymbol{x}$. Let $\rho(\boldsymbol{x}) = \lambda c(\boldsymbol{x})/C$ be the offered load level under $\boldsymbol{x}$. For a fixed offered load $\rho(\boldsymbol{x}) \leq 1$, we can show that in the large-system regime, almost all "First-ON" users can be served and the deadline violation probability under FOO($\boldsymbol{x}$) approaches the silent probability $p_0(\boldsymbol{x})$.

Lemma 1 *Fix the decision matrix $\boldsymbol{x}$ such that the load level satisfies $\rho(\boldsymbol{x}) \leq 1$. Under the FOO policy, the deadline violation probability approaches the silent probability as C grows to infinity, i.e.,*

$$\lim_{C\to\infty} v_{\mathrm{FOO}(\boldsymbol{x})}(C\rho(\boldsymbol{x})/c(\boldsymbol{x}), C) = p_0(\boldsymbol{x}), \tag{4}$$

and the convergence speed is at least $1/\sqrt{C}$.

Sketch of Proof: We prove the lemma by exploiting two critical properties of FOO. First, since each user is considered to be scheduled only when it is "First-ON", the candidate set for scheduling in each time-slot only depends on each user's own connectivity pattern. Second, FOO fully utilizes the resource to serve "First-ON" users in each time-slot. Using these two properties, we can show that as the system size increases, the probability that a user is "First-ON" but can not be served becomes negligible, with the convergence speed of at least $1/\sqrt{C}$ by the Central Limit Theorem.

More specifically, let Y_j ($j = 1, 2, \ldots, J$) be the number of "First-ON" users with channel state j. Because the arrival process is a discrete-time Poisson process, we can show that Y_j is a Poisson random variable with mean value $\pi_j^\sharp\lambda$, where $\lambda = C\rho(\boldsymbol{x})/c(\boldsymbol{x})$ and $\pi_j^\sharp$ is the probability that a user is "First-ON" and its channel state is j. Note that for $j = 1$, we have $Y_1 = 0$ because no user can request transmission when its data rate is $r_1 = 0$. For $j > 1$, by the Central Limit Theorem, we can show that Y_j will deviate from its mean value on the order of $\sqrt{\lambda}$. This implies that the expectation of the part of Y_j that exceeds its mean value is on the order of $\sqrt{\lambda}$, i.e., $\mathbb{E}[Y_j - \pi_j^\sharp\lambda]^+ = O(\sqrt{\lambda})$, and hence $\mathbb{E}\{\frac{[Y_j-\pi_j^\sharp\lambda]^+}{\lambda}\} = O(\frac{1}{\sqrt{\lambda}})$. Recall that $c(\boldsymbol{x})$ is the expected consumed resource if the user can be served when "First-ON". Hence, $\rho(\boldsymbol{x}) \leq 1$ indicates that the expected required resource for all "First-ON" users is

$\lambda c(\boldsymbol{x}) = \lambda \sum_{j=2}^{J} \pi_j^\sharp / r_j \leq C$. Therefore, we can show that most of the "First-ON" users will be served and the probability that a "First-ON" user is not served (due to too large value of $\sum_{i=1}^{J} Y_j$) cannot be larger than that $\frac{r_J \sum_{j=1}^{J} [Y_j - \pi_j^\sharp \lambda]^+}{r_2^2 C}$, which goes to 0 as $1/\sqrt{\lambda}$ as λ and C grow proportionally to infinity. Details are available in Appendix B.

2) Dominance of MTO and Proof of Proposition 2

The FOO policy only allows each user to request transmission *once* before its deadline. However, the proposed MTO policy removes this restriction and we can show that with the same individual decision matrix $\boldsymbol{x}$, the proposed MTO policy dominates FOO in any time-slot. Specifically, the candidate set of MTO is a superset of that of FOO in each time-slot, and thus the number of served users under MTO is no less than that under FOO in any time-slot. Therefore, Eq. (4) also holds for MTO($\boldsymbol{x}$). As a special case, when the individual decision matrix is $\boldsymbol{x}^*$, we have $\rho(\boldsymbol{x}^*) \leq 1$, and the conclusion of Proposition 2 then follows.

C. Work-Conserving Enhancement of MTO

Under MTO, resource may still be wasted if after serving all ON users, there is still capacity remaining. In this case, if we allow the BS to serve some of the OFF users, the MTO policy should perform even better. For example, consider the following policy called *MTO with Work-Conserving Enhancement (MTO-WCE)*. We consider another version of problem (2) where the constraint is relaxed to $c(\boldsymbol{x}) \leq (1+\xi)C/\lambda$, where $\xi > 0$ is a control factor that can be used for trading-off between the resource utilization and signaling overhead. We let $\boldsymbol{x}^{(\xi)}$ be the optimal solution to the relaxed individual decision problem (2). The users who request transmission based on $\boldsymbol{x}^*$ are still called ON users, and the users who request transmission based on $\boldsymbol{x}^{(\xi)}$ are called "secondary-ON" users. The MTO-WCE policy will serve the ON users first. If there is remaining capacity, the BS then serves the "secondary-ON" users. Clearly, MTO-WCE must achieve even better performance than MTO because we always serve the ON users first.

D. Comparison and Discussions

We briefly compare the above policies in Fig. 1. The detailed simulation setting will be given in Section V. In Fig. 1, we plot the deadline violation probability versus the system scale C in a single-class setting. As we can observe, FOO, MTO, and MTO-WCE approach the lower bound when the system size is large. However, FOO leads to a much

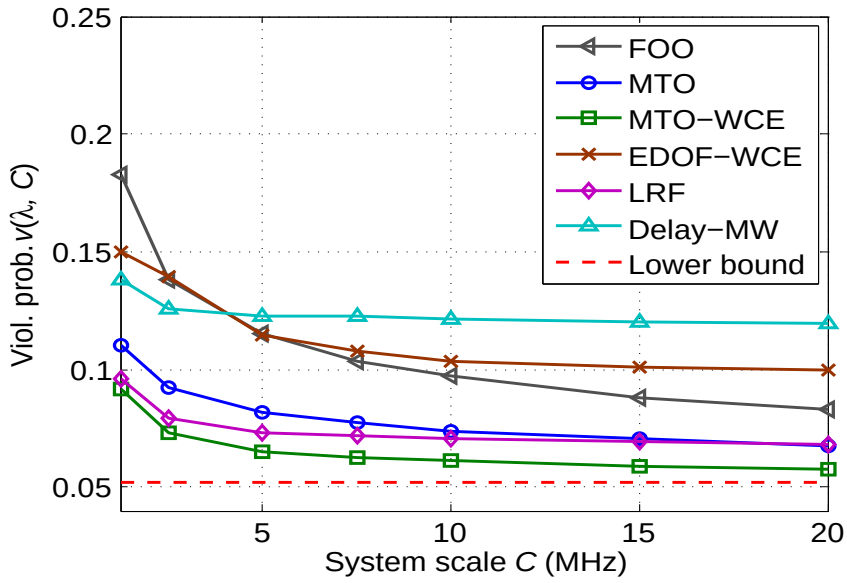


Fig. 1. Convergence of deadline violation probability ($D = 10$). Detailed settings for the file size and channel processes are presented in Section V.

