NetSchd: The Virtue of Channel and Load Awareness in Alleviating Cellular Congestion

Huasen Wu*†, Xiaojun Lin‡, Xin Liu†§, Kun Tan†, and Yongguang Zhang†

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

†Microsoft Research Asia, Beijing 100080, China

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

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

Abstract

Cellular networks, while being provisioned according to the peak demand, usually experience large fluctuations in both traffic load and channel conditions. Embracing these two dimensions of dynamics allows us to exploit the delay-tolerance of data traffic to alleviate network congestion, and thus reduce the peak. Existing approaches in the literature have considered how to schedule delay-tolerant traffic based on either load-variations or channel-variations alone. However, load-awareness and channel-awareness have never been studied jointly, partly due to the high complexity of the resulting network-wide scheduling problem. Further, the relative performance of the different approaches needs to be compared in a unified framework. Our first contribution in this paper is to develop an optimal solution to the problem of jointly considering load-and channel-awareness to reduce peak demand subject to deadline constraints. We also present a distributed algorithm that is practical to implement. Second, we use trace-driven simulations to carefully compare the performance of the load-only, channel-only, and joint approaches. Our simulation studies reveal the critical role of channel-awareness in wireless systems and the virtue of the joint approach, especially in multi-cell scenarios. To the best of our knowledge, this paper is the first work in the literature that studies load- and channel-awareness in a unified and rigorous manner.

I. Introduction

A grand challenge facing today's mobile service providers is to meet the exponentially increasing demand for mobile broadband services. This problem is particularly severe at the so-called "peak", where the network is highly loaded at specific times and locations. Currently, wireless providers invest heavily in new spectrum and infrastructure to accommodate the *peak* demand, but such efforts are costly and inefficient: since the network traffic at non-peak times is orders of magnitude lower than peaks, provisioning network capacity for peak demand will lead to poor utilization of network resources.

An alternative approach is to exploit the delay tolerance of mobile applications to improve the network utilization. Prior work has identified a class of applications that can tolerant some delay, ranging from a few minutes to hours [1–4]. For example, the analysis in [3] shows that more than 55% of multimedia contents in cellular networks are uploaded more than one day later after their creation time. More recently, the survey conducted in the TUBE project [1] indicates that users are actually willing to delay their data transmissions if appropriate incentives are provided, *i.e.* a discounted price. Motivated by these studies, in this paper we are interested in finding the best scheduling of delay-tolerant traffic to minimize the network congestion in wireless networks.

Exploiting delay-tolerance can alleviate network congestion and reduce peak capacity requirement in two directions: *load-awareness* and *channel-awareness*. First, by moving delay-tolerant traffic to the time and location (e.g., to a Wi-Fi hotspot or a different basestation (BS)) where the network is less loaded, i.e., *being load-aware,* the network can alleviate congestion and carry more traffic overall. This idea is akin to "peak-shedding" [1], where some of the "peak" traffic is moved to the "valleys" when the demand is low. Second, mobile Internet access is highly opportunistic in nature. Due to user mobility and wireless network dynamics, both the network connectivity and signal strength can vary significantly over time. Thus, by opportunistically scheduling traffic to a later period (or location) when the channel condition is more favorable, i.e., *being channel-aware,* the same amount of data transmission would consume less spectrum resources, which again alleviates network congestion and reduces peak capacity requirement. This second direction can be viewed as a form of opportunistic scheduling [5], albeit in a much larger time scale.

Previously, each aforementioned direction has been explorered separately. For example, the TUBE project [1] studies how to schedule data traffic to less-congested periods based

on user-specific deadlines and network incentives. However, TUBE does not consider users' time-varying wireless channels – hence we classify it as a "*load-only*" approach. In the other direction, a number of channel-aware scheduling schemes have been proposed at the mobile device to improve spectrum efficiency [2, 4, 6]. While this line of work takes advantage of the opportunistic nature of wireless networks, it has been limited to optimizing in a single mobile device. As a result, these schemes are oblivious to network congestion and hence we refer to them as "*channel-only*" approaches.

Clearly, existing cellular networks provide opportunities for both load-aware and channel-aware approaches. Hence, several questions could be raised. First, conceivably we can combine both load-awareness and channel-awareness. How can we design such a policy that exploits both load-variations and channel-variations in both single- and multi-cell settings? Second, how do we compare the performance and complexity of the three approaches? Under what scenarios will they perform well? The answers to these questions will be very important from practitioners' point of view in deciding which approach to adopt for future cellular networks, if one ever wants to exploit the delay-tolerance of the traffic.

In this paper, we present a unified analytical framework to study various scheduling policies for delay-tolerant traffics (i.e., load-only, channel-only, and joint approaches). We consider the scenario of a cellular network with multiple BSs and Wi-Fi hotspots. Each data transfer request has a pre-specified deadline, which is directly tied to the users' overall experience. The network's objective is to schedule such data transfers intelligently to alleviate the network congestion and reduce peak demand, subject to the deadline constraints of the data transfers. We define the network congestion cost as the sum of (strictly-)convex functions of the load at each BS/WiFi-hotspot and at each time. With the strict-convexity, the cost function naturally penalizes high peak demand and thus a cost-minimizing solution will tend to smooth out the traffic load across time and location.

Based on our framework, we further study how to design optimal control policies. Naturally, the design of optimal policies for the joint approach is the most difficult, because we have to jointly consider "peak-shedding" and "opportunistic scheduling". First, compared to the channel-only approach that only considers one mobile device [2, 4], here the size of the problem is very large, as a typical network may have hundreds of thousands of requests and a large number of BSs and Wi-Fi hotspots. In addition, if a central entity needs to know all requests and channel evolution statistics for each individual user, it raises concerns on both signaling overhead and privacy. Second, compared to the load-only approach [1], here the

channel uncertainty leads to significant difficulty in determining the amount of load that can be moved under a given policy. In our setting, the amount of traffic that users are willing to delay depends on their channels and the opportunistic scheduling algorithm, and thus the functional form is hard to obtain.

We make following contributions in the paper. First, we develop a distributed solution, referred to as NetSchd, for the joint approach. NetSchd uses a duality approach that addresses both the complexity issue and the signaling/privacy issues discussed earlier. Under NetSchd, the network does not need to know the statistics of all requests before hand, but updates a set of congestion signals based on the aggregated network load. At the same time, each user executes an individual decision policy based on the congestion signal and its own channel statistics. Second, somewhat surprisingly, we show the optimality of the dual approach. This is a non-trivial result given that our problem formulation accounts for a general set of policies that are complex, and it is not obvious that the corresponding objectives and constraints are convex with respect to these policies.

Finally, we have performed extensive trace-driven simulations to evaluate the performance gains of all three approaches. Our simulation results reveal following interesting insights: First, we find that the channel-only approach outperforms the load-only approach in our settings. Note that the load-only approach still requires the network to provide congestion signals, while the channel-only approach can be implemented entirely on the mobile device (without any network support). Thus, this suggests that channel-awareness is more effective and practical than load-awareness in wireless systems. Second, we find that in a single-cell, the joint approach, i.e., NetSchd, only leads to marginal performance gains compared to the channel-only approach. In other words, given that the channel-only approach defers transmission to time-instants with good channels, there is already a "spreading" effect across time that sheds the peak load. Thus, the additional room for NetSchd to further reduce the peak becomes smaller. In contrast, we find that in multi-cell scenarios, the potential gain for NetSchd to outperform the channel-only approach can be higher. This is because in a channel-only solution, it is possible that a congestion-oblivious mobile device may defer transfer until it moves to a BS with better signal quality. However, if this BS has a heavier load, such a channel-only approach will likely cause even higher peak at this BS.

In summary, this work makes both theoretical and practical contributions. Theoretically, the joint approach provides an optimal benchmark for comparition. Practically, we propose a simple distributed algorithm for optimal joint scheduling that is implementable in real

systems. Further, our comparative evaluations provide the cellular operators with operation guidelines to decide their most appropriate approaches among load-only, channel-only, and joint schemes that balance the gain and complexity.

II. Related Work

Load-awareness. TUBE is a theoretical and experimental study that leverages time-dependent pricing to alleviate network congestion [1, 7]. Its pilot trial conducted at Princeton with 50 AT&T data users demonstrates the feasibility of using time-dependent pricing to alleviate network congestion. TUBE leverages network load fluctuation while our work not only considers network load fluctuation, but also user channel variation.

