Online Scheduling for Delayed Mobile Offloading

Han Deng
Department of ECE
Texas A&M University
College Station, TX 77840, USA
Email: hdeng@email.tamu.edu

I-Hong Hou
Department of ECE
Texas A&M University
College Station, TX 77840, USA
Email: ihou@tamu.edu

***Abstract*—WiFi offloading, where mobile users opportunistically obtain data through WiFi rather than through cellular networks, is a promising technique to greatly improve spectrum efficiency and reduce cellular network congestion. We consider a system where the service provider deploys multiple WiFi hotspots to offload mobile traffic, and study the scheduling policy to maximize the amount of offloaded data. Since the movements of users are unpredictable, we focus on online scheduling policy where APs do not have any knowledge about the users' mobility patterns.**

We study performance of online policies by comparing against the optimal offline policy. We prove that any work-conserving policy is able to offload at least half as much data as the offline policy, and then propose an online policy that can offload $(e-1)/e$ as much data as the offline policy. We further study the case where the service provider can increase the capacity of WiFi so as to provide guarantees on the amount of offloaded data. We propose a simple online policy and prove that our policy only needs half as much capacity as current mechanism to provide the same performance guarantee.

I. Introduction

With the increasing number of smart phone users subscribing to 3G/4G networks, the mobile data traffic grows rapidly in recent years. The global mobile data traffic growth rate in 2013 exceeds 81%, and is expected to grow at a 61% compound annual growth rate (CAGR) from 2013 to 2018 [1]. Cellular networks face the challenge of serve this great increase in data consumption.

Since interference between links is the major obstacle to dramatically increasing the capacity of wireless networks, many studies have been proposed to migrate traffic from the high-power and high-interference macro base stations to networks with smaller transmission power and interference, such as femtocells [2], WiFi [3], and mobile-to-mobile opportunistic networks [4]. Offloading traffic through WiFi has been shown to be an effective way to reduce the mobile traffic [5] [3]. WiFi is faster and uses less energy to transmit data when there is a connection [5]. Thus WiFi can significantly reduce the mobile traffic through macro base stations in the next several years. For instance, 45% of the global mobile traffic is offloaded using WiFi in 2013, and the rate is estimated to raise to 52% in 2018 [1].

In this paper, we study the problem of using WiFi for delayed mobile traffic. We consider the downlink scenario where each mobile user needs to obtain some data from service provider before a certain deadline. When multiple users connecting to one WiFi access point (AP), the AP makes decision on which user to serve. Mobile users move around the system, which results in intermittent WiFi connections. They aim to use WiFi as much as possible, and will only use cellular networks to obtain their needed data after their respective deadlines. We aim to design scheduling policies for WiFi APs that maximize the amount of data offloaded to WiFi.

We show that the problem of maximizing the amount of offloaded data can be formulated as a linear programming problem, and an offline policy can solve it with standard linear programming techniques. However, such a formulation requires the knowledge of mobility patterns of all mobile users in advance. Instead, we study the performance of online scheduling polices that make scheduling decisions only based on system history and the current locations of users. When all APs use the same transmission rates for any connected users, we show that any work-conserving scheduling policy is able to offload at least 50% as much data as the optimal offline policy. On the other hand, when APs may use different transmission rates for different users based on their individual channel qualities, we propose a simple online algorithm that guarantees to deliver at least $\frac{e-1}{e}$ as much data as the optimal offline policy.

A minimum fraction of $\frac{e-1}{e}$ of data download from WiFi may not be sufficient to reduce the congestion. Hence, we further investigate the case when wireless service providers have a hard requirement on the amount of data offloaded to WiFi, so as to reduce cellular network congestion, and they are willing to increase the capacity of WiFi to meet this requirement. We then study the amount of capacity needed to provide offload guarantees for online scheduling policies. We propose a simple online scheduling policy and prove that, in order to offload at least $\frac{1}{\beta}$ as much data as the optimal offline policy, our policy needs to increase the capacity by approximated $\frac{1}{2(\beta-1)}$. On the other hand, even when APs only use a fixed transmission rate, the commonly-used round-robin policy needs to increase the capacity by at least $\frac{1}{\beta-1}$ to provide the same guarantee. In other words, our policy only needs half as much capacity to provide the same performance guarantee. We also derive a theoretical lower-bound on the amount of capacity needed to provide performance guarantees.

Theoretical analysis only shows that the worst-case performance of our policies is better than that of the round robin policy. We further conduct simulations to evaluate the performance of scheduling policies for a randomly generated system. Simulation results show that our policies still outperform the round robin policy on average. In particular, our policies offload 10% more traffic when APs only use a fixed transmission rate, and offload more than 80% more traffic when they may user variable rates for different users.

The rest of the paper is organized as follows: Section II reviews some existing work on WiFi offloading. Section III introduces our system model and problem formulation. Section IV introduces some basic linear programming properties that are vital to this work. Section V studies the amount of offloaded data by online scheduling policies when the APs have unit capacity. Section VI further studies the case when wireless service providers can increase the capacity of WiFi to provide performance guarantees. Section VII provides some simulation results. Finally, Section VIII concludes this paper.

II. Related Work

Many experimental studies have shown that mobile offloading is promising. Gass and Diot [6] compare WiFi and 3G network through experiments and show that WiFi is able to download more data than 3G network even if though connecting time is shorter. Balasubramanian et al. [7] study the availability of 3G and WiFi network from moving cars in three cities, and find that WiFi suffers greatly from limited connectivitly. They then propose a system called Wiffler to significantly improve the amount of offloaded traffic. Lee et al. [3] study the WiFi offload performance through an experiment with 100 iPhone users in Seoul, and observe that WiFi can upload about 65% of the traffic. Mota et al. [8] study the WiFi hotspots availability during bus routes in Paris, and show that current WiFi in Paris can offload up to 30% of mobile traffic.

An important challenge for mobile offloading is the unknown mobility patterns of mobile users. There are several studies that focus on deriving models for mobility patterns [9]–[11]. Cheung and Huang [12] study the WiFi offloading problem by formulating the problem as a finite-horizon Markov decision process by using the prediction in [11]. Li et al. [13] study using a small set of mobile users to offload data, and propose a policy based on submodular optimization. Whitbeck et al. [14] consider using offloading to reduce the burden in broadcasting messages. Hou et al. [15] propose a transport layer protocol to integrate 3G and WiFi networks for vehicular network access. Barbieri et al. [16] propose a system design for mobile offloading with pico base stations. Bennis et al. [17] and Singh et al. [18] consider the problem of network self-organizing for offloading traffic. Bilgir Yetim and Martonosi [19] propose offline scheduling policies for WiFi offloading.

