Efficient Broadcast for Timely Updates in Mobile Networks

Yu-Pin Hsu

Abstract—This letter considers a wireless network where an access point (AP) broadcasts timely updates to several mobile users. The timeliness of information owned by a user is characterized by the recently proposed age of information. While frequently broadcasting the timely updates and always using the maximum power can minimize the age of information for all users, that wastes valuable communication resources (i.e., time and energy). For addressing the age-energy trade-off, it is critical to develop an efficient scheduling algorithm identifying broadcast times and allocating the power. Moreover, unpredictable users’ movement would cause rapidly-varying communication channels; in particular, those channels can be non-stationary or adversarial. Our main contribution is to develop an online scheduling algorithm and a channel-agnostic scheduling algorithm for the mobile network with a provable performance guarantee.

I. INTRODUCTION

In recent years, there is an explosive growth of applications for mobile users that rely on timely updates. For example, online route planning applications need timely transportation or traffic information. In those applications, the timeliness of the information is critical. Thus, the age of information was recently proposed in [1] as a metric to measure the timeliness of the information. The goal is to develop a network for minimizing the age of information.

In this letter, we consider a scenario where several mobile users request the same updates (e.g., for transportation information) through an access point (AP). The AP transmits an update to those users simultaneously by broadcast (with a single transmission). To minimize the age of information, the AP would transmit as many latest updates as possible and also transmit them with the highest power level (such that most users can successfully decode the updates). However, as the dramatically increasing demands for mobile services, communication resources (i.e., time and power) at the AP are valuable. To fully exploit the precious resources, this letter aims to develop scheduling algorithms determining when to transmit an update and how to allocate the power for each transmission. The goal is to strike a balance between the age of information (for the users) and the energy consumption (for the AP).

The scheduling design for tackling the age-energy trade-off has drawn much attention, e.g., [2, 6]. Almost all prior works assumed stochastic environments where the environment variables (e.g., communication channels) follow some probability distributions or have some stationary assumptions. However, a mobile user can move at will. Because of the users’ highly unpredictable movement, it is infeasible to assume stationary channels for our problem. Our previous work [6] was the first attempt to investigate the trade-off in an adversarial environment. However, [6] only considered a single user in the ON-OFF channel caused by a fixed power manager. By contrast, this letter considers multiple mobile users in general fading channels (with more than two states). The great uncertainty caused by multiple mobile users poses a major challenge. Moreover, this letter allows adaptive power selections. That additional option can save more energy but further complicate our scheduling design.

The main contribution of this letter is to develop an online scheduling algorithm and a channel-agnostic scheduling algorithm. While the former needs the present channel state only (without future information), the latter further needs no channel state. We show that both algorithms have a universal performance guarantee, which is independent of the number of users and their movement.

II. SYSTEM OVERVIEW

A. Network model

We consider a wireless network consisting of an access point (AP) and N mobile users, where users 1, · · · , N move in the service area of the AP. The users run applications that request the same timely updates (e.g., for transportation information) through the AP. Divide time into slots 1, 2, · · · , T, where T is the time horizon under consideration.

At the beginning of each slot, the AP decides whether to serve the users or not. If the AP decides to do for a slot, then it obtains an update from an information source, allocates the transmission power, and transmits the update by broadcast during that slot. In this letter, we neglect the transmission time between the information source and the AP (through wired networks), while focusing on the bottleneck between the AP and the users (through wireless networks). Let d(t) ∈ {0, 1, · · · , M} be the decision of the AP for slot t, where d(t) = 0 if it decides not to serve and d(t) ∈ {1, · · · , M} is the power level allocated for broadcast in slot t if it decides to do. Suppose that the power consumption increases with the index of the power level.

We use 1i,k(t) ∈ {0, 1} to indicate if user i can receive (i.e., successfully decode) the update in slot t if the AP takes power level k, where 1i,k(t) = 1 if it can but 1i,k(t) = 0 if it cannot. The indicator function 1i,k(t) depends on the channel quality (e.g., caused by signal fading or noises) between the AP and user i in slot t. We suppose that, for each slot, the AP can deliver an update to all users with the maximum power level
Let $s_i(t) = (1_{i,1}(t), \ldots, 1_{i,M}(t))$ be the channel state of user $i$ in slot $t$. There are $M$ potential states for $s_i(t)$, i.e., $(1, \ldots, 1)$, $(0, 1, \ldots, 1)$, $\ldots$, $(0, \ldots, 0, 1)$. Let $s_i = (s_i(1), \ldots, s_i(T))$ be the channel state pattern of user $i$ over slots, representing its movement. Because of unpredictable movement, channel state pattern $s_i$ can be arbitrary without a stationary property, for all $i$. Moreover, $s_i$ and $s_j$ for different $i$ and $j$ can have any correlation, for example, when a group of users moves together. Let $s = (s_1, \ldots, s_N)$ represent the entire channel state pattern of all users.