larger violation probability in medium-sized system due to the restriction that we discussed earlier. Further, MTO-WCE outperforms MTO and reduces the violation probability even further. Other policies, such as Delay-driven MaxWeight (Delay-MW), may not approach the lower bound even when the system size is larger. Next, we discuss the implications of these observations.

1) Achieving asymptotic optimality is not trivial: Readers may have the impression that, since even a policy as simple as FOO achieves the same asymptotic optimality when the system size is large, perhaps any reasonable policy will be as good as MTO/MTO-WCE. This apparent triviality could be quite misleading. For example, consider a natural variant of EDF, called earliest-deadline-ON-user-first (EDOF), where in each time-slot, the BS serves ON users, i.e., those users requesting transmissions, according to the EDF discipline. As shown in Fig. 1, even for EDOF with work-conserving-enhancement (EDOF-WCE), the deadline violation probability is still larger than that under FOO and may not approach the lower bound p_0^* even when the system is large. What happens is that ON users closer to the deadline tend to request transmissions even with poor channels. Because EDOF prioritizes these users, it reduces the overall system performance. Another well-known policy, Delay-driven MaxWeight [11], does not approach the lower bound either when the system size is large, as shown in Fig. 1 (which is not surprising because Delay-driven MaxWeight is only throughput optimal but does not guarantee deadline performance). The above examples

therefore illustrate that, even when the system size is large, we must carefully design efficient scheduling policies based on rigorous theoretical principles, in particular, by choosing policies that dominate FOO.

2) Serving only the ON users (or secondary-ON users) is important: In the single-class case, one may envision other policies that do not rely on individual decision matrices $\boldsymbol{x}^*$ or $\boldsymbol{x}^{(\xi)}$. For example, consider the following Less-Resource-First (LRF) policy: all users are eligible for service and at each time-slot, and the BS gives priorities to the users that require less resource to be served. For identical-file-size systems, LRF can be viewed as a work-conserving enhancement of the Best-Rate policy in [10]. One can show that the LRF policy also dominates FOO and hence is asymptotically optimal in the single-class case. However, there are two reasons why MTO/MTO-WCE are more preferable than LRF. First, as we will see later in Section V, it is difficult to extend the LRF policy to the multi-class case because it is unable to balance the performance across different classes. In contrast, the MTO/MTO-WCE policies using the optimal individual decision matrices can be shown to be optimal in the multi-class case as well. Second, MTO and MTO-WCE incur much lower signalling overhead because only the ON (or secondary-ON) users need to report the channel state to the BS. In contrast, for LRF the BS needs to know the channel conditions of all users. Hence, there are both analytical and practical advantages to use MTO/MTO-WCE.

IV. Scheduling in Multi-Class Systems

In the previous section, we have shown that, when there is a single class, simple MTO and MTO-WCE policies are not only asymptotically optimal when the system size is large, but also perform well in medium-sized systems. In this section, we extend the results to multi-class systems.

In multi-class systems, the design of scheduling policies must be even more careful because we need to balance the performance across different classes. Due to the inter-class contention, it is impossible to simultaneously minimize the deadline violation probability of all classes. Thus, we turn to study the optimal DVP region (Definition 1). We will identify an outer bound for the optimal DVP region and show that MTO/MTO-WCE can asymptotically attain any point strictly inside the outer bound in the large-system regime. Further, we quantify the maximum throughput that can be supported for given requirement on the deadline violation probabilities, which will show the benefit of application-level scheduling.

A. Optimal DVP Region

For given system capacity C and arrival rate vector $\boldsymbol{\lambda}$, we define the optimal DVP region $\mathcal{V}(\boldsymbol{\lambda}, C)$ by Eq. (1). However, obtaining the accurate region of $\mathcal{V}(\boldsymbol{\lambda}, C)$ is difficult. Next, we will establish a simple outer bound for $\mathcal{V}(\boldsymbol{\lambda}, C)$, and show that an appropriately-designed MTO policy will attain this bound when the system size is large.

In order to obtain an outer bound for $\mathcal{V}(\boldsymbol{\lambda}, C)$, we first consider the scenario where each class is separately served with a certain proportion of the resource. Such separation allows us to use the results obtained in the single-class case. Specifically, let $\boldsymbol{\zeta} \in [0,1]^K$ satisfy $\sum_{k=1}^K \zeta_k = 1$, and let $\zeta_k C$ be the resource allocated to class k. By Proposition 1, we know that the lower bound on the deadline violation probability for each class is given by the optimal value $p^*_{0,k}(\zeta_k)$ of the following optimization problem:

$$\begin{aligned} p^*_{0,k}(\zeta_k) = \quad & \min\nolimits_{\boldsymbol{x}_k \in \mathcal{X}_k} \quad p_{0,k}(\boldsymbol{x}_k) \\ & \text{subject to} \quad c_k(\boldsymbol{x}_k) \leq \zeta_k C/\lambda_k. \end{aligned} \tag{5}$$

As a result, separating the resource according to $\boldsymbol{\zeta}$ should allow us to achieve any vector of deadline violation probability in $\{\boldsymbol{v} \in [0,1]^K : p^*_{0,k}(\zeta_k) \leq v_k \leq 1\}$. Taking the union of all possible $\boldsymbol{\zeta}$, we then obtain the following region:

$$\hat{\mathcal{V}}(\boldsymbol{\lambda}, C) = \bigcup_{\boldsymbol{\zeta} \in [0,1]^K, \sum_{k=1}^K \zeta_k = 1} \{\boldsymbol{v} \in [0,1]^K : p^*_{0,k}(\zeta_k) \leq v_k \leq 1\}.$$