III. System Model

We consider a system where mobile users move within the area of a cellular network. In order to reduce the congestion of the cellular network, the cellular operator deploys a number of WiFi hotspots within the region. We use $\mathcal{I}$ to denote the set of mobile users and $\mathcal{M}$ to denote the set of WiFi APs. Mobile users may enter the system at different times and at different locations. Upon entering the system, a mobile user i specifies the amount of data, denoted by C_i, that it needs to obtain, and a deadline T_i. The mobile user moves around the system and tries to obtain data from WiFi APs whenever possible. At time T_i, the mobile user downloads all the remaining data from the cellular network directly.

We assume that time is slotted and numbered as $t = 1, 2, \dots$. The location of a mobile user may change from time to time and it determines the connectivity and channel capacity between APs and itself. Since APs do not have users' location information in advance, channel conditions are unpredictable. Each AP makes scheduling decisions based on the past transmission history and current channel conditions. We use K_{imt} to denote the channel capacity between AP m and mobile user i at time t. If i cannot be connected to m at time t, we have $K_{imt} = 0$. As WiFi hotspots are usually far away from each other, we assume that, for each i and t, there is at most one AP m with $K_{imt} > 0$. Also, since i cannot download any data prior to its entrance, and it will use the cellular network to download data after its deadline, we set $K_{imt} = 0$ for all t prior to i's entrance or after its deadline. We normalize the system so that $0 \leq K_{imt} \leq 1$ for all i, m, t.

Each AP employs some scheduling policy to determine the portion of time it spends transmitting to each mobile user i at each time t, denoted by X_{imt}. The amount of data that mobile user i obtains from AP m at time t is then $K_{imt}X_{imt}$. Our goal is to design a scheduling policy that maximizes the total amount of data that are delivered through WiFi, which, in turn, minimizes the amount of data through the congested cellular network. Since each mobile user i needs to obtain C_i data, we formulate the following linear programming problem:

Offload:

$$Max \sum_{imt} X_{imt}K_{imt} \tag{1}$$

$$s.t. \sum_{mt} X_{imt}K_{imt} \leq C_i, \forall i \in \mathcal{I}, \tag{2}$$

$$\sum_{i} X_{imt} \leq 1, \forall m \in \mathcal{M}, t, \tag{3}$$

$$X_{imt} \geq 0, \forall i \in \mathcal{I}, m \in \mathcal{M}, t. \tag{4}$$

We use Γ_{opt} to denote the optimal value of $\sum_{imt} X_{imt}K_{imt}$ in the above problem. While this problem can be solved by standard linear programming techniques, doing so requires the knowledge of the entrance times and locations of all mobile users at time 0, which is impractical. Instead, we aim to derive online policies that choose the values of X_{imt} solely based on system history up to time t. We let $\Gamma_\eta(1)$ be the value of $\sum_{imt} X_{imt}K_{imt}$ under η, given K_{imt} and C_i.

We assume that the cellular operator may be able to increase the capacity of WiFi hotspots by, for example, upgrading APs or obtaining more spectrum. When the capacity of WiFi hotspots is increased by R, the channel capacity between i and m at time t becomes RK_{imt}. Equivalently, we can also describe the system as one with channel capacity K_{imt}, but the AP can spend an amount of R time transmitting to clients in each slot, that is, $\sum_i X_{imt} \leq R$. Therefore, we consider the following linear programming problem when the capacity is increased by R:

Offload(R):

$$Max \sum_{imt} X_{imt}K_{imt} \tag{5}$$

$$s.t. \sum_{mt} X_{imt}K_{imt} \leq C_i, \forall i \in \mathcal{I}, \tag{6}$$

$$\sum_{i} X_{imt} \leq R, \forall m \in \mathcal{M}, t, \tag{7}$$

$$X_{imt} \geq 0, \forall i \in \mathcal{I}, m \in \mathcal{M}, t. \tag{8}$$

Let $\Gamma_\eta(R)$ be the value of $\sum_{imt} X_{imt}K_{imt}$ for the Offload(R) problem under policy η. We evaluate the performance of η by its *competitive ratio*, which is defined slightly differently from most existing literature.

Definition 1: A policy η is said to be (R, β)-competitive if $\Gamma_{opt}/\Gamma_\eta(R) \leq \beta$, as $\min_{i\in\mathcal{I}} C_i \to \infty$, for all systems.

We note that when $R = 1$, the corresponding β becomes the competitive ratio commonly defined in existing literature. Our definition is richer in that it characterizes the amount of capacity needed to provide performance guarantees. Since the very reason of using WiFi offloading is that the cellular network is congested, the operator may have a hard requirement on the amount of data being offloaded through WiFi, and it is willing to purchase better equipments and more spectrum to achieve this requirement. In this case, it needs to know how much capacity is needed. Suppose the optimal offline policy can offload all data through WiFi, and the operator requires a portion $1/\beta$ of the data to be offloaded, our definition then reveals that the capacity needs to be increased by R so that the employed policy is (R, β)-competitive.

IV. Preliminary

This section introduces some basic theorems that will be used in this paper.