Channel-awareness. Channel-aware scheduling has been extensively studied to improve mobile battery performance and to reduce cellular network load. Both Wiffler [6] and Bartendr [4] consider the setting of vehicular systems to offload 3G data traffic to either WiFi networks or to time-instants when signal strength is stronger. In [2], Lyapunov-optimization-based algorithm is developed for the access link selection problem to reduce energy consumption of data transfers. The authors of [8] use the context information to form a WiFi connectivity profile. The authors use mobility model and AP database to yield a WiFi connectivity forecast in [9]. These channel-only solutions leverage WiFi availability and signal variability, but do not consider network load fluctuation.

Opportunistic scheduling has been widely studied and standardized in wireless networks [5, 10, 11], and a large number of scheduling policies, such as Proportional-Fair [10] and MaxWeight [11], have been proposed. Most existing work focuses on *packet-level* scheduling, where the number of users is assumed to be fixed and the performance is defined at the symbol level. In contrast, in this paper we consider a much larger time scale, e.g., the deadline can range from minutes to hours. Thus, we need to consider the impact of user arrivals and departures upon completion. Flow-level scheduling also considers user/flow arrivals and departures [12–14]. However, most studies of flow-level scheduling focus only on throughput-optimality, without considering deadlines [12, 15]. However, maximizing the stability region cannot guarantee the completion time. When users specify an application-dependent deadline, it becomes more important to maximize the throughput subject to the deadline constraint, which is the focus of our work.

Deadline-constrained scheduling. Scheduling with deadline constraints has been investigated in machine-job scheduling literature. When there is no channel variation, simple

policies such as Earliest-Deadline-First (EDF) have been proposed and shown to be optimal in underloaded systems, i.e., systems that are schedulable [16]. However, once there is channel variation, a difficult trade-off arises between serving more urgent users and serving users with better channel conditions. Only limited results are available for the special case with two-state channels and a fixed number of users, where variants of EDF are proposed to deal with this trade-off. In [17], a Feasible-Earlier-Due-Date (FEDD) policy is proposed for two-state channel models, and shown to be optimal for certain restricted arrival processes, e.g., periodic processes. A more recent work [18] proposes a policy called Earliest Positive-Debt Deadline First (EPDF) for scheduling live video streams without knowing channel states before transmission. However, for more general channel models and time-dependent traffic, we are not aware of a tractable methodology to find optimal scheduling policies subject to strict deadline constraints.

III. Problem Statement

For the ease of exposition, we first consider a cellular network with one BS. The proposed approach can be generalized to include multiple BSs and WiFi-hotspots, as discussed in Section IV-E. The problem stated here applies to both the uplink and downlink in cellular networks.

Assume that time is slotted and indexed by $t \in \{0, 1, \ldots\}$. Let N be the number of time-slots in each day. A typical time-slot length ranges from tens of seconds to a few minutes. Because of the large time scale, we assume that a data transfer request will be completed in one time-slot when the request is accepted, as in [1].

Data Traffic. In cellular networks, mobile users show similar *aggregated* behavior over time (e.g, weekdays), as shown in various measurement studies of cellular traffic [19, 20]. For example, in Fig. 1, real-life load traces of cellular BSs show clear patterns over weekdays and over weekends/holidays.

Consider a typical day. A sequence of data transfer requests enter the network with user-specified deadlines. We use the words "user" and "request" interchangeably and a human user may have multiple requests in a day. The requests depart upon completion or deadline violation. Let $\mathcal{I}$ be the index set of all users entering the system. For each user $i \in \mathcal{I}$, the request is represented by a triple (a_i, b_i, D_i), where a_i, b_i, and D_i denote the arrival time, the file size (in bits), and the deadline, respectively. All variables a_i, b_i and D_i are i.i.d. across i and are taken accordingly to some probability distributions that reflect the typical traffic

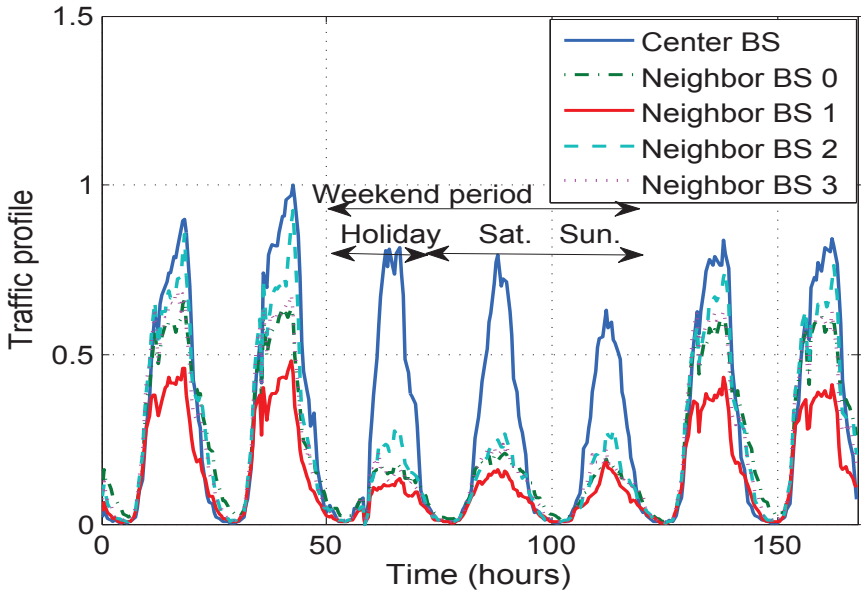


Fig. 1. Normalized cellular load from an anonymous mobile network operator in an urban area, obtained from http://anrg.usc.edu/www/Downloads/.

pattern of the day (see, e.g., Fig. 1). We assume that the file size b_i is available as soon as user i arrives.

Each request i is associated with a user- or application-specific deadline D_i, i.e., the maximum delay that a user can tolerate. The deadline ranges from minutes to hours for delay-tolerant traffic [1, 3], while is set to zero for real-time traffic. Such a deadline requirement depends on specific applications and can be set in various ways. For example, it could be a default setting in an application, e.g., syncing emails every half an hour. Or, it can be learned from user preference. Typically, a UI can provide users the control and flexibility of setting the deadline [1]. We can define a deadline constraint either deterministically or probabilistically, as follows

$$\mathbb{P}\{\text{transmission not finished by } a_i + D_i\} \leq \eta_i, \tag{1}$$

where $\eta_i \geq 0$ is the predefined maximum deadline violation probability. A small violation probability is typical in cellular services, although not necessarily explicitly stated. For example, service providers typically target a deadline violation probability of 1% to 2% for a balance between user satisfaction and cost. We note that $\eta_i = 0$ corresponds to the special case of deterministic deadline constraint.

Channel Model and Transmission Cost. Each user experiences time-varying network availability (e.g., WiFi availability) and channel conditions. This is captured by a stochastic

process $R_i(t)$ ($t \in \{0, 1, 2, \ldots\}$), where $R_i(t) \geq 0$ denotes the instantaneous rate per unit spectrum resource (e.g., a time-frequency block in LTE) at which the BS can transmit to user i in time-slot t. We assume that $R_i(t)$ is i.i.d. across users but may be correlated to (a_i, b_i, D_i). Thus, when user i in channel condition $R_i(t)$ is scheduled to transmit a file of size b_i, it consumes $b_i/R_i(t)$ units of spectrum resource. We assume that each user can estimate its current channel condition via measurements of received signal strength and interference levels. Further, although the future channel condition may be random, the user can learn its statistics based on historical measurements, as in [4, 6, 8, 9] and as discussed in Section IV-C.

Remark: By assuming that a file can be transmitted in one time-slot, the model in this paper focuses on exploiting larger time-scale variations, which are typically due to shadowing and/or users moving further/closer to the BS. Nevertheless, our model does not preclude the BS from using packet-level opportunistic scheduling schemes [5, 10] when serving the users within one time-slot. Indeed, we assume that $R_i(t)$ has already captured the effect of fast time-scale fading.