Next, we will show that $\hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$ is an outer bound for the optimal DVP region $\mathcal{V}(\boldsymbol{\lambda}, C)$. Further, we will show that the MTO policy with appropriately chosen individual decision matrices is asymptotically optimal in attaining any vector of deadline violation probabilities in this outer bound when the system size is large. Specifically, suppose that we are given a vector $\boldsymbol{v} = [v_1, v_2, \ldots, v_k] \in \hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$. Let $\boldsymbol{x}^\sharp_k(v_k)$ be the optimal solution to the following individual decision problem:

$$\begin{aligned} & \min\nolimits_{\boldsymbol{x}_k \in \mathcal{X}_k} \quad c_k(\boldsymbol{x}_k) \\ & \text{subject to} \quad p_{0,k}(\boldsymbol{x}_k) \leq v_k. \end{aligned} \tag{6}$$

Further, let $\boldsymbol{x}^\sharp(\boldsymbol{v}) = \{\boldsymbol{x}^\sharp_1(v_1), \boldsymbol{x}^\sharp_2(v_2), \ldots, \boldsymbol{x}^\sharp_K(v_K)\}$. We represent the MTO policy with individual matrices $\boldsymbol{x}^\sharp(\boldsymbol{v})$ as MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$). In other words, under MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$), class-$k$ users request transmissions using matrix $\boldsymbol{x}^\sharp_k(v_k)$, and those users from each class requesting transmission are called ON users. As in Section III, in each time-slot, the MTO policy serves as many ON users as possible, regardless of which class they are from.

Proposition 3 *For given system capacity C and arrival rate vector $\boldsymbol{\lambda}$, the optimal DVP region satisfies $\mathcal{V}(\boldsymbol{\lambda}, C) \subseteq \hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$. In addition, for a fixed amount of average resource $\bar{c} = C/\lambda$, arrival proportion vector $\boldsymbol{\alpha}$, and any $\boldsymbol{v} \in \hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$, we have*

$$\lim_{C\to\infty} v_{k,\mathrm{MTO}(\boldsymbol{x}^\sharp(\boldsymbol{v}))}(C\boldsymbol{\alpha}/\bar{c}, C) \leq v_k, \tag{7}$$

Sketch of Proof: The proof for the outer bound is similar to the proof of Proposition 1. For any achievable vector $\boldsymbol{v}$ under a stationary policy, we can map the system dynamics to an individual decision matrix for each class and the corresponding value of $\boldsymbol{\zeta}$. Then, based on the overall resource constraints, we can show that the vector $\boldsymbol{v}$ must belong to $\hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$.

To show the asymptotic optimality of MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$), we can first show that with individual decision matrices $\boldsymbol{x}^\sharp(\boldsymbol{v})$, the offered load level must satisfies $\rho(\boldsymbol{x}^\sharp(\boldsymbol{v})) = \frac{1}{C}\sum_{k=1}^{K} \lambda_k c_k^\sharp(v_k) \leq 1$, where $c_k^\sharp(v_k)$ is the optimal value of problem (6). Otherwise, the vector $\boldsymbol{v}$ cannot be in $\hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$. Thus, we can show that the conclusion holds for FOO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) by the similar approach as in Lemma 1. Then we need to extend the results to MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$). However, the extension is trickier than the single-class case, because even though MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) dominates FOO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) in terms of *total number of served ON users*, it does not dominate FOO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) in terms of *number of served ON users for each class*. We need to prove the conclusion by further examining the upper bound of the number of served ON users for each class. Specifically, we note that the expected number of served users in each time-slot should not exceed an upper bound given by the expected number of ON users. Using this upper bound, we can then show that the deadline violation probability of each class under MTO will approach a value no greater than v_k. The details are available in Appendix C.

Remark: As we discussed earlier for the single-class case, a highly desirable feature of the MTO policy is that each user computes independently its decision matrix $\boldsymbol{x}_k$, and decides whether it should be ON or OFF in each time-slot. Then, the BS only needs to schedule as many ON users as possible. Note that the BS needs not to know the connectivity pattern of each user, nor its targeted deadline violation probability. Hence, the MTO policy is easy to implement in a distributed manner. Note also that the individual decision matrices $\boldsymbol{x}_k$s play a crucial role in balancing the performance requirements of different classes of users. Without such control, it would have been much more difficult for the BS to decide who should be served. As we will see in the simulation results in Section V, this difficulty is precisely why policies such as LRF, which performs well for single-class systems, fail in multi-class systems. In LRF (or in other weight-based policies such as Delay-driven MaxWeight), although one

can introduce and adjust weights to control the priority of different classes, it is difficult to predict the achieved deadline violation probabilities in advance, without actually running the policy. Hence, they are ineffective in guaranteeing the delay performance in the deadline-constrained scenarios that we are interested in.

We also note that, similar to the single-class case, we can use work-conserving enhancement to further improve the performance. Specifically, we can solve the problem (2) with relaxed resource constraint $c_k(\boldsymbol{x}) \leq (1+\xi)c_k(\boldsymbol{x}^\sharp(v_k))$, and use the solution to decide the "secondary-ON" users as in Section III-C.

B. Application-Level Effective Capacity Region

Instead of minimizing the deadline violation probability subject to given offered load, a dual problem would be to maximize the offered load subject to given deadline violation probabilities. Let η_k be the maximum deadline violation probability for class-k users. Then, in the single-class system, the objective of the BS is to maximize the throughput while guaranteeing that the deadline violation probability does not exceed η. We refer to this maximum throughput as *Application-Level Effective Capacity (ALEC)*, to differentiate it from the Effective Capacity concept proposed in [18]. In a multi-class system, the ALECs are again coupled across different classes. Therefore, with given requirement $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_K]$, we define the ALEC region as follows.

Definition 4 (ALEC region) *Given system capacity C and required value $\boldsymbol{\eta}$ of deadline violation probabilities, the ALEC region is defined as follows,*

$$\Lambda(\boldsymbol{\eta}, C) = \big\{\boldsymbol{\lambda} \in [0,\infty)^K : \exists \text{ policy } \gamma \in \Gamma, \\ \text{such that } v_{k,\gamma}(\boldsymbol{\lambda}, C) \leq \eta_k \text{ for all classes } k\big\} \tag{8}$$

Similar to the analysis of the optimal DVP region, we consider the outer bound for $\Lambda(\boldsymbol{\eta}, C)$. Define the following region:

$$\hat{\Lambda}(\boldsymbol{\eta}, C) = \big\{\boldsymbol{\lambda} \in [0,\infty]^K, \sum_{k=1}^{K} c_k^\sharp(\eta_k)\lambda_k \leq C\big\},$$

where $c_k^\sharp(\eta_k)$ is the optimal value of the constrained optimization problem (6), with the deadline violation probability v_k replaced by η_k. Clearly, $\hat{\Lambda}(\boldsymbol{\eta}, C)$ increases linearly in C. Using the approach in Section IV-A, we can show that $\hat{\Lambda}(\boldsymbol{\eta}, C)$ is an outer bound for $\Lambda(\boldsymbol{\eta}, C)$ and is tight in the large-system regime.