### B. Age model

We use the age of information \[ \text{[1]} \] to measure the freshness of the information at each user. If a user receives an update in a slot, then its age of information becomes zero\[ \text{[4]} \] at the end of that slot (indicating the latest update); otherwise, the age of information at the end of that slot increases by one from the previous slot. Let $a_i(t)$ be the age of information for user $i$ at the end of slot $t$. Then, given decision $d(t) = k$ in slot $t$, we can describe $a_i(t)$ for user $i$ in slot $t$ by

$$a_i(t) = \begin{cases} 0 & \text{if } 1_{i,k}(t) = 1; \\ a_i(t-1) + 1 & \text{if } 1_{i,k}(t) = 0. \end{cases} \tag{1}$$

We assume the initial age $a_i(0) = 0$ for all $i$.

### C. Problem formulation

A scheduling algorithm $\pi = \{d(1), \ldots, d(T)\}$ specifies decision $d(t)$ for all slots. A scheduling algorithm is called an offline scheduling algorithm if it has the entire channel state patterns $s$ (along with the time horizon $T$) as a prior. By contrast, a scheduling algorithm is called an online scheduling algorithm if for each slot $t$ it has only the present channel state $s_i(t)$ for all users $i$ (by channel estimation techniques). Moreover, a scheduling algorithm is called a channel-agnostic scheduling algorithm if for each slot $t$ it has no knowledge about channel state $s_i(t)$ for any user $i$ or time horizon $T$.

To develop scheduling algorithms striking a balance between the freshness of the information at the users and the energy consumption at the AP, we define an age cost and a transmission cost as follows. Suppose that the age of one unit in a lot incurs an age cost of one unit in that slot. Then, the age cost incurred by the stale information at user $i$ in slot $t$ is $a_i(t)$. We consider the average age cost ($\sum_{t=1}^{T} a_i(t))/N$ over the number of users (for a fair comparison across different numbers $N$ of users). Moreover, suppose that the power level $k$ incurs a transmission cost of $C_k$ units. Suppose that $C_1 \leq \cdots \leq C_M$. Note that $C_k$ does not need to linearly increase with $k$.

\[ \text{[1]} \] From [1], the age of information should become one upon receiving an update. Then, the age of information for user $i$ in slot $t$ under that age model [1] is our $a_i(t)$ in Eq. (1) plus one. That additional age simply causes an additional constant term $T$ to our later optimization problem in Eq. (4), without changing the optimal solution. Thus, it suffices to focus on the simple age model in Eq. (1).

For an entire channel state pattern $s$, we define the total cost $J(s, \pi)$ under scheduling algorithm $\pi$ by the sum of the transmission costs and the average age costs:

$$J(s, \pi) = \sum_{t=1}^{T} \left( C_{d(t)} + \frac{1}{N} \sum_{i=1}^{N} a_i(t) \right), \tag{2}$$

where we set $C_0 = 0$ indicating the zero transmission cost when $d(t) = 0$. Eq. (2) captures the age-energy trade-off. The higher power level $d(t)$ the AP takes (in the first term of Eq. (2)), the more users can receive the latest update, yielding a smaller average age cost (in the second term of Eq. (1)). On the contrary, taking a lower power level increases the age of information for those users who cannot decode the update as a result of weak channels.

This letter will propose an online scheduling algorithm and a channel-agnostic scheduling algorithm. We will analyze the proposed algorithms in terms of the competitiveness against an optimal offline scheduling algorithm. To that end, for an entire channel state pattern $s$, let $OPT(s) = \min_{\pi'} J(s, \pi')$ be the minimum total cost for all possible (offline) scheduling algorithms $\pi'$. If there is a constant $\gamma$ such that $J(s, \pi) \leq \gamma \cdot OPT(s)$ for all possible entire channel state patterns $s$, then the constant $\gamma$ is called the competitive ratio of the scheduling algorithm $\pi$.