Scheduling Policy and Base Station Load. Let Γ denote a general scheduling policy that decides which users to transmit at a given time slot. We consider the set of all casual policies. Corresponding to each Γ, we let $l_t(\Gamma)$ be the expected aggregate amount of spectrum resource consumed by the users transmitting in time-slot t under policy Γ. We express $l_t(\Gamma)$ as

$$l_t(\Gamma) = \sum_{i \in \mathcal{I}} c_{i,t}(\Gamma), \;\; t = 0, 1, \ldots, N-1, \tag{2}$$

where $c_{i,t}(\Gamma)$ is the expected amount of resource consumed by user i in time-slot t. More specifically,

$$c_{i,t}(\Gamma) = \mathbb{E}_\Gamma(\mathbf{1}_{\{\text{user } i \text{ transfers at } t\}} b_i/R_i(t)), \tag{3}$$

where the expectation is with respect to the distribution of users' random arrival time and channel conditions. Note that the complexity and generality of the problem is abstracted and hidden in $l_t(\Gamma)$.

Objective. From the network point of view, the objective of scheduling is to minimize the total congestion cost in the horizon of N time slots under the deadline violation constraints. Let $f(\cdot)$ be a general (strictly-)convex congestion-cost function. Our objective is then

$$(\mathcal{P}_0) \quad \underset{\Gamma}{\text{minimize}} \quad F = \textstyle\sum_{t=0}^{N-1} f\big(l_t(\Gamma)\big),$$

subject to the deadline constraint defined in (1) for all users. Note that the convexity of $f(\cdot)$ penalizes peaks and thus favors load that is smoothed over time, which is desirable for network operators. In our numerical results, we use the following function $f(l) = \left(l/\bar{C}\right)^{\nu}$, where $\bar{C}$ is a positive constant and $\nu > 1$ is a factor for controlling the penalty. For a sufficiently large ν, e.g., $\nu = 8$ in our simulations, we can penalize the situation when the load is above $\bar{C}$ during network operation. It is worth noting that even though $f(\cdot)$ is a convex function, the general optimization problem *may not be convex with respect to policies* because the complex coupling of resource consumptions of users and their channel characteristics.

Challenges. First, the size of the problem is very large, as a typical network may have hundreds of thousands of jobs, multiple BSs and WiFi hotspots, over a time horizon of a day. In addition, deadline constraint is notoriously difficult to solve in general because of the resource coupling over time and among users. Second, the problem formulation assumes knowledge of all jobs and their detailed channel information. In practice, it is not feasible to gather such detailed information in a central entity because of both signaling overhead and privacy concerns. Last, the set of all possible policies Γ is very large. Most of these policies are complex to analyze because the expression of $c_{i,t}(\Gamma)$ depends not only on the policy Γ and the current delay $t - a_i$, but also on the evolution of channel process $R_i(t)$.

IV. Algorithm Design

To solve the highly complex problem $\mathcal{P}_0$, we resort to a dual decomposition approach, which allows us to decouple $\mathcal{P}_0$ into a network-side problem and multiple user-side problems. This decomposition approach leads to a distributed algorithm, which addresses both the complexity and privacy issues. This distributed algorithm is our first contribution. Note that dual decomposition does not in general guarantee optimality. Our second contribution is to show the somewhat surprising result on optimality. This is a non-trivial result because $\mathcal{P}_0$ consider all causal policies which are general and complex. In this section, we first lay out the theoretical results of the dual decomposition approach and its optimality. We then present the distributed implementation and discuss the operations on the mobile users and at the BS.

A. The Duality Approach

To use dual decomposition, we first introduce auxiliary variables $h_t \geq 0$ ($t = 0, 1, \ldots, N-1$), $\boldsymbol{h} = \{h_t, t = 0, 1, \ldots, N-1\}$, and rewrite problem $\mathcal{P}_0$ as

$$\begin{aligned} \underset{\Gamma, \boldsymbol{h}}{\text{minimize}} \quad & F = \sum_{t=0}^{N-1} f(h_t) \\ \text{subject to} \quad & l_t(\Gamma) \leq h_t, \quad \text{for all } t, \end{aligned} \tag{4}$$

subject to the deadline constraint defined in (1) for all users. Let $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_{N-1}]$ be the Lagrange multiplier vector corresponding to the constraint in Eq. (4). It will be clear that $\boldsymbol{\beta}$ serves as the congestion signal of the BS over time (in a day). Given $\boldsymbol{\beta}$, we formulate and decompose the Lagrangian as follows:

$$\begin{aligned} \mathcal{L}(\Gamma, \boldsymbol{h}, \boldsymbol{\beta}) &= \sum_{t=0}^{N-1} f(h_t) - \sum_{t=0}^{N-1} \beta_t \big[h_t - l_t(\Gamma)\big] \\ &= \sum_{t=0}^{N-1} [f(h_t) - \beta_t h_t] + \sum_{i \in \mathcal{I}} \sum_{t=0}^{N-1} \beta_t c_{i,t}(\Gamma). \end{aligned} \tag{5}$$

Let the objective function of the dual problem be $g(\boldsymbol{\beta})$, i.e.,

$$g(\boldsymbol{\beta}) = \inf_{\Gamma, \boldsymbol{h}} \mathcal{L}(\Gamma, \boldsymbol{h}, \boldsymbol{\beta}). \tag{6}$$

Because of the linearity of expectation, we can use a distributed policy Γ_i to minimize the expected consumed resource of user i such that the latter term in (5) is minimized. Therefore, for given $\boldsymbol{\beta}$, the dual objective function can be obtained by solving the following subproblems:

$$\begin{aligned} (\mathcal{SP}_0) \quad & \text{minimize}_{\boldsymbol{h} \geq \mathbf{0}} \ \sum_{t=0}^{N-1} [f(h_t) - \beta_t h_t], \\ (\mathcal{SP}_i) \quad & \text{minimize}_{\Gamma_i} \ \sum_{t=0}^{N-1} \beta_t c_{i,t}(\Gamma_i), \ i \in \mathcal{I}. \end{aligned}$$

The master dual problem is

$$\begin{aligned} (\mathcal{D}_0) \quad & \text{maximize}_{\boldsymbol{\beta}} \quad g(\boldsymbol{\beta}) \\ & \text{subject to} \quad \boldsymbol{\beta} \geq \mathbf{0}. \end{aligned}$$

Subproblem $(\mathcal{SP}_0)$ can be easily solved by convex optimization algorithms [21], because $f(\cdot)$ is convex. For subproblem $(\mathcal{SP}_i)$, we can view it as a constrained sequential decision problem and obtain the optimal policy Γ_i by backward induction [22] (see Section IV-C). Upon solving $(\mathcal{SP}_i)$, each user makes transmission decision in each time-slot according

to policy Γ_i. The BS then evaluates the aggregated load in each time-slot and updates the congestion signal $\boldsymbol{\beta}$ using the (sub-)gradient approach (see Sections IV-B and IV-D). This iteration process will finally find the optimal $\boldsymbol{\beta}$ for the dual problem $\mathcal{D}_0$.

Optimality of dual approach. For a general optimization problem, dual decomposition only guarantees weak duality, i.e., the dual solution only provides a lower bound to the original problem. Somewhat surprisingly, we show below that the duality gap of the above proposed approach is zero, and hence the algorithms $(\mathcal{SP}_0)$ and $(\mathcal{SP}_i)$ combined provides an optimal solution to $\mathcal{P}_0$. This optimality result is summarized as follows.

Proposition 1 *Given that the cost function $f(\cdot)$ is convex, the proposed dual decomposition algorithm provides an optimal solution to $\mathcal{P}_0$.*

Sketch of Proof: We note that the above optimality result is nontrivial because the policy Γ can be quite general and it is difficult to obtain the close-form representation of the performance under Γ. In particular, it is not obvious whether the problem $\mathcal{P}_0$ is convex or not with respect to the policy Γ. In fact, even in the case with temporally-independent channels, where the optimal solution to $(\mathcal{SP}_i)$ is a threshold policy (see Section IV-C), one can verify that the objective function F is not convex with respect to the thresholds.