Proposition 4 *Given the system capacity C and the required values $\boldsymbol{\eta}$ of deadline violation probabilities, the ALEC region satisfies $\Lambda(\boldsymbol{\eta}, C) \subseteq \hat{\Lambda}(\boldsymbol{\eta}, C)$. In addition, for any $\boldsymbol{\lambda}$ that is inside the interior of $\hat{\Lambda}(\boldsymbol{\eta}, 1)$, we have*

$$\lim_{C\to\infty} v_{k,\mathrm{MTO}(\boldsymbol{x}^\sharp(\boldsymbol{\eta}))}(\boldsymbol{\lambda}C, C) \leq \eta_k. \tag{9}$$

As a special case of Proposition 4, we conclude that for the single-class case (i.e., $K = 1$), the ALEC is upper bounded by $C/c^\sharp(\eta)$ and it approaches this upper bound as C grows to infinity. By evaluating this ALEC in Section V, we will demonstrate the benefit of application-level scheduling.

V. Simulation Results

A. Simulation Setup

We evaluate the performance of the proposed mechanism with typical LTE parameters [15], which are summarized in Table I. Since we consider application-level scheduling, we focus on a large time-scale and set the time-slot length to be 30 seconds. The file size of each user follows truncated lognormal distribution with mean 2 Mbytes, standard deviation 0.72 Mbytes, and maximum size 5 Mbytes [19]. We generate the channel processes based on the random waypoint (RWP) mobility model [20]. Specifically, we estimate the 1-step transition probabilities of the channel process for the users traveling in the cell with RWP model with a velocity of 3 Km/h. Then, the transition probabilities are used to drive a Markov model that simulates channel realizations.

TABLE I

System Parameters

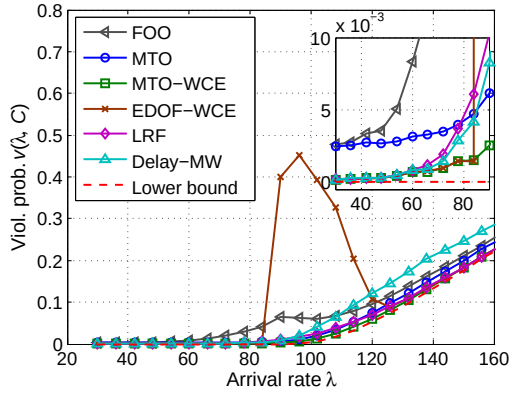
Property	Setting
Carrier frequency	2 GHz
System bandwidth	1.25, 2.5, 5, 7.5, 10, 15, 20 MHz
BS Tx power	46 dBm for 10 MHz
Coverage radius	500 m
Path loss	$128.1 + 37.6 \log_{10}(d[\mathrm{km}])$ dB,
Penetration loss	20 dB
Shadowing	Lognormal, standard deviation 8 dB
Noise power density	-170 dBm/Hz
Link adaption	Shannon's equation, clipped at -10 dB and 20 dB

We evaluate the deadline violation probability and ALEC under application-level scheduling with different disciplines. We use the optimal individual decision matrices for FOO, EDOF-WCE, and MTO/MTO-WCE. For the work-conserving enhancement, i.e., MTO-WCE and EDOF-WCE, the control factor is set to $\xi = 0.15$ (see the definition of ξ in Section III). We also compare with the LRF (see Section III-C) and Delay-driven MaxWeight (Delay-MW) [11] policies. In the multi-class case, weight vector is introduced in LRF and Delay-driven MaxWeight to trade-off between different classes. Specifically, the LRF policy prioritizes users according to $\zeta_k F_i/S_i(t)$, and Delay-driven MaxWeight prioritizes users according to $\zeta_k w_i S_i(t)$, where $0 \leq \zeta_k \leq 1$ reflects the additional weight of class-k users and $\sum_{k=1}^{K} \zeta_k = 1$.

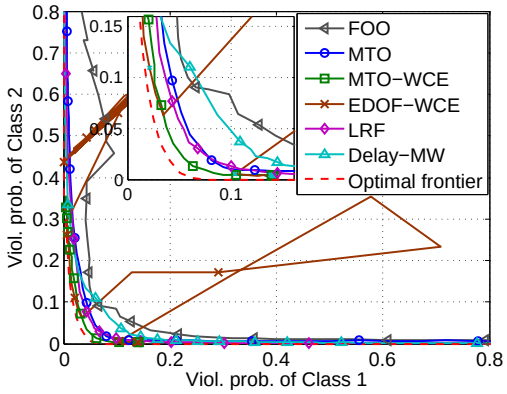
B. Deadline Violation Probability

Recall that we have evaluated the deadline violation probability versus the system size C in Fig. 1, when the relative load λ/C is fixed. Next, we evaluate the deadline violation probability with fixed C in Fig. 2. In the single-class case, Fig. 2(a) shows the deadline violation probability as a function of the arrival rate. The minimum silent probability given by (2) serves as a lower bound of the system, as stated in Proposition 1. We can observe that the deadline violation probability of MTO-WCE is very close to the lower bound and dominates all other policies in the whole range presented. When the load is light (e.g., $\lambda \leq 70$), all work-conserving policies achieve similar performance because the contention is low. However, as the load increases, the performance of different scheduling policies starts to differ. The MTO, MTO-WCE, and LRF policies perform very well, while other policies can perform significantly worse. For example, the deadline violation probability under Delay-driven MaxWeight can be much larger than that under MTO-WCE (by two times when $\lambda = 120$). The EDOF-WCE policy results in rather high deadline violation probability in the range of $85 < \lambda < 125$, likely due to the fact that the EDOF policy tends to serve users when their channel conditions are not favorable (refer to our discussions in Section III-D).