A standard form of linear programming problem (LP) is:

$$(P) : Max \sum_{i=1}^{n} c_i x_i,$$

$$s.t. \sum_{i=1}^{n} a_{ij}x_i \leq b_j, \forall 1 \leq i \leq n,$$

$$x_i \geq 0,$$

and its dual is

$$(D) : Min \sum_{j=1}^{m} b_j y_j,$$

$$s.t. \sum_{j=1}^{m} a_{ij}y_j \geq c_i, \forall 1 \leq j \leq m,$$

$$y_j \geq 0.$$

We have the following two fundamental theorems:

Theorem 1 (Weak Duality [20]): Let $\{x_i\} \in \mathbb{R}^n$ and $\{y_j\} \in \mathbb{R}^m$ satisfy the constraints of the primal (P) and the dual (D) LPs, respectively, then:

$$\sum_{i=1}^{n} c_i x_i \leq \sum_{j=1}^{m} b_j y_j.$$

Theorem 2 (Complementary Slackness [20]): Let $\{x_i\} \in \mathbb{R}^n$ and $\{y_i\} \in \mathbb{R}^m$ satisfy the constraints of the primal (P) and dual (D) LPs, respectively. Further, $\{x_i\}$ and $\{y_i\}$ have the following properties:

- If $x > 0$, then $c_i \leq \sum_{j=1}^{m} a_{ij}y_i \leq \beta \cdot c_i$ for some $\beta > 1$;
- If $y > 0$, then $\sum_{i=1}^{n} a_{ij}x_i = b_j$;

Then:

$$\sum_{j=1}^{m} b_j y_j \leq \beta \cdot \sum_{i=1}^{n} c_i x_i.$$

V. Competitive Ratio with Unit Capacity

In this section, we discuss the special case with $R = 1$. We first show that when K_{imt} is either 0 or 1, any work-conserving policy is $(1, 2)$-competitive. We then study the case when K_{imt} can be any real number in $[0, 1]$. We propose a simple online scheduling policy and prove that it is $(1, \frac{e}{e-1})$-competitive.

A. Performance of Work-Conserving Policy under On-Off Channels

We first consider the case where K_{imt} is either 0 or 1, which is usually referred as *On-Off channels*, and we say that client i is *connected* to AP m at time t if $K_{imt} = 1$. We study the performance of *work-conserving scheduling policy*, under which each AP m selects to serve one connected client that has yet to receive all the data, as long as there is one, and only idles when all connected clients have already received all their data.

Theorem 3: Any work-conserving policy is $(1, 2)$-competitive with ON-Off channels.

Proof: The offload problem is shown as (1) to (4), and its dual is

$$(D) : Min \sum_{mt} Y_{mt} + \sum_{i} C_i Z_i, \tag{9}$$

$$s.t.\ Y_{mt} + K_{imt} Z_i \geq K_{imt}, \forall i, m, t, \tag{10}$$

$$Y_{mt} \geq 0, \forall m, t, \tag{11}$$

$$Z_i \geq 0, \forall i, \tag{12}$$

where Y_{mt} is the dual variable for each constraint in (2), and Z_i is the dual variable for each constraint in (3).

We set $X_{imt} = 1$ if client i is served by AP m at time slot t, and $X_{imt} = 0$ otherwise. We set $Y_{mt} = 1$ if AP m schedules a client at time t, and $Y_{mt} = 0$ if m idles at t. We set $Z_i = 1$ if client i have received all its data before its deadline, and $Z_i = 0$ otherwise.

We will use Theorem 2 to establish the theorem. First, we show that X_{imt}, Y_{mt}, and Z_i satisfy the constraints (2) (3), and (10). (2) and (3) are satisfied because each AP schedules at most one client at any time, and it never schedules clients that have already received all their data.

Given i, m, t, if $K_{imt} = 0$, then (10) is satisfied since Y_{mt} and Z_i are non-negative. (10) also holds if $K_{imt} = 1$ and $Y_{mt} = 1$. Finally, if $K_{imt} = 1$ and $Y_{mt} = 0$, i.e., AP m does not schedule any client at time t, then all clients connected to AP m at time t must have already received all their data. Hence, $Z_i = 1$ and (10) still holds.

Next, we verify the complementary slackness conditions. If $Z_i > 0$, then client i obtains all its data, and $\sum_{mt} X_{imt} K_{imt} = C_i$. If $Y_{mt} > 0$, then AP m schedules some client at time t, and $\sum_i X_{imt} = 1$. In addition, if $X_{imt} > 0$, then $K_{imt} = 1$. Thus, $2K_{imt} = 2 \geq Y_{mt} + K_{imt} Z_i \geq K_{imt}$. By Theorem 2, we know $\sum_{mt} Y_{mt} + \sum_i C_i Z_i \leq 2 \cdot \sum_{imt} X_{imt} K_{imt}$. Further, $\Gamma_{opt} \leq \sum_{mt} Y_{mt} + \sum_i C_i Z_i$, by Theorem 1, and hence any work-conserving policy is $(1, 2)$-competitive. ■

B. Online Algorithm for General Channels

We now discuss the general case in which K_{imt} can be any real number between 0 and 1. We propose an online scheduling algorithm and prove that it is $(1, \frac{e}{e-1})$-competitive.

In our algorithm, APs keep track of and update a variable Z_i for each client i. Z_i is initially set to 0. If each time t, each AP m chooses to serve the client i that maximizes $K_{imt}(1 - Z_i)$. AP m then updates Z_i by setting it to be $Z_i(1 + \dfrac{K_{imt}}{C_i}) + \dfrac{K_{imt}}{(d-1)C_i}$. Here d is a value only used in calculation and it is set to be $(1 + 1/C_{min})^{C_{min}}$. We show the value chosen for d is reasonable in proof of Lemma 1. AP m broadcasts the updated Z_i to all APs. Algorithm 1 formally describes the algorithm. In Algorithm 1, we also introduce two other variables, X_{imt} and Y_{mt}. These two variables are only used to establish the competitive ratio, and are not needed in actual implementations.

Algorithm 1

```
 1: Initially, $X_{imt} = 0$, $Y_{mt} = 0$, $Z_i = 0$.
 2: $C_{min} \leftarrow \min_i C_i, d \leftarrow (1 + 1/C_{min})^{C_{min}}$.
 3: for each time slot $t$ do
 4:   for each AP $m$ do
 5:     $i^*_m \leftarrow \underset{i}{\text{argmax}}\{K_{imt}(1 - Z_i)\}$.
 6:     if $K_{i^*_m mt}(1 - Z_{i^*_m}) > 0$ then
 7:       $Y_{mt} \leftarrow K_{i^*_m mt}(1 - Z_{i^*_m})$.
 8:       $Z_{i^*_m} \leftarrow Z_{i^*_m}(1 + \frac{K_{i^*_m mt}}{C_{i^*_m}}) + \frac{K_{i^*_m mt}}{(d-1)C_{i^*_m}}$.
 9:       $X_{i^*_m mt} \leftarrow 1$.
10:       AP $m$ transmits to client $i$ at time $t$.
11:       if $\sum_{p,s \leq t} X_{i^*_m ps} K_{i^*_m ps} > C_{i^*_m}$ then
12:         $X_{i^*_m mt} \leftarrow \frac{C_{i^*_m} - \sum_{p,s<t} X_{i^*_m ps} K_{i^*_m ps}}{K_{i^*_m mt}}$.
13:       end if
14:     end if
15:   end for
16: end for
```

In Algorithm 1, each of Y_{mt} and X_{imt} is only updated at time slot t, while Z_i may be updated in many different time slots. We note that the value of Z_i is non-decreasing in each update.