### III. SCHEDULING ALGORITHM DESIGN

This letter approaches the scheduling problem leveraging primal-dual techniques [2] for linear programs. To that end, Section III-A constructs a virtual queueing system for describing the age evolution in Eq. (1); particularly, with the assist of the virtual queueing system, Section III-B proposes a linear program whose optimal objective value is a lower bound on the total cost. In addition, Section III-C-E formulates the dual program of the linear program (as a primal program) for analyzing our proposed algorithms (by the duality theory). Then, Section III-C proposes a primal-dual algorithm that produces a feasible solution to the primal program and the dual program in the online fashion. Next, Section III-D proposes a (randomized) online scheduling algorithm by casting the fractional solution produced by the primal-dual algorithm in each slot to a randomized decision for that slot. Finally, Section III-E proposes a (randomized) channel-agnostic scheduling algorithm achieving the same competitive ratio as the online scheduling algorithm does.

#### A. Virtual queueing system

This section constructs a virtual queueing system consisting of queues $1, \ldots, N$ (corresponding to users $1, \ldots, N$) so that the queue size evolution is equivalent to the age evolution in Eq. (1). The virtual queueing system will facilitate a linear program formulation in the next section.

At the beginning of each slot $t$, each queue $i$ has a newly arriving packet $j$. Note that a total of $N$ packet $j$’s arrive at the queueing system in slot $t$ because the system has $N$ queues. In addition, for each slot $t$, if user $i$ can receive the update (in the real-world mobile network), then queue $i$ flushes all its
packets (in the virtual queuing system); otherwise, queue \(i\) idles.

According to the arrival process and the service process of queue \(i\), its queue size at the end of slot \(t\) becomes zero if \(1_{i,d(t)}(t) = 1\), or increases by one if \(1_{i,d(t)}(t) = 0\). The queue size evolution is identical to the age evolution in Eq. (1). Thus, the queue size of queue \(i\) at the end of slot \(t\) exactly expresses \(a_i(t)\).

B. Primal-dual formulation

Leveraging the queuing system constructed in Section III-A this section proposes an integer program for optimally solving the offline scheduling problem. Let \(z_{i,j}(t) \in \{0, 1\}\) indicate if packet \(j\) (arriving in slot \(j\)) stays at queue \(i\) at the end of slot \(t\), where \(z_{i,j}(t) = 1\) if it does but \(z_{i,j}(t) = 0\) otherwise. According to Section III-A, age \(a_i(t)\) in slot \(t\) is the queue size of queue \(i\) at the end of slot \(t\). Thus, age \(a_i(t)\) in Eq. (1) can be expressed by \(a_i(t) = \sum_{j=1}^{t} z_{i,j}(t)\), counting all packets arriving at queue \(i\) by slot \(t\). Moreover, let \(x_k(t) \in \{0, 1\}\) indicate if power level \(k\) is selected in slot \(t\), where \(x_k(t) = 1\) and \(x_{k'}(t) = 0\) for all \(k' \neq k\) if and only if \(d(t) = k\). Then, cost \(C(t)\) in Eq. (2) can be expressed by \(C(t) = \sum_{k=1}^{M} C_k x_k(t)\). Substituting \(C(t)\) and \(a_i(t)\) in Eq. (1) by the new expressions, we can rewrite the total cost \(J(s, \pi)\) by a linear function in terms of \(x_k(t)\) and \(z_{i,j}(t)\):

\[
J(s, \pi) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{t} z_{i,j}(t) = \sum_{t=1}^{T} \left( \sum_{k=1}^{M} C_k x_k(t) + \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{t} z_{i,j}(t) \right). \tag{3}
\]

Then, we propose the following integer program for optimally solving our scheduling problem when the entire channel state pattern \(s\) is known in advance.

\[
\begin{align*}
\min \quad & \sum_{t=1}^{T} \left( \sum_{k=1}^{M} C_k x_k(t) + \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{t} z_{i,j}(t) \right) \quad \tag{4a} \\
\text{s.t.} \quad & \sum_{j=1}^{t} \sum_{\tau=1}^{\tau_{max}} I_{i,k}(\tau) x_k(\tau) \geq 1 \\
& \text{for all } i = 1, \ldots, N, \quad j = 1, \ldots, t, \quad \text{and } t = 1, \ldots, T; \tag{4b} \\
& x_k(t), z_{i,j}(t) \in \{0, 1\} \text{ for all } i, j, k, t. \tag{4c}
\end{align*}
\]