Fig. 2(b) shows the deadline violation probabilities for the 2-class scenario. For LRF and Delay-driven MaxWeight, each pair of deadline violation probabilities corresponds to a weight vector $\boldsymbol{\zeta}$. From the figure, we can see that the proposed MTO-WCE policy achieves close-to-optimal deadline violation probabilities. Because of the reasons discussed in Fig. 2(a), EDOF-WCE behaves strangely in certain regions. The deadline violation probability under LRF and Delay-MW is greater than that under MTO-WCE. Moreover, the exact impact of weight vector is unpredictable and difficult to tune in practice.



(a)



(b)

Fig. 2. Deadline violation probability, (a) single-class scenario with arrival rate ($C = 10$ MHz), (b) 2-class scenario ($C = 10$ MHz, $\boldsymbol{D} = [5, 15]$, and $\boldsymbol{\lambda} = [15, 20]$).

In summary, designing the optimal scheduling policies is non-trivial and some heuristic policies, e.g., EDOF, may perform rather poorly in certain range. The rigorous theoretical framework in this paper provides a principled approach to design and analyze the scheduling policies. Under this framework, the proposed MTO/MTO-WCE policies not only achieve the optimal bound in the large-system regime, but also perform well in medium-sized systems.

C. Application-Level Effective Capacity

In this section, we evaluate ALEC under different system sizes and requirements. The ALEC is normalized by the bandwidth and shown as spectrum efficiency (bps/Hz). Because MTO-WCE consistently outperforms FOO, MTO, and EDOF-WCE in earlier simulations, we will mainly use MTO-WCE in the rest of the simulations.

Fig. 3 shows the convergence of ALEC in a single-class system as the system size increases.

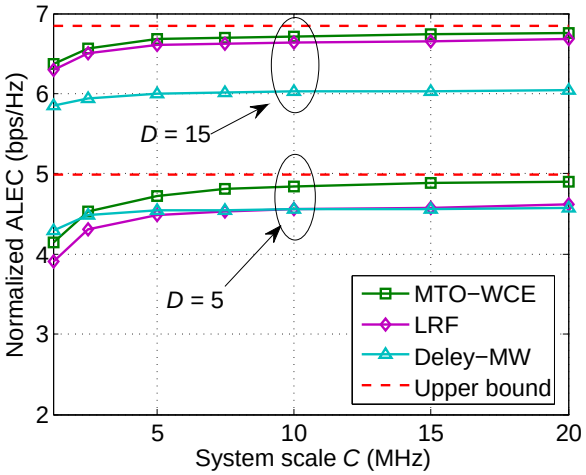
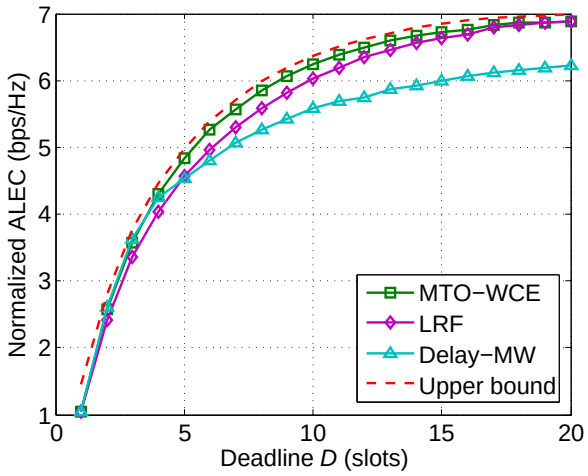
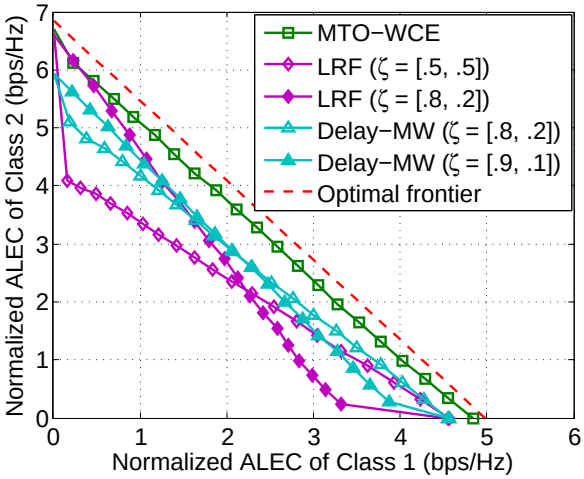


Fig. 3. Convergence of ALEC for different deadlines ($C = 10$ MHz, and $\eta = 0.05$).



(a)



(b)

Fig. 4. Evaluation of ALEC, (a) ALEC for single-class system with different deadlines ($C = 10$ MHz, and $\eta = 0.05$), (b) effective throughput region for 2-class system ($C = 10$ MHz, $\boldsymbol{D} = [5, 15]$, and $\boldsymbol{\eta} = [0.05, 0.05]$).

We can see that under MTO-WCE, the supportable traffic load approaches the upper bound stated in Proposition 4 (the dashed line). The gap from the upper bound is negligible when $C \geq 10$ MHz, which is a typical value of the bandwidth in cellular networks. Hence, we use $C = 10$ MHz for the rest of the simulation.

Fig. 4(a) shows the ALEC in a single-class system as a function of deadline. It clearly demonstrates the benefit of exploiting the delay tolerance of the traffic. Namely, the capacity can be significantly improved if the users can tolerate certain delay. For example, if users require to finish the transmission task within 1 slot (30 seconds), the spectrum efficiency is about 1 bps/Hz. However, with application-level scheduling, this efficiency can be increased to more than 6 bps/Hz if the users can tolerate a delay of 10 slots (5 minutes). Comparing to Delay-driven MaxWeight, we see that although MTO-WCE performs similarly to Delay-driven MaxWeight when the deadline is small, it clearly outperforms Delay-driven MaxWeight for larger deadlines. Comparing to the upper bound, we can see that the room for further improvement over the proposed MTO-WCE policy is very small.

Fig. 4(b) shows the ALEC region for a 2-class system. We can see that MTO-WCE achieves an ALEC region that is quite close to the outer bound. In contrast, for a given weight vector $\boldsymbol{\zeta}$, the ALEC regions under Delay-driven MaxWeight and LRF are smaller than that achieved by MTO-WCE. It is interesting to observe that, if we take the union of the ALEC region under LRF or Delay-driven MaxWeight over different choices of $\boldsymbol{\zeta}$, the union becomes closer to the optimal. However, in practice, it is difficult to predict the delay performance of LRF or Delay-driven MaxWeight in advance. As a result, it is difficult to tune the parameter $\boldsymbol{\zeta}$ for these algorithms under a given mixture of deadline-constrained traffic, without actually running the algorithms. Therefore, we believe that the theoretical results and our proposed MTO/MTO-WCE policies are particularly useful for multi-class systems with different deadline constraints.