When m transmits to i at time t, it is possible that i already obtains most of its data and only needs less than K_{imt} data to complete its download. In this case, m uses only a fraction of a time slot to deliver all remaining data that i needs. Step 12 addresses this case, and the total amount of offloaded data is $\sum_{imt} X_{imt} K_{imt}$.

Lemma 1: Let $Z_i^{(t)}$ be the value of Z_i at the end of time slot t. Then,

$$Z_i^{(t)} \geq (\frac{1}{d-1})(d^{\sum_{m,s \leq t} \frac{X_{imt} K_{imt}}{C_i}} - 1). \tag{13}$$

Proof: We prove (13) by induction on t.

When $t = 0$, $Z_i^{(t)} = 0 = (\frac{1}{d-1})(d^0 - 1)$, and (13) holds.

Suppose (13) holds for all time before s. Consider time $t = s + 1$. If i is not scheduled at $s + 1$, $X_{imt} = 0$ for all m at $t = s + 1$ and $Z_i^{(s+1)} = Z_i^{(s)}$. Hence (13) holds.

On the other hand, if i is scheduled by AP p at time

$s+1$,

$$\begin{aligned}{Z_i}^{(s+1)} &= {Z_i}^{(s)}(1+\frac{K_{ip(s+1)}}{C_i}) + \frac{K_{ip(s+1)}}{(d-1)C_i}\\ \geq&\frac{1}{(d-1)}(d^{\sum_{m,t\leq s}\frac{X_{imt}K_{imt}}{C_i}}-1)(1+\frac{K_{ip(s+1)}}{C_i})\\ &+\frac{K_{ip(s+1)}}{(d-1)C_i}\\ =&\frac{1}{(d-1)}[d^{\sum_{m,t\leq s}\frac{X_{imt}K_{imt}}{C_i}}(1+\frac{K_{ip(s+1)}}{C_i})-1]\end{aligned}$$

It is easy to verify that $ln(1+x)/x$ is decreasing when $x \in [0,1]$. Thus $(1+y) \geq (1+x)^{(y/x)}$ for $x \geq y$. Let $y = \frac{K_{ip(s+1)}}{C_i}$ and $x = \frac{1}{C_{min}}$. We then have

$$\begin{aligned}&{Z_i}^{(s+1)}\\ \geq&\frac{1}{(d-1)}[(d^{\sum_{m,t\leq s}\frac{X_{imt}K_{imt}}{C_i}})(1+\frac{1}{C_{min}})^{\frac{K_{ip(s+1)}C_{min}}{C_i}}-1]\end{aligned}$$

Recall that the value of d is $(1+1/C_{min})^{C_{min}}$. Thus

$${Z_i}^{(s+1)} \geq \frac{1}{(d-1)}(d^{\sum_{m,t\leq s+1}\frac{X_{imt}K_{imt}}{C_i}}-1),$$

and (13) holds. By induction, (13) holds for all t. ■

Theorem 4: Algorithm 1 is $(1, \frac{e}{e-1})$-competitive.

Proof: The offload problem and and its dual are stated as (1) to (4), and (9) to (12), respectively. We prove Algorithm 1 is $(1, \frac{e}{e-1})$-competitive by the following steps:

First, we show that the dual solutions $\{Y_{mt}\}$ and $\{Z_i\}$ satisfy constraints (10) to (12).

Since $i^*_m \leftarrow \underset{i}{\text{argmax}}\{K_{imt}(1-Z_i)\}$, we have:

$$K_{i^*_m mt}(1-Z_{i^*_m}) \geq K_{imt}(1-Z_i), \forall i, m, t.$$

Further, by step 7 in Algorithm 1, we have:

$$\begin{aligned}&Y_{mt}+K_{imt}Z_i-K_{imt}\\ \geq&K_{i^*_m mt}(1-Z_{i^*_m})+K_{imt}Z_i-K_{imt}\\ \geq&K_{imt}(1-Z_i)+K_{imt}Z_i-K_{imt}=0.\end{aligned}$$

Thus (10) is satisfied. It is easy to check that Y_{mt} and Z_i are non-negative, and (11) and (12) hold.

Second, we show that X_{imt} satisfy constraints (2) to (4). Step 12 ensures that (2) holds. By Lemma 1, $Z_i^{(t)} < 1$ only when i does not receive all its data on or before time t. Hence, X_{imt} is only updated if $\sum_{p,s<t} X_{ips}K_{ips} < C_i$, which makes (3) and (4) hold.

Third, we show that every time steps 7 – 9 are invoked, the ratio between the change of the dual objective function (9) and change of the primal objective function (1) is $\frac{d}{d-1}$. We note that we ignore the change of (1) by step 12 now, which will be taken into account later.

When m schedules i at time t, X_{imt} is increased from 0 to 1, and (1) is increased by K_{imt}. Meanwhile, (9) is increased by

$$\begin{aligned}&K_{imt}(1-Z_i^{(t-1)})+C_i(Z_i^{(t-1)}\frac{K_{imt}}{C_i}+\frac{K_{imt}}{(d-1)C_i})\\ =&(1+\frac{1}{d-1})K_{imt}.\end{aligned}$$

Thus the ratio between change of (9) and (1) is $1 + \frac{1}{d-1} = \frac{d}{d-1}$.

Let Γ_{opt} be the optimal value of (1), $\Gamma_{dual,\eta}$ be the value of (9) under Algorithm 1, and $\Gamma^*_{prim,\eta}$ be the value of (1) under Algorithm 1 without step 12. We have established that $\Gamma_{opt} \leq \Gamma_{dual,\eta} = \frac{d}{d-1}\Gamma^*_{prim,\eta}$, where $\Gamma_{opt} \leq \Gamma_{dual,\eta}$ because of Theorem 1.