We address these difficulties by reformulating $\mathcal{P}_0$ into an alternative form that exhibits a convex structure albeit with a prohibitively large number of variables. The alternative form, named $\mathcal{P}_1$, is discussed in detail in the Appendix of [23]. Roughly speaking, $\mathcal{P}_1$ assigns a decision variable for each user in each possible state (a state is a possible combination of all users' channel conditions, which is prohibitively huge). These decisions are coupled together because of the load aggregation, the deadline constraints, and channel state evolution. On the other hand, because $\mathcal{P}_1$ is convex, its dual, called $\mathcal{D}_1$, has zero duality gap with $\mathcal{P}_1$. We can then further show that the proposed backward induction approach for $(\mathcal{SP}_i)$, when combined with $(\mathcal{SP}_0)$, is optimal for $\mathcal{D}_1$, and thus solves $\mathcal{P}_1$ optimally. Because of the equivalence of $\mathcal{P}_1$ and $\mathcal{P}_0$, our proposed duality approach is optimal for $\mathcal{P}_0$. Details of the proof are available in the Appendix.

B. Distributed Implementation

Based on the optimality result in Proposition 1, we propose the following distributed implementation, summarized in Algorithm 1. In this distributed algorithm, the BS decides

its congestion signal vector in an iterative fashion and each user individually decides its transmission schedule based on the congestion signal and its channel information.

Algorithm 1 Distributed solution to problem $\mathcal{P}_0$.

Init:

set $k = 0$ and $\beta_t^{(0)} = 1$ for all $t = 0, 1, \ldots, N-1$.

Iteration: (day k)

1) at time $t = 0$, $\beta_t^{(k)}$ $(t = 0, 1, \ldots, N-1)$ is announced to all users;

2) Each user $i \in \mathcal{I}$ solves subproblem $\mathcal{SP}_i$ as they arrive;

3) For $t = 0 \rightarrow N-1$,

Each makes decision based on its channel states;

The BS serves requested users and observes the load level $l_t^{(k)}$;

4) The BS solves subproblem $\mathcal{SP}_0$ and updates $\beta_t^{(k)}$ using (7) (see below);

5) Set $k \leftarrow k+1$ and go to step 1).

In Algorithm 1, we follow a (sub-)gradient method to solve the dual problem $\mathcal{D}_0$:

$$\beta_t^{(k+1)} = \left[\beta_t^{(k)} + \alpha^{(k)}\left(l_t^{(k)} - h_t^{(k)}\right)\right]^+, \ \forall t \tag{7}$$

where k is the iteration index, $\alpha^{(k)}$ is the step-size, and $[\cdot]^+$ denotes the projection to nonegative numbers.

We note that this framework could be applied in both an online and an offline fashion. In the offline fashion, the BS would use all users' channel information and terminate the algorithm if certain stopping criterion is satisfied (e.g., when the duality gap is smaller than a predetermined threshold). Because the offline solution would require all statistical information to be available, it may incur high signaling overhead and privacy issues.

On the other hand, we can choose to implement an online solution, where each iteration represents one day (or weekday) assuming traffic statistics does not change significantly across days. In this case, the BS only needs to update the congestion signal $\beta_t^{(k)}$ based on the observed traffic load. In other words, the BS does not need to know the detailed information of users, and thus resolves the concerns on signaling overhead and user privacy. Note that in this case, the value of $\beta_t^{(k)}$ in (7) should be replaced by its random realization in the k-th iteration. This modified version of (7) then has the flavor of stochastic approximation algorithms [5, 24]. When the step-size α_k is small, we would expect the modified version to

also converge to a small neighborhood of the optimal solution. In the rest of the paper, we focus on the online approach. Next, we discuss the operations on the mobile side and on the network side, respectively.

C. Mobile-side Operation

On the mobile-side, each user operates independently with three components: 1) maintaining a detailed record of channel information, 2) deciding an optimal policy based on the congestion signal and user information, and 3) executing the policy based on the instantaneous channel condition.

Channel Profile. To make better opportunistic transmission decisions, one needs a detailed statistical channel profile. A mobile user can collect and build this profile using time-stamped channel condition information $R_i(t)$ and other contextual information. The rational is that individual human mobility is repetitive, and therefore historical network profile of each user can serve as a stochastic model for its future network conditions. There is a significant amount of research in predicting user mobility and network conditions, e.g., in [4, 6, 8, 9]. For example, Bartendr builds a signal strength profile along popular paths of a user and uses such information to predict channel conditions [6]. We leverage the existing results in this area.

For signaling overhead and privacy reasons, it is expected that the channel profile information stays only on the mobile device, as in NetSchd.

Policy Generation. For a given $\boldsymbol{\beta}$ (i.e., the congestion signal), the subproblem $\mathcal{SP}_i$ under the deadline constraint in Eq. (1) turns out to be a constrained sequential decision problem [22]. In particular, one can introduce a cost for a deadline violation. The mobile minimizes $\mathcal{SP}_i$ plus the deadline violation cost by using backward induction, where the user makes decision by comparing the transmission cost in current time-slot and the cost-to-go. We discuss the deterministic deadline constraint case as follows and refer the readers to [22] for the probabilistic deadline constraint case.

In the deterministic deadline constraint case when $\eta_i = 0$, all data should be transmitted before expiration. Therefore, for user i arriving at a_i, it requires that $\Gamma_i(a_i, D_i-1) = \text{Transmit}$. To guarantee a finite transmission cost, we assume that for each user,

$$\mathbb{E}\{b_i/R_i(|a_i + D_i - 1|_N)|\mathcal{E}_{i,D_i-1}\} < +\infty, \quad i \in \mathcal{I}, \tag{8}$$

where $|\cdot|_N$ is the mod-over-N function and $\mathcal{E}_{i,D_i-1}$ represents the event that user i does not transmit before $a_i + D_i - 1$. In the case with temporally-Markovian channels, using the

principle of optimality and taking the multipliers $\boldsymbol{\beta}$ into account, we can obtain the optimal decision

$$\begin{aligned}&\Gamma_i(a_i,w)\\&=\begin{cases}\text{Transmit}, & \text{if } \frac{\beta_{|a_i+w|_N}}{R_i(|a_i+w|_N)}\leq \mathbb{E}[V^*_{i,w+1}|R_i(|a_i+w|_N)]\\ \text{Wait}, & \text{otherwise},\end{cases}\end{aligned} \tag{9}$$

where $\mathbb{E}[V^*_{i,w+1}|R_i(|a_i+w|_N)]$ is the expected future cost, which can be calculated by backward induction:

$$\begin{aligned}&\mathbb{E}[V^*_{i,w+1}|R_i(|a_i+w|_N)]\\&=\begin{cases}\mathbb{E}\left[\frac{\beta_{|a_i+D_i-1|_N}}{R_i(|a_i+D_i-1|_N)}\middle| R_i(|a_i+D_i-2|_N)\right]\\ \qquad\qquad\qquad\qquad\qquad\qquad \text{for } w=D_i-2;\\ \mathbb{E}\left[\min\left(\frac{\beta_{|a_i+w+1|_N}}{R_i(|a_i+w+1|_N)},V^*_{i,w+2}\right)\middle| R_i(|a_i+w|_N)\right],\\ \qquad\qquad\qquad\qquad\qquad\qquad \text{for } w=D_i-3,D_i-4,\ldots,0.\end{cases}\end{aligned}$$

As a special case of Markovian channels, when the channel process is independent across time-slots, it is easy to verify that the policy becomes a threshold policy, i.e., there exists a threshold T_w for each w, each that the transfer occurs if $R_i(a_i+w)\geq T_w$.

Runtime Execution. The runtime execution of the mobile is simple. It estimates its instantaneous channel condition and then makes decision using the decision table, as shown in Eq. (9).

D. Network-side Operation

The network-side operation consists of two components: serving users and updating congestion signals (for the next day). The update mechanism is described in Eq. (7), and thus we focus on serving users next.

1) Serving without Admission Control: In the ideal case with infinite resource, the BS serves all "ON" users, i.e., the users who have decided to transmit. In each time-slot t during the k-th iteration/day, the BS allocates resource to each "ON" user according to its file size and channel condition, and measures the total consumed resource $l_t^{(k)}$. At the end of the iteration, the BS solves subproblem $\mathcal{SP}_0$ to obtain the optimal $h_t^{(k)}$s, ($t=0,1,\ldots,N-1$). Then, the BS updates the multiplier vector $\boldsymbol{\beta}$ using (7) and broadcasts it for the next iteration/day. This

result can serve as the guideline for network planning (i.e., to quantify the network load). In the following, we also consider modifications needed for network operation.