VI. Conclusions and Future Work

In this paper, we study application-level scheduling mechanisms for delay-tolerant traffic with deadline requirements. The objective of the network is to minimize the deadline violation probability for given arrival traffic. We present a lower bound on the deadline violation probability, and develop simple threshold-based policies, MTO and MTO-WCE, that achieve the lower bound in the large-system regime, under general channel models and multiple classes. These schemes also perform well in medium-size systems. We note the insights

from the analysis are important in the design of scheduling policies as some commonly studied policies may not perform well in certain regimes. Further, based on the asymptotic approach, we propose estimation approach for the ALEC region. Numerical results show that under application-level scheduling, if users can tolerate certain delay, the capacity can be improved significantly. For example, the capacity can be increased by about 6 times if the users can tolerate a delay of 10 time-slots (e.g., 5 minutes).

Although the results in the paper focus on the single-cell settings, we believe that the key insights are applicable to more general settings. For future work, we will study how to generalize the algorithms and insights into multi-cell settings.

APPENDIX A
PROOF OF PROPOSITION 1

The scheduling problem of the whole system can be viewed as a MDP. Solving this network-scale MDP is extremely challenging as we discussed before, but we know that there exists an optimal stationary policy for this problem. Then, we bound its performance by showing that any network-scale stationary policy can be mapped to an individual decision policy subject to the constraint of (2).

Specifically, let $Q_{w,j}(t)$ denote the number of users with waiting time w and channel state j in time-slot t. Then, the scheduling problem can be viewed as a MDP, where the cost is the deadline violation probability and the state space is the set of possible values of $Q_{w,j}(t)$ ($w = 0, 1, \ldots, D-1$; $j = 1, 2, \ldots, J$). For such a MDP, there exists a stationary policy that minimizes the deadline violation probability. Note that corresponding to each stationary policy Π, there is a stationary distribution matrix,

$$\mathbf{\Phi} = [\phi_{w,j}]_{D \times J},$$

where $\phi_{w,j} \in [0, 1]$ represents the probability that a user is served when its waiting time is w and channel state is j. Hence, the deadline violation probability is $v(\lambda, C) = 1 - \sum_{w=0}^{D-1} \sum_{j=1}^{J} \phi_{w,j}$. In addition, because of the resource constraint, we must have $\lambda \sum_{w=0}^{D-1} \sum_{j=2}^{J} \frac{\phi_{w,j}}{r_j} \leq C$ ($\phi_{w,1} = 0$ since $r_1 = 0$).

On the other hand, by considering a scenario with infinite available resource, each feasible $\mathbf{\Phi}$ can be uniquely mapped to an individual decision matrix $\boldsymbol{x}$ as follows.

$\mathbf{\Phi} \rightarrow \boldsymbol{x}$ mapping: Consider a scenario where all users use an identical individual decision matrix $\boldsymbol{x}$. In addition, the available resource is infinite and any user sending the request can be served immediately. Let $\pi'_{w,j}$ be the ratio of users that still stay in the system after waiting w slots and is in channel state j. For $w = 0$, we know that this ratio is equal to the stationary distribution of the channel process, i.e., $\pi'_{0,j} = \pi_j$ ($j = 1, 2, \ldots, J$). Then, we can decide $x_{0,j}$ by solving $\pi'_{0,j} x_{0,j} = \phi_{0,j}$ (for $\pi'_{0,j} = 0$, we let $x_{0,j} = 0$, which will not affect the value of other variables). For $w > 0$, we can decide $x_{w,j}$ in an iterative manner. Specifically, after obtaining $x_{w,j}$, we can calculate $\pi'_{w+1,j}$ as

$$\pi'_{w+1,j} = \sum_{j'=1}^{J} (1 - x_{w,j'}) \pi'_{w,j'} p_{j'j},$$

and obtain $x_{w+1,j}$ by solving $\pi'_{w+1,j} x_{w+1,j} = \phi_{w+1,j}$.

Then, under individual decision matrix $\boldsymbol{x}$, the expected consumed resource and the silent probability are equal to their corresponding values under Π. Therefore, the expected consumed resource must satisfy $c(\boldsymbol{x}) = \sum_{w=0}^{D-1} \sum_{j=2}^{J} \frac{\phi_{w,j}}{r_j} \leq C/\lambda$ and the deadline violation probability satisfies $v(\lambda, C) = p_0(\boldsymbol{x}) \geq p_0^*$. Note that these expressions precisely correspond to the constraint and objective of problem (2). Hence, we conclude that $v(\lambda, C)$ must be greater than p_0^*.

APPENDIX B
PROOF OF LEMMA 1

We prove the lemma by exploiting two critical properties of FOO. First, since each user is considered to be scheduled only when it is "First-ON", the candidate set for scheduling in each time-slot only depends on each user's own connectivity pattern, which is independent across users. Second, FOO fully utilizes the resource to serve "First-ON" users in each time-slot. Using these two properties, we can show that as the system size increases, the probability that a user is "First-ON" but can not be served becomes negligible, with the convergence speed of at least $1/\sqrt{C}$ by the Central Limit Theorem.

Consider FOO($\boldsymbol{x}$), i.e., the FOO policy with individual decision matrix $\boldsymbol{x}$. As discussed above, we only need to focus on an arbitrary time-slot and will omit the slot index for simplicity. Let Y_j ($j = 1, 2, \ldots, J$) be the number of "First-ON" users with channel state j for a given individual decision matrix $\boldsymbol{x}$. Further, let $\phi_{w,j} \in [0, 1]$ represent the probability that a user is served in channel state j at w slots after their arrivals. Note that the "First-ON" users evolve from the users arriving in the past D slots. Using the property of Poisson variables, we know that Y_j is a Poisson random variable with mean value $\pi_j^\sharp \lambda$, where $\pi_j^\sharp = \sum_{w=0}^{D-1} \phi_{w,j}$ is the probability that a user is "First-ON" at channel state j within D slots. In addition, the expected required resource for all "First-ON" users is $\lambda c(\boldsymbol{x}) = \lambda \sum_{j=2}^{J} \pi_j^\sharp / r_j$, and the offered load level $\rho(\boldsymbol{x}) = \lambda c(\boldsymbol{x})/C \leq 1$. Note that the summation is calculated from $j = 2$ because $Y_1 = 0$ since a user cannot request transmission with zero data rate, i.e., r_1.