Finally, we address the influence of step 12. Step 12 is only invoked when i^*_m obtains all its data, i.e., $\sum_{p,s} X_{i^*_m ps}K_{i^*_m ps} = C_{i^*_m}$. By Lemma 1, step 12 is invoked at most once for each client. Further, when step 12 is invoked, (1) decreases by no more than $K_{i^*_m mt} \leq 1$. Let $\Gamma_{prim,\eta}$ be the value of (1) under Algorithm 1 with step 12. We now have $\Gamma_{prim,\eta} \geq \Gamma^*_{prim,\eta}(1-\frac{1}{C_{min}}) \geq \Gamma_{opt}\frac{d-1}{d}(1-\frac{1}{C_{min}})$. Since $d \rightarrow e$, as $C_{min} \rightarrow \infty$, Algorithm 1 is $(1, \frac{e}{e-1})$-competitive. ■

VI. Competitive Ratio for Variable Capacity

In the previous section, we obtain a $(1, \frac{e}{e-1})$-competitive online algorithm. Thus, Algorithm 1 guarantees to offload 63% as much data as an optimal offline algorithm does. However, this also indicates that, when the optimal offline algorithm offloads all data, our algorithm may miss almost 37% of the data. Then, how much capacity is needed to guarantee offloading, say, 95% of the data? We will focus on this problem in this section.

It is first of interests to study whether it is feasible to increase the capacity by R times so as to guarantee an onine algorithm can always offloads as much data as an optimal offline algorithm with unit capacity does. Or, with our terminology, to study whether there exists a $(R, 1)$-competitive policy. The following example shows that $(R, 1)$-competitive policy does not exists, for any R.

Example 1: Fix R. Consider a system consisting of $N = R+1$ clients and one AP with On-Off channels. Each client has a file size of C. One of the clients, say, client 1, is connected to the AP only in times $1 \leq t \leq C$, while all other clients are connected to the AP forever. The optimal offline policy schedules client 1 in times $1 \leq t \leq C$, and then schedules other clients after $t = C$. Hence, the optimal offline policy offloads all data. On the other hand, since online policies do not know which client is connected to the AP only in times $1 \leq t \leq C$, and they can at most offload $RC < NC$ data in times $1 \leq t \leq C$, they cannot guarantee to offload all data. □

When the capacity is increased by R, the corresponding offload problem is described in (5)–(8). In this section, we first propose two online policies and study their competitive ratios. We then derive a theoretical lower-bound for the competitive ratio of all online policies. Finally, we study the competitive ratio of the round robin policy.

A. Two Online Policies and Their Competitive Ratios

We propose two simple online policies and study their competitive ratios. The first one is very similar to Algorithm 1. It is described in Algorithm 2. The only differences are that we choose $d = (1 + 1/C_{min})^{C_{min}/R}$, and we assign $X_{imt} = R$ if i is scheduled by m at time t.

Algorithm 2

```
1:  Initially, $X_{imt} = 0$, $Y_{mt} = 0$, $Z_i = 0$.
2:  $C_{min} \leftarrow \min_i C_i, d \leftarrow (1 + 1/C_{min})^{C_{min}/R}$.
3:  for each time slot $t$ do
4:    for each AP $m$ do
5:      $i^*_m \leftarrow \underset{i}{\text{argmax}}\{K_{imt}(1 - Z_i)\}$.
6:      if $K_{i^*_m mt}(1 - Z_{i^*_m}) > 0$ then
7:        $Y_{mt} \leftarrow K_{i^*_m mt}(1 - Z_{i^*_m})$.
8:        $Z_{i^*_m} \leftarrow Z_{i^*_m}(1 + \frac{K_{i^*_m mt}}{C_{i^*_m}}) + \frac{K_{i^*_m mt}}{(d-1)C_{i^*_m}}$.
9:        $X_{i^*_m mt} \leftarrow R$.
10:       AP $m$ transmits to client $i^*_m$ at time $t$.
11:       if $\sum_{p,s\le t} X_{i^*_m ps}K_{i^*_m ps} > C_{i^*_m}$ then
12:         $X_{i^*_m mt} \leftarrow \frac{C_{i^*_m} - \sum_{p,s<t} X_{i^*_m ps}K_{i^*_m ps}}{K_{i^*_m mt}}$.
13:       end if
14:     end if
15:   end for
16: end for
```

We now study the competitive ratio of Algorithm 2.

Lemma 2: Let $Z_i{}^{(t)}$ be the value of Z_i at the end of time slot t. We have

$$Z_i^{(t)} \ge (\frac{1}{d-1})(d^{\sum_{m,s\le t}\frac{X_{imt}K_{imt}}{C_i}} - 1). \qquad (14)$$

Proof: We prove (14) by induction on t. The proof is similar to that of Lemma 1. ■

Theorem 5: Algorithm 2 is $(R, \frac{e^{1/R}}{R[e^{1/R}-1]})$-competitive. It is approximately $(R, 1 + \frac{1}{2R})$-competitive.

Proof: We prove Algorithm 2 is $(R, \frac{e^{1/R}}{R[e^{1/R}-1]})$-competitive by the following steps:

First, $\{Y_{mt}\}$ and $\{Z_i\}$ satisfy constraints (10)–(12). The proof is the same as the proof for Theorem 4.

Second, we show that X_{imt} satisfy constraints (6)–(8). Step 12 ensures (6). Further, by Lemma 2, $Z_i^{(t)} < 1$ only if $\sum_{m,s\le t} X_{imt}K_{imt} < C_i$. Therefore, a client is only scheduled when it is yet to receive all its data, which ensures (7) and (8).

Third, we show that whenever steps 7 – 9 are invoked, the ratio between the change of (5) and the change of (9) is $\frac{d}{(d-1)R}$. We ignore the change of (5) due to step 12 now.

Suppose client i is scheduled by AP m at time t. We have $X_{imt} = R$, and (5) is increased by RK_{imt}. On the other hand, (9) is increased by

$$\begin{aligned}&K_{imt}(1 - Z_i^{(t-1)}) + C_i(Z_i^{(t-1)}\frac{K_{imt}}{C_i} + \frac{K_{imt}}{(d-1)C_i})\\ =&(1 + \frac{1}{d-1})K_{imt}.\end{aligned}$$

Thus the ratio between the change of objective functions (5) and (9) is $(1 + \frac{1}{d-1})/R = \frac{d}{R(d-1)}$.