In the integer program, the constraints in Eq. (4b) mean that packet \(j\) arriving at queue \(i\) by slot \(t\) either stays at queue \(i\) in slot \(t\) (i.e., \(z_{i,j}(t) = 1\)) or has been flushed by slot \(t\) (i.e., \(I_{i,k}(\tau) x_k(\tau) = 1\) for some \(\tau = j, \ldots, t\) and some \(k = 1, \ldots, M\)). Note that the optimization problem neglects a redundant constraint \(\sum_{k=1}^{M} x_k(t) \leq 1\) for all \(t\), because the minimization problem automatically constrains that constraint.

By relaxing the integral constraints in Eq. (4c) to real numbers, we have a linear program. Because of the relaxation, the minimum objective value for the linear program is a lower bound on the minimum total cost (obtained by the integer program). Moreover, unlike the integer program, a feasible solution to the linear program can be fractional, which no longer represents an immediate decision for broadcasting an update or allocating the power. Later, Section III-D will cast a fractional solution for \(x_k(t)\) to a probabilistic decision in slot \(t\).

To analyze the solution produced by the proposed algorithms, we leverage the duality theory [7]. Thus, we refer to the linear program as a primal program and formulate its dual program as follows.

\[
\begin{align*}
\max \quad & \sum_{t=1}^{T} \sum_{i=1}^{N} y_{i,j}(t) \quad \tag{5a} \\
\text{s.t.} \quad & \sum_{j=1}^{T} \sum_{\tau=1}^{\tau_{max}} I_{i,k}(\tau) y_{i,j}(\tau) \leq C_k \text{ for all } k, t; \tag{5b} \\
& 0 \leq y_{i,j}(\tau) \leq \frac{1}{N} \text{ for all } i, j, \tau. \tag{5c}
\end{align*}
\]

C. Primal-dual algorithm

This section proposes a primal-dual algorithm in Alg. 1 for obtaining a feasible solution to the primal program and the dual program in the online setting, where for each new slot \(t\) Alg. 1 has the constraints in Eqs. (4b) and (5b) only until slot \(t\) (but has no entire set of constraints).

Alg. 1 initializes all the variables to zeros in Line 1. For each new slot \(t\), Alg. 1 updates all variables according to the present channel state \(s(t)\) without the future channel state pattern. Line 3 identifies a maximum value (denoted by \(k^*_t\) for slot \(t\)) of \(k\) such that \(I_{i,k}(t) = 1\) for all \(i\). The underlying idea is that for each slot \(t\) our scheduling algorithm (in Section III-D) decides either \(d(t) = k^*_t\) with some probability or \(d(t) = 0\) otherwise. The decision \(d(t) = k^*_t\) allocates the minimum power for successfully broadcasting to all users in slot \(t\); in turn, all queues flushes their packets in slot \(t\).

That probabilistic decision is based on the present value of \(x_{k^*_t}(t)\) updated in Line 4. For each slot \(t\), Alg. 1 updates the value of \(x_{k^*_t}(t)\) by iteration (in Line 4) from iteration \(j = 1\) (for the first arriving packet) until iteration \(j = t\) (for the newly arriving packet) if the condition in Line 6 holds. To understand the idea behind the condition in Line 6 and the update in Line 7, we interpret the value of \(\sum_{\tau=1}^{\tau_{max}} I_{i,k}(\tau) x_k(\tau)\) in Eq. (4b) as the cumulative probability that packet \(j\) gets flushed by slot \(t\) since its arrival (in slot \(j\)). Note that for each slot \(\tau\) Alg. 1 updates the value of \(x_{k^*_t}(\tau)\) only (in Line 7) but keeps the value of \(x_k(\tau)\) for all \(k \neq k^*_t\) unchanged. Also note that \(I_{i,k^*_t}(\tau) = 1\) for all \(\tau\).
Thus, the value of $\sum_{\tau=j}^{t} \sum_{k=1}^{M} 1_{i,k}(\tau)x_{k}^{*}(\tau)$ becomes that of $\sum_{\tau=j}^{t} x_{k}^{*}(\tau)$. In other words, the value of $\sum_{\tau=j}^{t} x_{k}^{*}(\tau)$ implies the cumulative flushing probability of packet $j$ by slot $t$.