Next we show that since $\rho(\boldsymbol{x}) \leq 1$, the probability that a "First-ON" user is unserved due to overflow tends to 0 as λ and C grow proportionally to infinity, with the convergence speed at least $1/\sqrt{\lambda}$.

Let $L(Y_1, Y_2, \ldots, Y_J)$ be the number of ON users that are unserved due to overflow when the number of ON users at channel state j is Y_j. Note that the total amount of resource exceeding the system capacity satisfies $\left[\sum_{j=2}^{J} r_j^{-1} Y_j - C\right]^+ \leq \sum_{j=2}^{J} r_j^{-1} [Y_j - \pi_j^\sharp \lambda]^+$. Thus,

the number of drop users satisfies

$$\begin{aligned} L(Y_1, Y_2, \ldots, Y_J) &\leq \frac{\sum_{j=2}^{J} r_j^{-1}[Y_j - \pi_j^\sharp \lambda]^+}{r_J^{-1}} \\ &\leq \frac{r_J}{r_2} \sum_{j=2}^{J} [Y_j - \pi_j^\sharp \lambda]^+. \end{aligned} \tag{10}$$

On the other hand, when overflow occurs, i.e., $\sum_{j=2}^{J} r_j^{-1} Y_j > C$, then $\sum_{j=2}^{J} Y_j > C r_2$. Thus, we can bound the probability that a "First-ON" user is unserved as follows

$$\begin{aligned} p_{\text{unserved}}(C) &= \mathbb{E}\Big\{ \frac{L(Y_1, Y_2, \ldots, Y_J)}{\sum_{j=2}^{J} Y_j} \Big\} \\ &\leq \frac{r_J}{r_2^2} \mathbb{E}\Big\{ \frac{\sum_{j=2}^{J} [Y_j - \pi_j^\sharp \lambda]^+}{C} \Big\} \end{aligned} \tag{11}$$

For a Poisson random variable Y_j with mean value $\mathbb{E}[Y_j] = \pi_j^\sharp \lambda$, we have

$$\begin{aligned} \mathbb{E}[Y_j - \pi_j^\sharp \lambda]^+ &= \sum_{\pi_j^\sharp \lambda}^{\infty} \frac{(y - \pi_j^\sharp \lambda)(\pi_j^\sharp \lambda)^y e^{-\pi_j^\sharp \lambda}}{y!} \\ &= \pi_j^\sharp \lambda \mathbb{P}\big\{ \pi_j^\sharp \lambda - 1 \leq Y_j \leq \pi_j^\sharp \lambda \big\}. \end{aligned}$$

Hence,

$$p_{\text{unserved}}(C) \leq \frac{r_J}{\bar{c} r_2^2} \sum_{j=2}^{J} \pi_j^\sharp \mathbb{P}\big\{ \pi_j^\sharp \lambda - 1 \leq Y_j \leq \pi_j^\sharp \lambda \big\}. \tag{12}$$

On the other hand, for $2 \leq j \leq J$, Y_j can be viewed as the summation of $\pi_j^\sharp \lambda$ of i.i.d. Poisson random variables with mean value 1. Therefore, by the Central Limit Theorem, we know that as C grows to infinity (so does λ), then

$$\sqrt{\pi_j^\sharp \lambda} \big[\frac{Y_j}{\pi_j^\sharp \lambda} - 1 \big] \xrightarrow{dist} \mathcal{N}(0, 1). \tag{13}$$

Thus,

$$\begin{aligned} \mathbb{P}\big\{ \pi_j^\sharp \lambda - 1 \leq Y_j \leq \pi_j^\sharp \lambda \big\} &\approx \int_{-1/\sqrt{\pi_j^\sharp \lambda}}^{0} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \mathrm{d}y \\ &\leq \sqrt{\frac{\bar{c}}{2\pi \pi_j^\sharp C}}. \end{aligned}$$

Consequently,

$$p_{\text{unserved}}(C) \;\leq\; \frac{r_J}{\bar{c}r_2^2}\sum_{j=2}^{J}\sqrt{\frac{\bar{c}\pi_j^\sharp}{2\pi C}},$$

implying that as C grows to infinity, $p_{\text{unserved}}(C)$ converge to 0, with convergence speed of at least $1/\sqrt{C}$. The conclusion of Lemma 1 then follows.

APPENDIX C
PROOF OF PROPOSITION 3

A. Outer bound on the optimal DVP region

The proof for the outer bound is similar to the proof of Proposition 1. Consider any deadline violation probability $\boldsymbol{v}$ that is achievable, i.e., $\boldsymbol{v} \in \mathcal{V}(\boldsymbol{\lambda}, C)$. Using the similar approach as in Appendix A, we can map $\boldsymbol{v}$ to individual decision matrices $\boldsymbol{x}_k(v_k)$ $(k = 1, 2, \ldots, K)$ and the silent probability under $\boldsymbol{x}_k(v_k)$ is v_k. Also, corresponding to each $\boldsymbol{x}_k(v_k)$, there is an expected consumed resource $c_k(\boldsymbol{x}_k(v_k))$. Because the achievability of $\boldsymbol{v}$, we know that the total expected consumed resource satisfies the resource constraint, i.e., $\sum_{k=1}^K \lambda_k c_k(\boldsymbol{x}_k(v_k)) \leq C$. Next, let $\rho(\boldsymbol{v}) = \sum_{k=1}^K \lambda_k c_k(\boldsymbol{x}_k(v_k))/C$ and $\zeta_k = \frac{\lambda_k c_k(\boldsymbol{x}_k(v_k))}{C\rho(\boldsymbol{v})}$. Then, we have $\sum_{k=1}^K \zeta_k = 1$, and the solution of Problem (5) satisfies $p_{0,k}^*(\zeta_k) \leq v_k \leq 1$ because $\zeta_k C/\lambda \geq c_k(\boldsymbol{x}_k(v_k))$. Therefore, the vector $\boldsymbol{v}$ belongs to $\hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$ and hence $\mathcal{V}(\boldsymbol{\lambda}, C) \subseteq \hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$