2) Serving with Admission Control: For an existing network, the available resource of the BS is limited. Hence, certain "ON" users should be temporarily declined if the amount of resource required by all "ON" users exceeds the available capacity. On the other hand, when there are users waiting to transmit and the resource is not fully utilized, the remaining resource can be allocated to the other users.

Dealing with overload. When in a given time-slot, the system is overloaded by "ON" users, the BS serves the requesting users according to a MTB (Maximum-Total-Bits) discipline. In other words, the BS prioritizes the best channel conditions and serves "ON" users in the descending order of their data rate.

Dealing with underload. When there are too few "ON" users in a given time-slot, there may still be resource available after all "ON" users are served. To avoid resource wasting, the network needs a work-conserving enhancement. We introduce a data rate threshold $\bar{R}$. The BS serves the "ON" users first. If there is remaining resource, the BS broadcasts the threshold $\bar{R}$, and all users with channel condition exceeding the threshold send requests to the BS. The BS then serves these users with the remaining resource. Other policies could be adopted as well.

We note that the approaches to deal with overload and underload are important in handling bursty traffic. In the proposed architecture, the multiplier vector $\boldsymbol{\beta}$ is obtained by observing the accessing history and mainly depends on the mean values of the load and channel process. However, because of the burstiness, the system may be overloaded or underloaded. Hence, approaches proposed above are expected to improve the performance under these cases.

E. Multi-cell Networks

For ease of exposition, we have so far focused on the single-cell scenario. Next, we explain how the proposed algorithm can be easily extended to include multiple BSs and WiFi hotspots. First, we note that a cellular BS and a WiFi AP have no conceptual difference in terms of the problem formulation, except that their corresponding congestion cost functions could differ because of the difference in capacity and cost. Second, to extend from one BS to multiple BSs, one can expand the objective (i.e., the total congestion cost) to include all BSs' congestion cost at all time slots. Then, in the duality-based solution, we let each BS have its own congestion signal for each time slot. Therefore, instead of a congestion signal vector,

we introduce a congestion signal matrix $[\beta_{mt}]$, where β_{mt} represents the congestion signal broadcast by BS m in time-slot t. Similar to Section IV-A, we can use $[\beta_{mt}]$ to decompose the primal problem, and rearrange it into the mobile-side and BS-side problems as in $\mathcal{SP}_i$ and $\mathcal{SP}_0$ (except that there are multiple equations similar to $\mathcal{SP}_0$, one for each BS). On the mobile side, each user maintains a profile of the channel condition with respect to each BS over time. For example, at 10am, the user may have 80% chance in cell 1 and 20% in cell 2, and its channel condition may follow a certain distribution depending on the BS that it connects to. Upon receiving the congestion signals from all BSs that it may connect to, the user can then compute the decision table regarding when and which BS it may use to complete the data transfer, while meeting the deadline constraints. The load at each BS is thus determined by mobile users' opportunistic decisions. Finally, at the end of the day, after all mobiles perform their data transfer, each BS updates its congestion signal as in Eq. (7).

In general, different BSs often have different offered load to begin with, as shown in Fig. 1. With a load-aware scheduling policy, the network would prefer a portion of the data transfers to be moved from heavily-loaded BSs to lightly-loaded BSs. In our NetSchd solution, at a given time a heavily-loaded BS will tend to have a larger value for its congestion signal than a lightly-loaded BS. Therefore, in the mobile-side decision, the threshold to transmit for the heavily-loaded BS will be correspondingly higher, which serves the goal of moving an appropriate amount of traffic to other lightly-loaded BSs. In contrast, under channel-only approaches, mobile devices are only aware of the channel condition at each BS, but not its congestion signal. Thus, it is possible that a mobile device delays its traffic until it connects to a BS with a stronger signal, but only finds that the BS has heavy load. In this case, a channel-only solution may not best alleviate network congestion, while NetSchd performs better, as shown next in the numerical results.

F. From NetSchd to Load-only/Channel-only approaches

We mainly focus on the joint approach in the previous sections. Under the proposed framework, we can also investigate the load-only and channel-only approaches, which are discussed as follows and will be evaluated in Section V.

1) Load-only approach: A load-only approach balances the load without considering the channel variations. To compare with the best performance of this type of policies, we consider an optimal *offline* load-only policy that can be viewed as a modification of TUBE [1]. We assume that the knowledge of the traffic (e.g., distributions of arrival time and deadline) is

available by the BS, and the data can be transmitted in any time-slot before the deadline. Then, the corresponding load-balancing problem can be formulated as a convex optimization problem and solved by standard algorithms [21]. We note that the TUBE work focuses on single-cell systems and only *temporal* load-variations are studied in [1]. If one was to also consider *spacial* load-variations, the scheduling problem would be similar to our multi-cell scenarios. However, in that case at least the user-connectivity profile across multiple cells must be taken into account. In other words, the load cannot be considered independently from the connectivity/channel profiles. Due to this reason, similar to [1], we will not study the load-only approach in the multi-cell setting in Section V.

2) Channel-only approach: When the congestion signals are identical across all time-slots and all BSs, NetSchd degenerates to a channel-only approach. We consider the optimal channel-only policy, where each user applies an individual decision policy to minimize the expectation of the consumed resource based on its own channel condition profile under the deadline constraint. The Bartendr policy proposed in [4] can be viewed as one of the channel-only policies, while a fixed threshold is used for any waiting time. The performance of Bartendr is slightly worse than that of the optimal channel-only policy and will not be evaluated in Section V for more concise presentation.

V. Evaluation

We evaluate the performance of load-only, channel-only, and NetSchd approaches via trace-driven simulations. As a baseline, we also consider *ImTrans*, where all users immediately transfer the data when the requests arrive.

A. Simulation Setup

We use a slot length of 10 minutes and each day is divided into 144 time-slots. We consider both a single-cell scenario and a two-cell scenario, except the load-only policy will only be evaluated in the single-cell scenario as discussed in Section IV-F.

1) Traffic Arrival Pattern: We assume a time-dependent Poisson arrival process, i.e., the total number of requests arriving in time-slot t is a Poisson random variable with mean value λ_t ($t = 0, 1, \ldots, N-1$). For the single-cell network, the mean arrival rates are set based on the weekday traffic profile of the center BS shown in Fig. 1. For the multi-cell network, we use the weekday traffic profile of the center BS and the neighbor BS 1 (again in Fig. 1). To capture the delay-tolerance of traffic, we apply the waiting function proposed in [1], and

use the patience indices for the different traffic classes estimated from the U.S. survey in [1]. Specifically, for the delay-tolerant traffic ("Time-Dependent Pricing" traffic in [1]), the probability that user i wants to wait D_i slots is proportional to $\frac{1}{(D_i+1)^\rho}$, where the patience index ρ is 2.0 for video traffic and 0.6 for others. In addition, the usage distribution of the different traffic classes is taken from recent estimates [25], where the proportion of video traffic is about 65%.

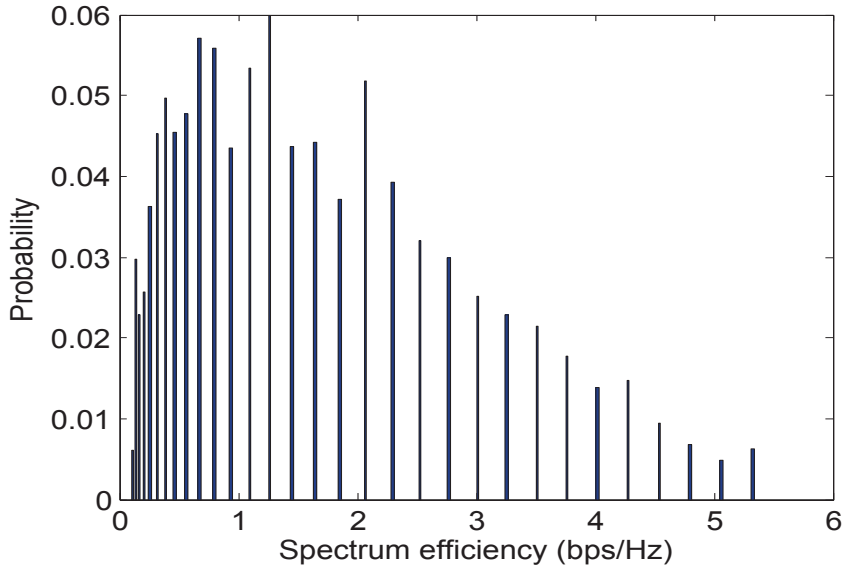


Fig. 2. Distribution of spectrum efficiency.