With the above interpretation, the condition in Line 5 indicates if packet $j$ has been flushed by slot $t$. On one hand, if $\sum_{\tau=j}^{t} x_{k}^{*}(\tau) \geq 1$, then packet $j$ has been flushed; thus, no variable needs to be updated. On the other hand, if $\sum_{\tau=j}^{t} x_{k}^{*}(\tau) < 1$, then packet $j$ might still exist in slot $t$; thus, its associated variables get updated. For each packet that might exist in slot $t$, Line 7 increases the value of $x_{k}^{*}(t)$, i.e., the more packets exist, the higher the flushing probability is.

Moreover, according to Line 8 the cumulative flushing probability $\sum_{\tau=j}^{t} x_{k}^{*}(\tau)$ gets updated by

$$\sum_{\tau=j}^{t} x_{k}^{*}(\tau) \leftarrow \sum_{\tau=j}^{t} x_{k}^{*}(\tau) + \frac{1}{C_{k}^{*}} \sum_{\tau=j}^{t} x_{k}^{*}(\tau) + \frac{1}{\theta \cdot C_{k}^{*}} \left(1 + \frac{1}{C_{k}^{*}} \right) \sum_{\tau=j}^{t} x_{k}^{*}(\tau) + \frac{1}{\theta \cdot C_{k}^{*}}$$

That is, Line 8 increases the cumulative flushing probability with the multiplicative scale of $1 + 1/C_{k}^{*}$ and the additive scale of $1/(\theta \cdot C_{k}^{*})$. The term $C_{k}^{*}$ appears in the denominators because the higher the value of $C_{k}^{*}$ is, the less the flushing probability rises. Moreover, the value of the constant $\theta$ in Line 8 is specified in Line 2 for satisfying the dual constraints in Eq. 5b. We want to emphasize that, with the decision $d(t)$, all queues’ sizes are the same for each slots $t$. Thus, Line 8 has been scaled to capture all packet $j$’s for each iteration $j$. Such a user-number-independent update for each iteration can reduce the computational complexity of our channel-agnostic algorithm (in Section III-D).

In addition, Line 9 updates the value of $z_{i,j}(t)$ to that of $1 - \sum_{\tau=j}^{t} x_{k}^{*}(\tau)$ for satisfying the primal constraints in Eq. 4b. Finally, Line 10 updates the value of $y_{i,j}(t)$ to $1/N$ for maximizing the dual objective value in Eq. 5a.

We analyze the primal objective value in Eq. 4a computed by Alg. 1 as follows (dy the duality theory).

**Theorem 1.** The primal objective value in Eq. 4a computed by Alg. 1 at the end of slot $T$ is bounded above by

$$\left(1 + \frac{1}{(1 + \frac{1}{C_{M}^{*}})C_{1}^{*} - 1}\right) OPT(s),$$

for all possible entire channel state pattern $s$.

**Proof.** See Appendix A

**D. Online scheduling algorithm**

Leveraging the value of $x_{k}^{*}(t)$ produced by Alg. 1, this section proposes a (randomized) online scheduling algorithm in Alg. 2. Alg. 2 updates variable $x_{k}^{*}(t)$ in Line 7 in the same way as Alg. 1 does. Moreover, Alg. 2 introduces additional variables $x_{pre-sum}$ and $x_{sum}$. While the value of $x_{pre-sum}$ in slot $t$ is the cumulative value of $\min\{x_{k}^{*}(t), 1\}$ until $t-1$ (see Line 10), the value of $x_{sum}$ in slot $t$ is the cumulative value of $\min\{x_{k}^{*}(t), 1\}$ until slot $t$ (see Line 11).

Alg. 2 picks a uniformly random number $u \in [0,1)$ in Line 4. According to Lines 12 and 14, if there exists a $k \in \mathbb{N}$ such that $u + k \in [x_{pre-sum}, x_{sum})$, then the AP decides $d(t) = k_{i}^{*}$ for slot $t$ (see Line 14), i.e., the AP allocates the minimum power such that all users can receive the update; otherwise, the AP decides $d(t) = 0$ (see Line 16). The idea behind Alg. 2 is that, with the uniformly random choice of $u$, the probability of broadcasting to all users (or the probability of flushing all queues) in slot $t$ becomes $\min\{x_{k}^{*}(t), 1\}$ and the cumulative probability of flushing packet $j$ in slot $t$ since its arrival is $\min\{\sum_{\tau=j}^{t} x_{k}^{*}(\tau), 1\}$. The similar idea was also used to generate random numbers for a given cumulative distribution function (CDF).