B. Achieving the outer bound in the large-system regime with MTO

To show the asymptotic optimality of MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$), we can first show that with individual decision matrices $\boldsymbol{x}^\sharp(\boldsymbol{v})$, the offered load level must satisfies $\rho(\boldsymbol{x}^\sharp(\boldsymbol{v})) = \frac{1}{C}\sum_{k=1}^K \lambda_k c_k^\sharp(v_k) \leq 1$, where $c_k^\sharp(v_k)$ is the optimal value of problem (6). Otherwise, the vector $\boldsymbol{v}$ cannot be in $\hat{\mathcal{V}}(\boldsymbol{\lambda}, C)$. Thus, we can show that the conclusion holds for FOO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) by the similar approach as in Lemma 1. Then we need to extend the results to MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$). However, the extension is trickier than the single-class case, because even though MTO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) dominates FOO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) in terms of *total number of served ON users*, it does not dominate FOO($\boldsymbol{x}^\sharp(\boldsymbol{v})$) in terms of *number of served ON users for each class*. We need to prove the conclusion by further examining the upper bound of the number of served ON users for each class. Specifically, we note that the expected number of served users in each time-slot should not exceed an upper bound given by the expected number of ON users. Using this upper bound, we can then show that the deadline violation probability of each class under MTO will approach a value no greater than v_k.

Specifically, let $\bar{Z}_k$ be the expected number of class-k users receiving service before expiration, i.e.,

$$\bar{Z}_k = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Z_k(t)].$$

Note that for any $\boldsymbol{\zeta} \in [0,1]^K$ satisfying $\sum_{k=1}^K \zeta_k = 1$, we know that $\rho\big(\boldsymbol{x}^*(\boldsymbol{\zeta})\big) \leq 1$. Using the similar argument in the proof of Lemma 1, we know that

$$\lim_{C\to\infty} \frac{\bar{Z}_{k,\mathrm{FOO}}}{\lambda} = \alpha_k[1 - p^*_{0,k}(\zeta_k)],$$

and

$$\lim_{C\to\infty} \frac{\bar{Z}_{\mathrm{FOO}}}{\lambda} = \sum_{k=1}^K \alpha_k[1 - p^*_{0,k}(\zeta_k)],$$

where $\bar{Z}_{\mathrm{FOO}} = \sum_{k=1}^K \bar{Z}_{k,\mathrm{FOO}}$ is the expected total number of users being served under FOO.

Now we take the performance of FOO as a benchmark for analyzing MTO. Because in each time-slot, the candidate user set of FOO is a subset of that for MTO, we know that the total number of users being served under MTO is no less than FOO. Hence,

$$\lim_{C\to\infty} \frac{\bar{Z}_{\mathrm{MTO}}}{\lambda} \geq \lim_{C\to\infty} \frac{\bar{Z}_{\mathrm{FOO}}}{\lambda} = \sum_{k=1}^K \alpha_k[1 - p^*_{0,k}(\zeta_k)],$$

where $\bar{Z}_{\mathrm{MTO}} = \sum_{k=1}^K \bar{Z}_{k,\mathrm{MTO}}$ is the expected total number of users being served under MTO. Hence, we have

$$\lim_{C\to\infty} \frac{\bar{Z}_{\mathrm{MTO}}}{\lambda} = \lim_{C\to\infty} \frac{\sum_{k=1}^K \bar{Z}_{k,\mathrm{MTO}}}{\lambda}.$$

On the other hand, the expected number of served users from each class is bounded by the ON probability, i.e.,

$$\lim_{C\to\infty} \frac{\bar{Z}_{k,\mathrm{MTO}}}{\lambda} \leq \alpha_k[1 - p^*_{0,k}(\zeta_k)].$$

Combining with the bound of the expected total served users, we have

$$\lim_{C\to\infty} \frac{\bar{Z}_{k,\mathrm{MTO}}}{\lambda} = \alpha_k[1 - p^*_{0,k}(\zeta_k)].$$

Equation (7) then follows.

REFERENCES

[1] M. Ra, J. Paek, A. Sharma, R. Govindan, M. Krieger, and M. Neely., "Energy-delay tradeoffs in smartphone applications," in *Proc. ACM MobiSys'10*, San Francisco, CA, June 2010, pp. 255 – 270.

[2] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Taming user-generated content in mobile networks via drop zones," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2840 – 2848.

[3] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, no. 4, pp. 451 – 474, Mar. 2003.

[4] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high dara rate personal communication wireless system," in *Proc. IEEE VTC*, 2000.

[5] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communication Magazine*, vol. 39, no. 2, pp. 150 – 154, Feb. 2001.

[6] A. K.-L. Mok, "Fundamental design problems of distributed systems for the hard-real-time environment," Ph.D. dissertation, MIT, May. 1983.

[7] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *ACM/Baltzer Wireless Networks*, vol. 8, no. 1, pp. 13 – 26, Jan 2002.

[8] I.-H. Hou and R. Singh, "Capacity and scheduling of access points for multiple live video streams," in *Proc. ACM MobiHoc, 2013*, to appear.

[9] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 81, pp. 1452 – 1463, 2006.

[10] U. Ayesta, M. Erausquin, M. Jonckheere, and I. Verloop, "Stability and asymptotic optimality of opportunistic schedulers in wireless systems," in *Proc. the 5th ICST VALUETOOLS*, 2011.

[11] B. Sadiq and G. de Veciana, "Throughput optimality of delay-driven MaxWeight scheduler for a wireless system with flow dynamics," in *Proc.47th Annual Allerton Conference on Communication, Control, and Computing*, Sept. - Oct. 2009, pp. 1097 – 1102.

[12] M. J. Neely, "Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks," *IEEE/ACM Trans. Networking*, vol. 16, no. 5, pp. 1188 – 1199, Oct. 2008.

[13] C. Courcoubetis and R. Weber, "Buffer overflow asymptotics for a buffer handling many traffic sources," *Journal of Applied Probability*, pp. 886 – 903, 1996.

[14] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, "TUBE: Time-dependent pricing for mobile data," in *Proc. ACM SIGCOMM'12*, Helsinki, Finland, Aug. 2012.

[15] 3GPP TR 25.814 V7.1.0, "Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA)," Sept. 2006.

[16] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 636 – 647, Jun. 2005.

[17] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999, vol. 7.

[18] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Trans. Wireless Communications*, vol. 2, no. 4, pp. 630–643, 2003.

[19] R. Irmer (Editor in charge), "NGMN radio access performance evaluation methodology," MGMN White Paper, Jan. 2008.

[20] E. Hyytiä and J. Virtamo, "Random waypoint mobility model in cellular networks," *Journal of Wireless Networks*, vol. 13, no. 2, pp. 177 – 188, Apr. 2007.