Finally, we consider the influence of step 12. Step 12 is only invoked when i^*_m obtains all its data, i.e., $\sum_{p,s} X_{i^*_m ps}K_{i^*_m ps} = C_{i^*_m}$. By Lemma 2, step 12 is invoked at most once for each client. Further, when step 12 is invoked, (5) decreases by no more than $RK_{i^*_m mt} \le R$. Therefore, throughout the system lifetime, the ratio of decrease caused by step 12 is no more than $\frac{R}{C_{min}}$.

As $C_{min} \to \infty$, $d \to e^{1/R}$. By Theorem 1 and the above arguments, we establish that Algorithm 2 is $(R, \frac{e^{1/R}}{R[e^{1/R}-1]})$-competitive.

We can approximate $\frac{e^{1/R}}{R[e^{1/R}-1]}$ by Taylor series as follows:

$$\begin{aligned}\frac{e^{\frac{1}{R}}}{R[e^{\frac{1}{R}}-1]} =&\frac{1 + \frac{1}{R} + \frac{1}{2!R^2} + ...}{R(\frac{1}{R} + \frac{1}{2!R^2} + ...)}\\ =&\frac{1 + \frac{1}{R} + \frac{1}{2!R^2} + ...}{1 + \frac{1}{2!R} + ...}\\ \approx&\frac{1 + \frac{1}{R}}{1 + \frac{1}{2R}} = 1 + \frac{\frac{1}{2R}}{1 + \frac{1}{2R}} \approx 1 + \frac{1}{2R},\end{aligned}$$

when $R \gg 1$.

Thus, the competitive ratio is approximately $(R, 1+\frac{1}{2R})$. Fig. 1 plots $\frac{e^{1/R}}{R[e^{1/R}-1]}$ and $1+\frac{1}{2R}$. As can be seen in the figure, $1 + \frac{1}{2R}$ is a very accurate approximation even for small R. ■

When implementing Algorithm 2, APs need to keep track of Z_i and update the value of Z_i with each other. Further, Algorithm 2 needs to know the value of C_{min} to set d. Below, we describe an approximation of Algorithm 2 that does not require any information exchange.

Using an argument similar to the proof of Lemma 2, we approximate $Z_i^{(t)}$ by $(\frac{1}{d-1})(d^{\sum_{m,s\le t}\frac{X_{imt}K_{imt}}{C_i}} - 1)$. We further approximate d by $e^{1/R}$, and e^x by $1 + x$, for all $0 < x < 1$.

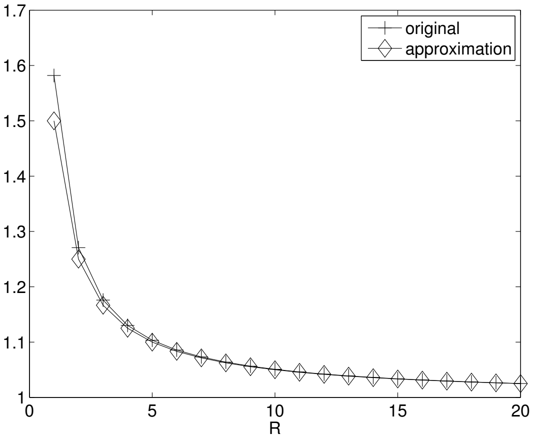

Fig. 1. Illustration of the approximation.

With these approximations, we have

$$\begin{aligned} K_{imt}(1-Z_i) \approx& K_{imt}[1-(\frac{1}{d-1})(d^{\sum_{m,s<t}\frac{X_{ims}K_{ims}}{C_i}}-1)] \\ \approx& \frac{K_{imt}}{d-1}(e^{1/R}-e^{\sum_{m,s<t}\frac{X_{ims}K_{ims}}{RC_i}}) \\ \approx& \frac{1}{(d-1)R}K_{imt}(1-\frac{\sum_{m,s<t}X_{ims}K_{ims}}{C_i}). \end{aligned}$$

Since Algorithm 2 makes AP m schedule the client with the largest $K_{imt}(1-Z_i)$, we can approximate Algorithm 2 by making each AP m schedule the client with the largest $K_{imt}(1-\frac{\sum_{m,s<t}X_{ims}K_{ims}}{C_i})$. Further, we note that $(1-\frac{\sum_{m,s<t}X_{ims}K_{ims}}{C_i})$ is the portion of data that i is yet to obtain. Therefore, this policy simply schedules the client with the largest product of channel capacity and portion of undelivered data, both values are readily available at APs. The policy is summarized in Algorithm 3. It can be easily implemented in a fully distributed fashion. Finally, we note that when all clients need to obtain the same amount of data, i.e. $C_i \equiv C$, then Algorithm 3 becomes the same as the well-known Max-Weight scheduling policy.

Algorithm 3

```
for each time slot $t$ and each AP $m$ do
    $i^*_m \leftarrow \text{argmax}_i\{K_{imt}\frac{\text{Amount of undelivered data}}{C_i}\}$.
    AP $m$ transmits to client $i^*_m$ at time $t$.
end for
```

B. A Lower Bound on Competitive Ratio

In the previous section, we show that the competitive ratio of our online scheduling policy is approximately $(R, 1+\frac{1}{2R})$. In this section, we are interested in the best competitive ratio that online policies can achieve.

Theorem 6: The competitive ratio of any online scheduling policy cannot be better than $(R, 1+\frac{1}{4R})$.

Proof: Given R, we construct a system with one AP and $N=2R$ clients as follows: $C_i = C$ for all i; $K_{11t}=1$ for $1 \le t \le C$, and $K_{11t}=0$ for $t > C$. For $i \neq 1$, $K_{i1t}=1$ for all t. In other words, client 1 is connected to the AP in time $[1, C]$, while all other clients are connected to the AP forever.

If K_{imt} of all clients are known in advance, the optimal offline policy with unit capacity is to schedule client 1 in the first C time slots, and then schedule the rest clients. The optimal offline policy is then able to transmit all NC amount of data.