We want to emphasize that always choosing $d(t) = k_{i}^{*}$ for each transmission (such that everyone can receive the update) might not be optimal. However, we can show that the expected competitive ratio (over the randomness of $u$) of our algorithm can be guaranteed as follow.

**Theorem 2.** The expected competitive ratio of Alg. 2 is

$$\left(1 + \frac{1}{(1 + \frac{1}{C_{M}^{*}})C_{1}^{*} - 1}\right).$$

**Proof.** See Appendix B

The competitive ratio in Theorem 2 is independent of the number $N$ of users, the entire channel state pattern $s$ (i.e., their movement), and the time horizon $T$. Moreover, when $C_{1}^{*}$ is large (compared with $C_{M}^{*} - C_{1}^{*}$), the expected competitive ratio approaches $e/(e-1) \approx 1.58$.

Regarding the computational complexity of Alg. 2 we note that Line 4 takes $O(\log N)$ for the minimum search. Moreover, following [4], we can show that there are at most $\sqrt{C_{1}^{*}}$ itera-
tions such that the condition in Line 5 holds, i.e., we can revise
the iteration in Line 5 to starting from
s
i
(t) if the condition in Line 5 holds, i.e., we can revise
the number of users.

E. Channel-agnostic scheduling algorithm

This section develops a channel-agnostic scheduling algo-

rithm by modifying Algs. 1 and 2. Instead of searching for
the minimum power level such that all users can receive the
update (in Line 3 of Alg. 1 and Line 4 of Alg. 2), the
channel-agnostic scheduling algorithm modifies both lines by
setting k∗
M
i
← M for all slot t. Moreover, the channel-agnostic
scheduling algorithm modifies Line 13 of Alg. 2 to

d(t) ← M, i.e., always allocating the maximum power level with
some probability. Following the proofs of Theorems 1 and 2 (with
minor modifications, e.g., modify C
k
i
 to C
M
 in Eq. (6)), we
can show that the channel-agnostic scheduling can achieve the
same competitive ratio as Alg. 2 does in Theorem 2. Moreover,
the computational complexity of the channel-agnostic
scheduling algorithm for each slot is O(√C
T
), independent of
the number of users.

IV. NUMERICAL RESULTS

While Theorem 2 analyzes the proposed algorithms in the
adversarial (worst-case) scenario, this section validates them
in stochastic scenarios. We run our algorithms under the 4-
state Markov-modulated channel in Fig. 1 (with random initial
states) for 10,000 slots. Moreover, we set the transmission cost
to C
k
= C
k+5(k−1) for k = 1, · · · , 4. While Fig. 2 simulates
the time-average total cost (∑
T
i=1(C
d
i
(t) + ∑
N
i=1 a
i
(t)/N))/T and the time-average age (∑
T
i=1(∑
N
i=1 a
i
(t)/N))/T for various values of C
T
(fixed N = 5), Fig. 3 does for various values of N (fixed C
T
= 30). In the figures, we compare the
proposed algorithms with two online greedy algorithms. Greedy Alg. 1 chooses d(t) for minimizing the total cost
C
d
i
(t) + (∑
N
i=1 a
i
(t)/N) at the end of slot t. Let g
i
(t) be the
cumulative age cost for user i from the slot when it received
the previous update until the present slot t, i.e., g
i
(t) = 0 if user i can receive the update in slot t; otherwise, g
i
(t) = g
i
(t − 1) + a
i
(t). Greedy Alg. 2 chooses d(t) for minimizing
C
d
i
(t) + (∑
N
i=1 g
i
(t)/N) at the end of slot t. We can observe that Greedy Alg. 2 outperforms Greedy Alg. 1. That is because Alg. 1 neglects the cost incurred by the stale information in the
previous slots. Moreover, we can observe that our algorithms
significantly outperform the greedy algorithms in both cost (in
Figs. 2-(a) and 3-(a)) and age (in Figs. 2-(b) and 3-(b)). Our
algorithms are more efficient in the use of power. In particular,
the channel-agnostic scheduling algorithm is not only simple,
but also performs almost as well as the online scheduling
algorithm does.