2) Channel Profile: We collected a set of Received Signal Strength Indication (RSSI) values from a group of anonymous mobile users to best emulate the spectrum efficiency in cellular networks. We assume that the interference strength is a constant and thus the RSSI value represents the SINR, which determines the spectrum efficiency. We follow the LTE-Advanced standards [26], and map the measured RSSI to the proper modes of Modulation and Coding Scheme (MCS). We use the 5-bit CQI and the distribution of the corresponding spectrum efficiency is shown in Fig. 2. To capture time-varying and location-dependent channel conditions, we use a Markov model where Markovian transitions between adjacent channel states (RSSI values) are assumed in each time-slot [27]. We assume that all users use the same channel model. One limitation of the model is that the parameters (e.g., transition probabilities) do not change over time while real human users may have more time-dependent behavior (e.g., 2am at home vs. 2pm at work). We hope to further collect real-life channel profile traces for a more realistic evaluation of real networks in the future.

3) Performance Metrics: We consider two performance metrics: network load and violation probability. We first examine the evolution of the *network load* level, i.e., the total amount of resource required to serve all user requests in each time-slot. Ideally, we prefer the network load level to be lowered and smoothed out. Second, we study the *violation probability* under a given capacity (i.e., the amount of available resource at each BS). Note that we do not impose a hard constraint on the resource available at each BS in the problem formulation. However, in practice the amount of available resource at each BS is finite. If the load is higher than the available resource, some jobs cannot be served. The BS deals with the overload issue as discussed in Section IV-D2. We average across time the fraction of users that are rejected due to resource constraint, and refer to it as the *violation probability*. This probability provides a lower bound on the rejected probability in practice since the rejected users would attempt to transmit in the following time-slots before expiration.

B. Network Load

Fig. 3 shows the network load in one day obtained by different approaches. The three subfigures represent different settings. Fig. 3(a) and Fig. 3(b) are for single-cell systems with 50% and 75% of load being delay-tolerant, respectively. In contrast, Fig. 3(c) is for a multi-cell system with 50% of load being delay-tolerant. We can make a number of interesting observations from Figs. 3(a) and 3(b). Specifically, from Fig. 3(a), we can see that by moving the delay-tolerant traffic into "valleys", the peak load obtained by the load-only policy is about 80% of that under ImTrans. On the other hand, using the channel-only policy, the peak is reduced to about 75% of ImTrans. A similar observation can be made from Fig. 3(b), while the peak load reduction is more significant since there is more delay-tolerant traffic. This finding suggests that channel-awareness can be more effective than load-awareness in wireless systems.

Further, although NetSchd leads to even lower peak consumption by considering both load-awareness and channel-awareness, the additional gain compared to the channel-only policy is relatively marginal in the single-cell setting in Fig. 3(a) and Fig. 3(b) (about 8% reduction in both figures). We note that, under the channel-only policy, users defer their transmissions when waiting for good channels. Therefore, a "peak-shedding" effect also occurs under the channel-only approach. Since the traffic fluctuation is not large, the room for NetSchd to further move traffic is relatively small. However, the multi-cell simulation in Fig. 3(c) illustrates different behaviors. By moving the delay-tolerant traffic to the neighbor BS (i.e.,

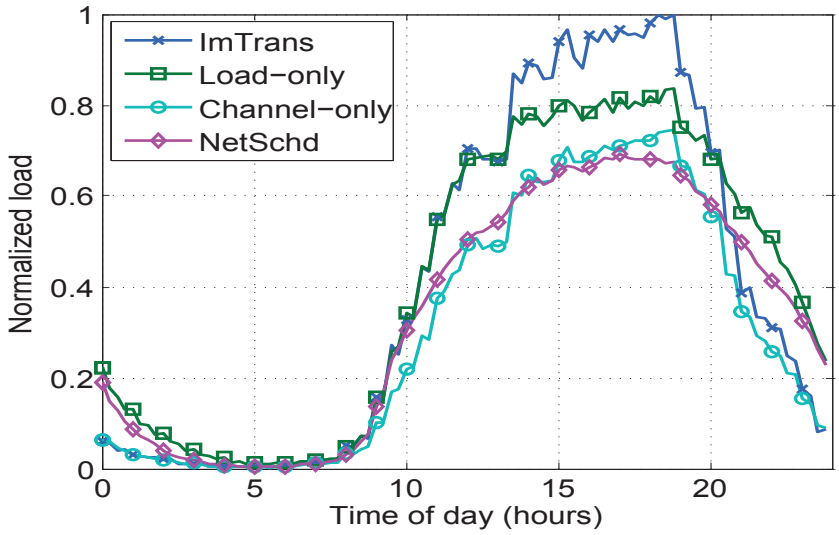
BS 2), the peak of network load (corresponding to the load in BS 1 at about 18:00) is reduced by about 20% by NetSchd compared to the channel-only policy.

C. Violation Probability

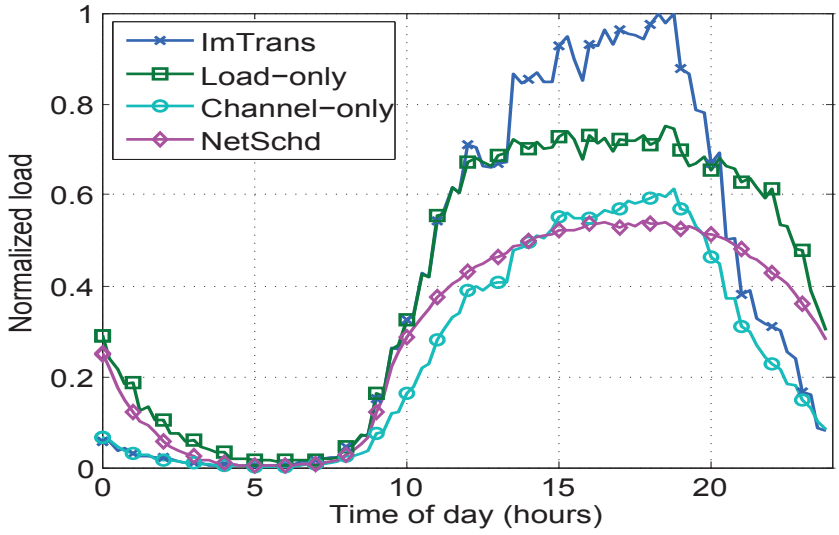
Fig. 4 shows the violation probability under different capacity. The simulations assume 50% of delay-tolerant traffic. In the single-cell scenario, as shown in Fig. 4(a), the load-only and channel-only policies achieve similar violation probability, which is much lower than that under ImTrans. In comparison, with both load- and channel-awareness, NetSchd leads to an slightly bigger reduction. For example, for a 2% of violation probability requirement, NetSchd requires about 315 units of resource, which is 20% reduction compared to ImTrans and about 7% reduction compared to the channel-only policies). The performance of NetSchd is close to the channel-only policy in this single-cell setting due to the same reason discussed in Section V-B.

The relative gain of NetSchd is much higher in Fig. 4(b). Though we consider low mobility (users stay in the same cell with probability 0.9 in consecutive time slots), NetSchd is able to move more delay-tolerant traffic to the neighboring cell that has a much lower load. As a result, the load-awareness brings more significant improvement than that in the single-cell system. For example, the capacity required for a 2% violation probability is 315 units for channel-only, and is 260 units for NetSchd (about 17.5% reduction). These results indicate that, when large traffic fluctuation occurs across BSs, scheduling with both load- and channel-awareness alleviates network congestion and improve the resource utilization.