On the other hand, online policies cannot know K_{imt} in advance. Also, all clients have the same C_i. Therefore, when the system capacity is increased by R times, the best that the AP can do is to evenly distribute its capacity R among all connected clients. The AP delivers a total RC amount of data in the first C time slots, and client 1 receives $\frac{RC}{N}$ amount of data. All other clients receive all their data. Thus, the AP can at best deliver $\frac{RC}{N}+(N-1)C$ amount of data. The competitive ratio is then $\beta = \frac{NC}{\frac{RC}{N}+(N-1)C}$. Since we choose $N=2R$,

$$\begin{aligned} \beta = \frac{NC}{\frac{RC}{N}+(N-1)C} &= \frac{2RC}{\frac{C}{2}+(2R-1)C} \\ &= \frac{4R}{4R-1} > 1+\frac{1}{4R} \end{aligned}$$

Thus, any online scheduling policy cannot have a better competitive ratio than $(R, 1+\frac{1}{4R})$. ■

C. The Competitive Ratio of Round Robin Scheduling

In Section V-A, we have shown that the competitive ratio of any work-conserving policy is at least $(1,2)$ with On-Off channels. In comparison, the competitive ratio of Algorithm 1 is $(1, \frac{e}{e-1}) \approx (1, 1.58)$. It appears that the competitive ratio of any work-conserving policy is close to that of Algorithm 1. We now study the competitive ratio of work-conserving policies when the capacity is increased by R times. In particular, we establish a lower-bound on competitive ratio for the widely used round robin policy. With round robin policy, each AP evenly distributes its capacity among all connected clients.

Theorem 7: Round robin scheduling policy cannot have better competitive ratio than $(R, 1+\frac{1}{R})$.

Proof: Given R, we construct a system with one AP and N clients as follows: $C_1 = C$, and $C_i = \frac{R}{N}C$, for all $i \neq 1$. $K_{11t}=1$ for $1 \le t \le C$, and $K_{11t}=0$ for $t > C$. For $i \neq 1$, $K_{i1t}=1$ for all t. In other words, client 1 is connected to AP in time $[1, C]$, while all other clients are connected to the AP forever.

The optimal offline policy is able to deliver all $C_1 + \sum_{i=2}^{N} C_i$ amount of data with unit capacity, while round robin transmits $\frac{RC_1}{N}+\sum_{i=2}^{N} C_i$ amount of data with R capacity. The competitive ratio of round robin is then at

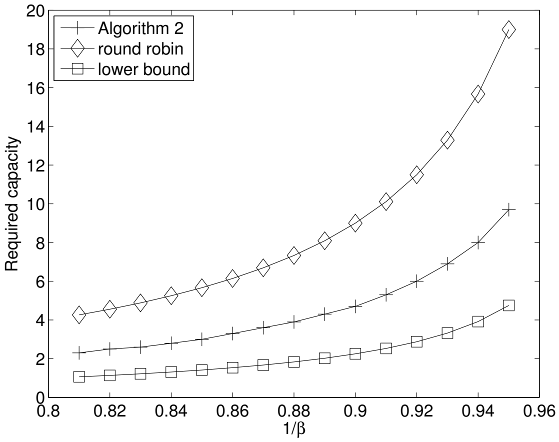

Fig. 2. Capacity requirements of different policies.

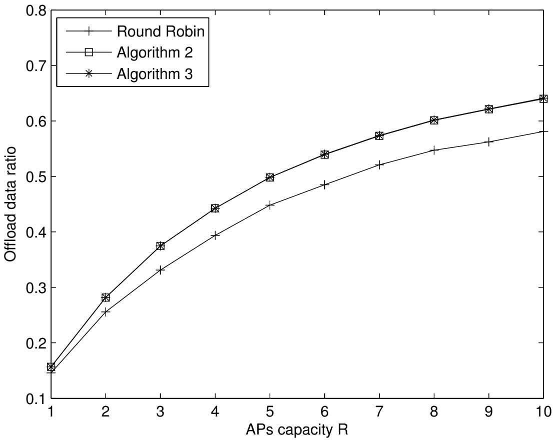

Fig. 3. Performance comparison for On-Off channels.

least:

$$\beta = \frac{C_1 + \sum_{i=2}^{N} C_i}{\frac{RC_1}{N} + \sum_{i=2}^{N} C_i} = \frac{\frac{N}{R} + (N-1)}{1 + (N-1)}$$
$$\to 1 + \frac{1}{R},$$

as $N \to \infty$. ■

For a large R, $\frac{e^{1/R}}{R(e^{1/R}-1}$ is very close to $1+\frac{1}{R}$, and it may seem that the competitive ratio of round robin is close to that of Algorithm 2. However, we can also interpret our results on competitive ratios as follows: Suppose it is required that online policies need to offload at least $\frac{1}{\beta}$ as much data as the optimal offline policy with unit capacity, for some given $\beta > 1$. With Algorithm 2, the system needs to increase its capacity by approximately $\frac{1}{2(\beta-1)}$ times. On the other hand, even with the simple On-Off channels, the round robin policy still needs at least $\frac{1}{\beta-1}$ capacity to achieve the requirement. Finally, Theorem 6 states that any online policy needs at least $\frac{1}{4(\beta-1)}$ capacity. In other words, Algorithm 2 needs about half as much capacity to provide the same guarantee as the round robin policy, and the capacity requirement of Algorithm 2 is about twice as much as the theoretical lower-bound. Thus, our policy is much more preferable to provide stringent performance guarantees. Fig 2 illustrates the capacity requirements for different β. In order to guarantee to offload at least 95% as much data as the optimal offline policy, Algorithm 2 needs to increase the capacity by 9.7 times, while the round robin policy needs to increase the capacity by at least 19 times.

VII. Simulation

In this section, we evaluate the performance of the two algorithms we proposed and the round robin policy. We construct a system with 5 APs and 50 clients. Each client enters the system at any time between$(0, 10)$ and has a deadline between $(10, 510)$. Therefore, the amount of time that a client is in the system is between $(0, 510)$. We set $C_i = 550 -$ (amount of time i is in the system). Each client can either get access to one of the APs or stay out of service in any time slot.For each client, the connect probability is $\frac{5}{6}$ and disconnect probability is $\frac{1}{6}$. When a client is connected to an AP, the distance between the client and the AP is uniformly distributed in $[1, 10]$.

We consider both On-Off channels and general channels. With On-Off channels, $K_{imt} = 1$ if and only if i is connected to m at time t. With general channels, K_{imt} is $1/($distance between i and $m)$ if i is connected to m. For each simulation run, we compute the portion of data that each policy is able to offload to WiFi. All simulation results are the average of 30 simulation runs.