V. CONCLUSION

This letter treated a wireless network where an access point
(AP) broadcasts timely updates to several mobile users. We
developed an online scheduling algorithm and a channel-
agnostic scheduling algorithm for striking a balance between
the freshness of information (for the users) and the power
consumption (for the AP). The proposed algorithms can achieve a
constant competitive ratio, independent of the number of users
and their movement.

APPENDIX A

PROOF OF THEOREM 1

The proof needs the following technical lemma.

Lemma 3. Alg. 1 produces a feasible solution to the primal
program and the dual program.

Proof. The solution produced by Alg. 1 satisfies the constrains in Eq. (4b) according to Line 6. Moreover, the solution
produced by Alg. 1 also satisfies the constraints in Eq. 5b. 
because, for each power level $k$ and slot $t$, we can obtain

$$\sum_{i=1}^{N} 1_{i,k}(t) \sum_{j=1}^{t} y_{i,j}(\tau) \leq \sum_{i=1}^{N} \sum_{j=1}^{t} y_{i,j}(\tau)$$

$$= \sum_{j=1}^{t} \sum_{i=1}^{N} \left( \sum_{\tau=t}^{T} \frac{1}{N} \right) \cdot 1_{\text{update in iteration } j \text{ of slot } \tau}$$

$$= \sum_{j=1}^{t} 1_{\text{update in iteration } j \text{ of slot } \tau} \leq \left[ C_{1} \right] \leq C_{k},$$

where $1_{\text{update in iteration } j \text{ of slot } \tau}$ indicates if the condition in Line 5 holds for iteration $j$ of slot $t$ (such that $y_{i,j}(\tau)$ gets updated). The equality in (a) is according to Line 8. Moreover, the inequality in (b) follows the lines in [6].

Finally, the solution produced by Alg. 1 also satisfies the constraints in Eq. (5c) according to Line 8.

Then, let $\Delta \mathcal{P}_{j}(t)$ be the increment of the primal objective value in Eq. (4a) caused by Alg. 1’s iteration $j$ of slot $t$. First, if the condition in Line 5 of Alg. 1 holds in iteration $j$ of slot $t$, then $\Delta \mathcal{P}_{j}(t)$ becomes

$$\Delta \mathcal{P}_{j}(t) = C_{k_{j}}(t) \left( \frac{1}{C_{k_{j}}} \sum_{\tau=t}^{T} x_{k_{j}}(\tau) + \frac{1}{\theta} \cdot C_{k_{j}} \right)$$

$$+ \left( 1 - \sum_{\tau=t}^{T} x_{k_{j}}(\tau) \right) N = 1 + \frac{1}{\theta}, \quad (6)$$

where (a) is according to Line 7, (b) is according to Line 8. Second, if the condition in Line 5 of Alg. 1 fails in iteration $j$ of slot $t$, then $\Delta \mathcal{P}_{j}(t) = 0$.

Similarly, let $\Delta \mathcal{D}_{j}(t)$ be the increment of the dual objective value in Eq. (5a) caused by Alg. 1’s iteration $j$ of slot $t$. First, if the condition in Line 5 holds in iteration $j$ of slot $t$, then $\Delta \mathcal{D}_{j}(t) = (1/N) \cdot N = 1$ according to Line 8. Second, if the condition in Line 5 fails in iteration $j$ of slot $t$, then $\Delta \mathcal{D}_{j}(t) = 0$. Thus, we can establish that $\Delta \mathcal{D}_{j}(t) = (1 + \frac{1}{\theta}) \mathcal{D}_{j}(t)$ for all $j$ and $t$.

Let $\mathcal{P}$ and $\mathcal{D}$ be the primal objective value and the dual objective value computed by Alg. 1 respectively. Then, $\mathcal{P} = \sum_{t=1}^{T} \sum_{j=1}^{t} \Delta \mathcal{P}_{j}(t)$ and $\mathcal{D} = \sum_{t=1}^{T} \sum_{j=1}^{t} \Delta \mathcal{D}_{j}(t)$; moreover,

$$\mathcal{P} = \left( 1 + \frac{1}{\theta} \right) \mathcal{D} \leq \left( 1 + \frac{1}{\theta} \right) OPT(s),$$

where (a) is from the weak duality theory [7]. Substituting $\theta$ in the above inequality by its definition yields the theorem.