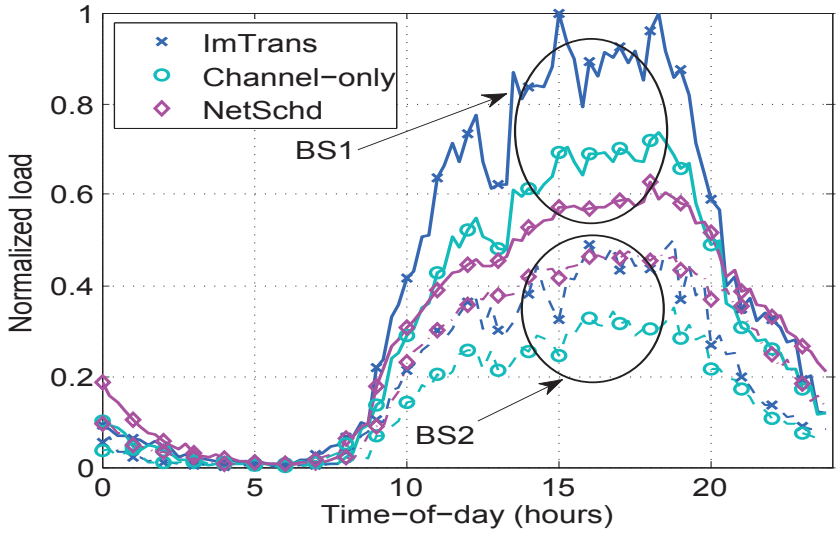
In summary, we observe that channel awareness is rather important in wireless networks and load balancing among multiple cells provides additional gain. Further, we note that the channel-only approach is easier to implement than the load-only/NetSchd approach in general. The load-only/NetSchd approaches require the network to provide time-dependent congestion signals/prices. In contrast, the channel-only approach can be implemented entirely on mobile devices, without any network support. Based on the optimal benchmark provided by NetSchd, operators could choose the most suitable options for their specific network environment. Specifically, when traffic fluctuation is marginal across times and locations, the operator may choose channel-only for its simplicity. When there exist large load differences among BSs, the operator may choose the joint approach, i.e., NetSchd, for further performance gains.



(a)

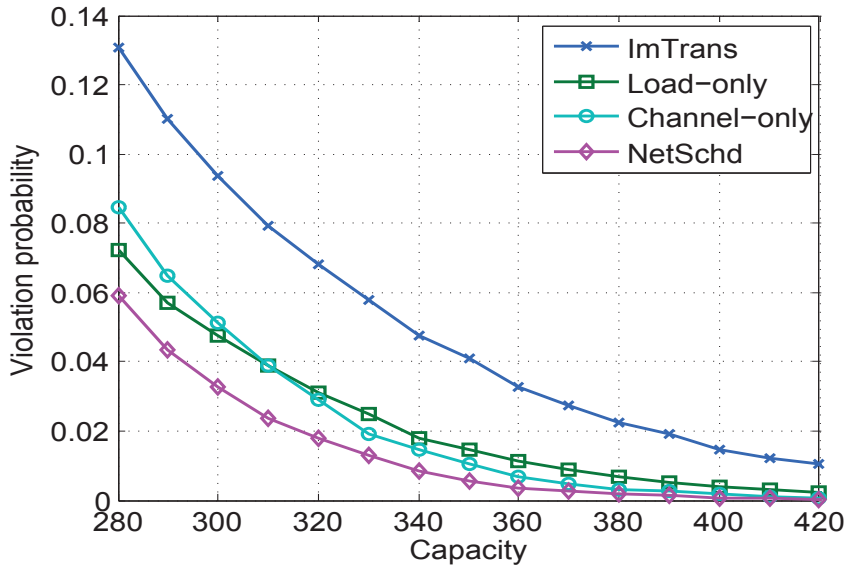


(b)

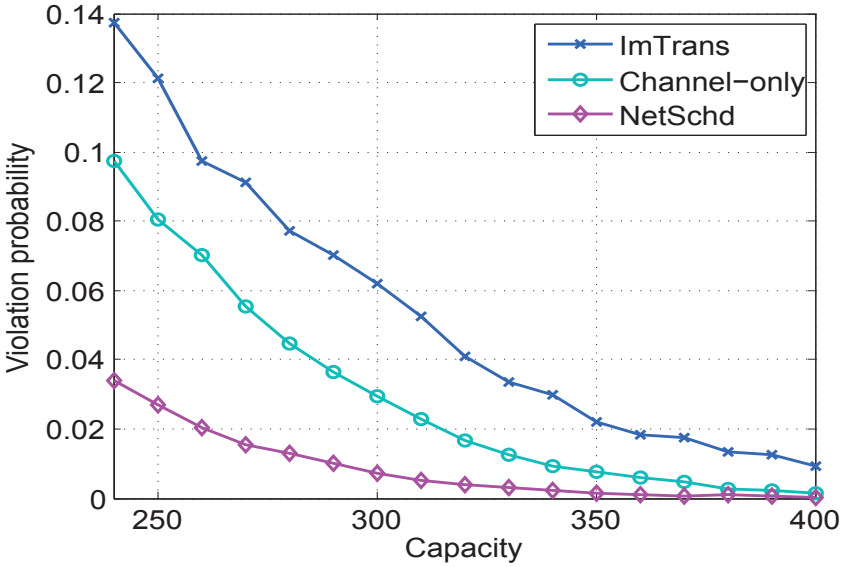


(c)

Fig. 3. Load level under different approaches and settings (a) single-cell with 50% elastic traffic, (b) single-cell with 75% elastic traffic, and (c) multi-cell with 50% elastic traffic.



(a)



(b)

Fig. 4. Violation probability vs capacity (50% of delay-tolerant traffic). (a) single-cell system, (b) multi-cell system.

VI. Conclusions

In this paper, we present a unified framework to study the effectiveness of load- and/or channel-awareness in deadline-constrained scheduling of delay-tolerant traffic for the purpose of alleviating cellular network congestion. Despite the high complexity of the joint scheduling problem with explicit deadline constraints, we develop an optimal solution, called NetSchd, and propose its distributed implementation.

The results in the paper are of both theoretical and practical values. Theoretically, the joint

approach provides an optimal benchmark for comparing with other solutions. Practically, our proposed policy can be implemented in a distributed manner in real systems. Further, our comparative evaluations provide the cellular operators with operation guidelines to decide their most appropriate approaches. Specifically, our numerical results suggest that channel-awareness is rather important in wireless networks. For single-cell systems, channel-only may be preferred due to its simplicity and relatively good performance. For multi-cell systems with load variations, NetSchd can attain significant additional gains. Finally, we note that the relative performance of the three approaches may vary depending on the actual load and mobility patterns. Hence, for future work it will be highly desirable to use real-life traces from users within the same set of multiple cells, including their mobility and load variations, to conduct more comprehensive evaluations.

VII. APPENDIX

To prove the optimality of the proposed dual decomposition approach, we show that problem $\mathcal{P}_0$ can be reformulated to a convex optimization problem $\mathcal{P}_1$, albeit with an exponentially large number of decision variables. In addition, our proposed threshold policy can optimally solve the dual of $\mathcal{P}_1$, named $\mathcal{D}_1$, and thus our scheme is optimal. For simplicity, we assume the file-size is fixed for each user and the expectation in equation (3) is with respect to the random arrival times and channel conditions. The results can be directly extend to the case with random file size.

First, we show that any policy Γ can be represented by a stochastic policy Ψ. Each causal policy Γ makes decision based on the history of the arrival sequence and channel processes. To represent the history, for each user $i \in \mathcal{I}$, we introduce $A_i(t)$ to represent its present status in time-slot t. Namely, if user i arrives in time-slot a_i (we let $a_i = N$ represent the event that user i does not appear), then $A_i(t) = -1$ if $a_i > t$, and $A_i(t) = a_i$ if $a_i \leq t$. Recall that $R_i(t)$ ($i = 0, 1, \ldots, N-1$) is the channel process of user i. Hence, the history of the system up to time-slot t is given by

$$\boldsymbol{S}_t = [\boldsymbol{A}_t \ \boldsymbol{R}_t],$$

where $\boldsymbol{A}_t = [A_1(t), A_2(t), \ldots, A_{|\mathcal{I}|}(t)]^{\mathrm{T}}$ and

$$\boldsymbol{R}_t = \begin{bmatrix} R_1(0) & R_1(1) & \ldots & R_1(t) \\ R_2(0) & R_2(1) & \ldots & R_2(t) \\ \vdots & \vdots & \ddots & \vdots \\ R_{|\mathcal{I}|}(0) & R_{|\mathcal{I}|}(1) & \ldots & R_{|\mathcal{I}|}(t) \end{bmatrix}.$$