The simulation results for both channels are shown in Figure 3 and Figure 4, respectively. We notice that Algorithm 2 and Algorithm 3 have almost identical performance. Recall that Algorithm 3 is designed to be an approximation to Algorithm 2 with smaller overhead and easier implementation. These simulation results confirm that it is indeed an accurate approximation.

Further, we note that our policies outperform the round robin policy in all scenarios. With On-Off channels, our policies are able to offload 10% more traffic than the round robin policy even when R is as large as 10. The theoretical analysis in Section VI only proves that the worst-case performance of our policies is better than that of the round robin policy. These simulation results further suggest that our policies are still more preferable on average. The difference between our policies and the round robin policy becomes even larger when general channels are considered. Figure 4 shows that our policies offload 85% more data than round robin when R is 1 and still offload 60% more data when R is 10. This is because the round robin policy does not consider channel capacity, and will use a large amount of time serving clients with poor channel qualities.

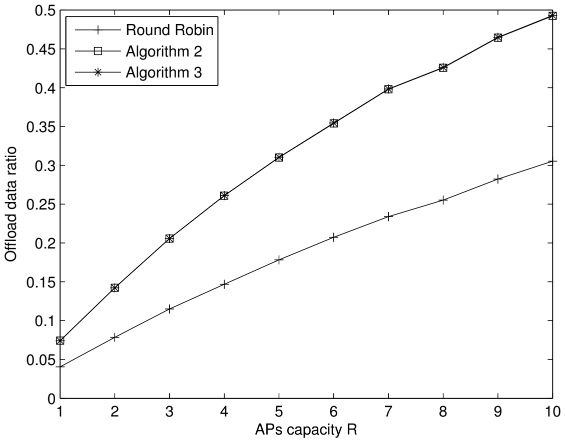

Fig. 4. Performance comparison for general channels.

VIII. Conclusion

In this paper, we study the delayed mobile offloading problem with unpredictable user movement pattern. We aim to download as much data through WiFi as possible. We present two online algorithms for the problem and study their performance by comparing how much data they are able to offload to the optimal offline policy. We propose the first algorithm by primal-dual method and prove that it is $(R, \frac{e^{\frac{1}{R}}}{R[e^{\frac{1}{R}}-1]})$-competitive. After that, we propose an alternative Max-Weight scheduling algorithm which has the same performance with the first one. Then we further study the case that how much capacity is needed by online policies to provide offload guarantees. We compare our algorithm with the commonly-used round robin policy and the result shows that our policy only need to increase half as much capacity to reach the same performance of round robin. We further simulate our algorithm as well as round robin to compare their performance in a randomly generated system. The results show that the two algorithms we proposed have the same performance and they both have higher offloading ratio than round robin policy. Especially, the difference between the performance of our policy and round robin policy is much larger in general channels case than fixed channels case.

References

[1] Cisco, “Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018,” 2014.

[2] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, “Femtocells: Past, present, and future,” *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 3, pp. 497–508, 2012.

[3] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, “Mobile data offloading: how much can wifi deliver?,” in *Proceedings of the 6th International COnference*, p. 26, ACM, 2010.

[4] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, “Mobile data offloading through opportunistic communications and social participation,” *Mobile Computing, IEEE Transactions on*, vol. 11, no. 5, pp. 821–834, 2012.

[5] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: a measurement study and implications for network applications,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pp. 280–293, ACM, 2009.

[6] R. Gass and C. Diot, “An experimental performance comparison of 3g and wi-fi,” in *Passive and Active Measurement*, pp. 71–80, Springer, 2010.

[7] A. Balasubramanian, R. Mahajan, and A. Venkataramani, “Augmenting mobile 3g using wifi,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 209–222, ACM, 2010.

[8] V. F. Mota, D. F. Macedo, Y. Ghamri-Doudane, and J. M. S. Nogueira, “On the feasibility of wifi offloading in urban areas: The paris case study,” in *Wireless Days (WD), 2013 IFIP*, pp. 1–6, IEEE, 2013.

[9] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, “Human mobility patterns and their impact on routing in human-driven mobile networks,”

[10] Y.-B. Lin, C.-C. Huang-Fu, and N. Alrajeh, “Predicting human movement based on telecom’s handoff in mobile networks,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 6, pp. 1236–1241, 2013.

[11] A. J. Nicholson and B. D. Noble, “Breadcrumbs: forecasting mobile connectivity,” in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pp. 46–57, ACM, 2008.

[12] M. H. Cheung and J. Huang, “Optimal delayed wi-fi offloading,” in *Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), 2013 11th International Symposium on*, pp. 564–571, IEEE, 2013.

[13] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, “Multiple mobile data offloading through delay tolerant networks,” in *Proceedings of the 6th ACM workshop on Challenged networks*, pp. 43–48, ACM, 2011.

[14] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan, “Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pp. 1–10, IEEE, 2011.

[15] X. Hou, P. Deshpande, and S. R. Das, “Moving bits from 3g to metro-scale wifi for vehicular network access: An integrated transport layer solution,” in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, pp. 353–362, IEEE, 2011.

[16] A. Barbieri, P. Gaal, S. Geirhofer, T. Ji, D. Malladi, Y. Wei, and F. Xue, “Coordinated downlink multi-point communications in heterogeneous cellular networks,” in *Information Theory and Applications Workshop (ITA), 2012*, pp. 7–16, IEEE, 2012.

[17] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, “When cellular meets wifi in wireless small cell networks,” *Communications Magazine, IEEE*, vol. 51, no. 6, 2013.

[18] S. Singh, H. S. Dhillon, and J. G. Andrews, “Offloading in heterogeneous networks: Modeling, analysis, and design insights,” *Wireless Communications, IEEE Transactions on*, vol. 12, no. 5, pp. 2484–2497, 2013.

[19] O. Bilgir Yetim and M. Martonosi, “Adaptive usage of cellular and wifi bandwidth: an optimal scheduling formulation,” in *Proceedings of the seventh ACM international workshop on Challenged networks*, pp. 69–72, ACM, 2012.

[20] N. Buchbinder and J. (Seffi) Naor, “The design of competitive online algorithms via a primal: Dual approach,” *Found. Trends Theor. Comput. Sci.*, vol. 3, pp. 93–263, Feb. 2009.