Let Ω be the set of possible realizations of arrival sequence and channel processes, i.e., the possible realization of $\boldsymbol{S}_{N-1}$. Then each policy Γ can be represented by a stochastic policy Ψ, which is a $\Omega \mapsto [0,1]^{|\mathcal{I}| \times N}$ mapping: for each $\boldsymbol{s} \in \Omega$,

$$\Psi(\boldsymbol{s}) = \begin{bmatrix} \psi_1(\boldsymbol{s}_0) & \psi_1(\boldsymbol{s}_1) & \ldots & \psi_1(\boldsymbol{s}_{N-1}) \\ \psi_2(\boldsymbol{s}_0) & \psi_2(\boldsymbol{s}_1) & \ldots & \psi_2(\boldsymbol{s}_{N-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{|\mathcal{I}|}(\boldsymbol{s}_0) & \psi_{|\mathcal{I}|}(\boldsymbol{s}_1) & \ldots & \psi_{|\mathcal{I}|}(\boldsymbol{s}_{N-1}) \end{bmatrix},$$

where $\boldsymbol{s}_t$ is the history of arrival sequence and channel processes up to time-slot t for the realization $\boldsymbol{s}$, and $\psi_i(\boldsymbol{s}_t) \in [0,1]$ is the transmission probability of user i in time-slot t. Note that for the user with $a_i + D_i \geq N$, it may transmit in the following day and the transmission

probability is represented by $\psi_i(\boldsymbol{s}_{|a_i+w|_N})$ ($w=0,1,\ldots,D_i-1$), where $|\cdot|_N$ is a mod over N function. For these users, $\boldsymbol{s}_t$ includes the history in the past day. Therefore, the policy Ψ makes decisions based on the arrival sequence and channel conditions until the current slot and is thus causal.

Second, we study the expected resource consumed by user i under $\Psi(\boldsymbol{s})$. For each $\boldsymbol{s}\in\Omega$ where user i arrives in time-slot a_i, we can calculate the probability that user i transmits in slot a_i+w as follows

$$\varphi_i(\boldsymbol{s}_t)=\begin{cases}\psi_i(\boldsymbol{s}_{a_i}), & t=a_i\\ \psi_i(\boldsymbol{s}_{|a_i+w|_N})\sum_{w'=0}^{w-1}[1-\psi_i(\boldsymbol{s}_{|a_i+w'|_N})], & \\ & t=|a_i+w|_N, 0<w\le D_i-1\\ 0, & \text{otherwise.}\end{cases}$$

For given $\boldsymbol{s}$, the expected consumed resource of user i in time-slot t is

$$c'_{i,t}(\boldsymbol{s},\Psi)=\frac{b_i\varphi_i(\boldsymbol{s}_t)}{R_i(t)}.$$

In addition, note that all users should satisfy the deadline constraint. Hence,

$$\sum_{w=0}^{D_i-1}\varphi_i(\boldsymbol{s}_{|a_i+w|_N})\ge 1-\eta_i,\quad \boldsymbol{s}\in\Omega, i\in\mathcal{I}. \tag{10}$$

Moreover, using the relationship between $\varphi_i(\cdot)$ and $\psi_i(\cdot)$, a $\varphi_i(\cdot)$ satisfying (10) can be mapped to a policy Ψ [1].

Consequently, problem $\mathcal{P}_0$ is equivalent to

$$\begin{aligned}(\mathcal{P}_1)\ \underset{\Psi,h'}{\text{minimize}}\quad & F=\sum_{t=0}^{N-1}f\big(h'_t\big),\\ \text{subject to}\quad & \sum_{w=0}^{D_i-1}\varphi_i(\boldsymbol{s}_{|a_i+w|_N})\ge 1-\eta_i,\ \boldsymbol{s}\in\Omega, i\in\mathcal{I},\\ & l'_t(\Psi)\le h'_t,\quad t=0,1,\ldots,N-1,\end{aligned}$$

where

$$l'_t(\Psi)=\sum_{\boldsymbol{s}\in\Omega}\sum_{i\in\mathcal{I}}\pi(\boldsymbol{s})c'_{i,t}(\boldsymbol{s},\Psi). \tag{11}$$

We can verify that $\mathcal{P}_1$ is a convex optimization problem because $f(\cdot)$ is a convex function and all the constraints are linear constraints. However, we do note that it is impractical to

[1] If $\sum_{w'=0}^{w}\varphi_i(\tilde{\boldsymbol{r}}_{0:a_i+w'})=1$ for some $w<D_i-1$, then for $w'>w$, $\psi_i(\boldsymbol{r}_{i,w})$ can be artificially set to be 0, which will not affect the behavior of Ψ.

solve $\mathcal{P}_1$ directly because of its large number of variables. Recall that there are $|\mathcal{I}| \times N$ decision variable for each possible state. Note that there are $N^{|I|}$ possible arrival sequence and we assume the channel state of each user can be quantized to K values. Then there are $N^{|I|}K^{|\mathcal{I}|\times N}$ possible states, and thus $|\mathcal{I}| \times N \times N^{|I|}K^{|\mathcal{I}|\times N}$ decision variables, which is clearly intractable. We note that the formulation can be considered as a linear representation of a centralized Markov Decision Policy, which clearly suffers the curve of dimensionality.

Again, we resort to the dual decomposition approach to study $\mathcal{P}_1$. Similar to the approach in Section IV, we can introduce a dual variable for each time slot, and then rearrange the variables that belongs to each user, resulting the dual problem of $\mathcal{P}_1$, called $\mathcal{D}_1$. Then we have a similar format as in $\mathcal{SP}_0$ and $\mathcal{SP}_i$. We can verify that solving $\mathcal{D}_0$ in fact solves $\mathcal{D}_1$. Because of the convexity of $\mathcal{P}_1$, $\mathcal{D}_1$ has a zero dual-gap, and so does $\mathcal{D}_0$.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpandez, C. Grunewald, K. Jain, and V. N. Padmanabhan, "Bartendr: A practical approach to energy-aware cellular data scheduling," in *Proc. ACM MobiCom' 10*, Chicago, Sept. 2010, pp. 85 – 96.

[5] 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.

[6] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *Proc. of ACM MobiSys'10*, San Francisco, CA, June 2010, pp. 209–222.

[7] TUBE: Time-Dependent Pricing. [Online]. Available: http://scenic.princeton.edu/tube/tdp.html

[8] A. Rahmati and L. Zhong, "Context-for-wireless: context-sensitive energy-efficient wireless data transfer," in *Proceedings of ACM MobiSys'07*, New York, NY, USA, 2007, pp. 165–178.

[9] A. J. Nicholson and B. D. Noble, "Breadcrumbs: forecasting mobile connectivity," in *Proc of ACM MobiCom'08*, New York, NY, USA, 2008, pp. 46–57.

[10] 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.

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

[12] 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.

[13] S. Liu, L. Ying, and R. Srikant, "Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics," *IEEE/ACM Trans. Networking*, vol. 19, no. 4, pp. 1057 – 1070, Aug. 2011.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, "Primary user behavior in cellular networks and implications for dynamic spectrum access," *IEEE Comm. Mag.*, vol. 47, no. 3, pp. 88–95, Mar. 2009.

[20] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Towards dynamic energy-efficient operation of cellular network infrastructure," *IEEE Communications Magazine*, vol. 49, no. 6, 2011.

[21] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[22] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. Wiley, 2005.

[23] H. Wu, X. Lin, X. Liu, K. Tan, and Y. Zhang, "NetSchd: the virtue of channel and load awareness in alleviating cellular congestion," *Technical Report*, 2013. [Online]. Available: https://engineering.purdue.edu/%7elinx/papers.html

[24] H. J. Kushner and G. G. Yin, *Stochastic Approximation and recursive algorithms and applications*. Springer, 1997.

[25] "Cisco visual networking index: Global traffic forcast update, 2012 - 2017," Feb. 2013. [Online]. Available: http://www.cisco.com

[26] 3GPP TS 36.213 V11.0.0, "Physical layer procedures," Sept. 2012.

[27] H. S. Wang and N. Moayeri, "Finite-state markov channel-a useful model for radio communication channels," *IEEE Trans. Vehicular Technology*, vol. 44, no. 1, 1995.